DOUBLE MASTER'S DEGREE

MASTER IN MATHEMATICS - M2 MATH

Discrete Mean Field Optimal Stopping: Theoretical Analysis and Deep Learning Algorithms

SPONSORED BY NEW YORK UNIVERSITY

上海纽约大学
NYU SHANGHAI

Supervisor:
Prof. **Mathieu Lauriere** (NYU-Shanghai)
Co-Supervisor:
Prof. **Markus Fischer** (Unipd)

Candidate:
Lorenzo Magnino
Italian Student Number: 2087940
French Student Number: 22300035

September 2024 - Academic Year 2023/2024

Preface

"You can know the name of a bird in all the languages of the world, but when you're finished, you'll know absolutely nothing whatever about the bird... So let's look at the bird and see what it's doing — that's what counts. I learned very early the difference between knowing the name of something and knowing something."

— Richard P. Feynman, "What Do You Care What Other People Think?": Further Adventures of a Curious Character

Everyone searches for their "why" in the things they do and aspire to accomplish. Simply asking this question and beginning the search, I believe, is already an important step. Over these past two years, my search has begun, and while I am still not certain if I have found my "why" in doing research, I am sure that in the process, I felt at home. I believe in ideas and in the determination not to accept what is perceived as established.
I was fortunate enough, seven months ago, to meet Mathieu, who was the right person at the right time. I owe much of this achievement to his trust in me and his willingness to join me on my adventure in Shanghai. In him, I found a kind and generous person, driven by curiosity, who allowed me to express myself fully. He opened the doors of research wide for me, acting as much more than a supervisor, and I cannot thank him enough.
I am also grateful to New York University - Shanghai for sponsoring me during my time in China.
I want to thank all the people who were by my side during these two years, starting with my family, who never stopped supporting me in every choice I made. I want to express my gratitude to the friends I found in Paris, especially those at the Saint Ouen residence, who became my refuge for almost a year of my life. To my lifelong friends from Pordenone, who have shown me how strong and vital our bond is, I am deeply thankful. I know you won't be able to take these words seriously, and reading them together will probably end in laughter—that's also why you are so important to me.
With the hope of never ceasing to look "beyond the numbers."

Lorenzo Magnino
September 2024

Abstract

Optimal stopping is a fundamental problem in optimization, which has found applications in risk management, finance, economics, and recently in fields of computer science, for example in the paradigm of exploration and exploitation in learning algorithms. Here we consider an extension of the standard framework in which a large population of agents cooperatively try to solve a collection of optimal stopping problems. We call multi-agent optimal stopping (MAOS) this problem. The agents interact through the population distribution of states. We study two variants of the problem: in the asynchronous stopping problem, the agents can stop independently, while in the synchronous version, they stop according to the same probabilities. However, the finite-agent problem is hard to solve numerically. Letting the number of agents tend to infinity, we turn to the mean field version of the problem, that we call mean field optimal stopping (MFOS) problem. We prove that it provides an approximately optimal solution for the MAOS problem. Since the usual setting of MFOS is not Markovian, we extend our space so as to establish a dynamic programming principle. Based on this, we propose two deep learning methods for the MFOS problem: the first one directly aims at learning the optimal decision by simulating whole trajectories, while the second one uses the DPP to efficiently learn stopping decisions using backward induction. We develop a theoretical analysis of the algorithm proving the convergence of the value. We present numerical experiments to illustrate the efficacy of our methods. We believe this framework lays the foundation for new advancements in the study of optimal stopping framework.

Contents

List of Figures **7**

List of Algorithms **9**

Notation **14**

1 Introduction **17**

2 Framework and Motivations **23**
- 2.1 Motivations: multi agent model with common noise 23
- 2.2 Mean Field Model 24
- 2.3 Convergence of the measure and ϵ-optimality approximation 27

3 Dynamic Programming **31**
- 3.1 Generale case 31
- 3.2 Synchronous Stopping Times 33

4 Deep Learning Methods: Theoretical Analysis **35**
- 4.1 Algorithms 35
- 4.2 Convergence analysis of the algorithms 35

5 Experiments **45**
- 5.1 Architecture 45
- 5.2 Experiments 46
  - 5.2.1 Towards the Uniform 1D 46
  - 5.2.2 Towards the uniform 2D 48
  - 5.2.3 Drones Match the Target Distribution 51

Conclusion **57**

List of Figures

5.1 Example 1. DA results. Top: Evolution of the distribution and stopping probability at every time step. Bottom: training and testing losses after training. 47
5.2 Example 1. DPP results. Top:Evolution of the distribution and stopping probability at every time step. Bottom: training and testing losses after training. 48
5.3 Example 2. DA results: Top -Evolution of the distribution and stopping probability at every time step. Bottom- training and testing losses after training. 49
5.4 Example 2. DPP results: Top -Evolution of the distribution and stopping probability at every time step. Bottom- training and testing losses after training. 50
5.5 Heat map of the target distribution ρ 51
5.6 3D plot of the target distribution ρ 51
5.7 Example 3. DA results. Top: Evolution of the distribution and stopping probability at every time step. Bottom - training and testing losses after training. 52
5.8 Example 3. Direct Approach: 3D Evolution of the mass 53
5.9 Example 3. DPP results. Top: Evolution of the distribution and stopping probability at every time step. Bottom- training and testing losses after training. 54
5.10 Example 3. DPP 3D Evolution of the mass 55

List of Algorithms

1 Direct Approach for MFOS 36
2 Dynamic Programming Approach for MFOS 37

Notation

Symbol	Description
MULTI AGENT SETTING	
$\mathcal{X}$	state space
X_n^i	state of agent i at time n
$\boldsymbol{\alpha}$	vector of pure stopping decisions.
$\boldsymbol{X}_n^{\boldsymbol{\alpha}}$	vectors of states at time n driven by vectors of controls $\boldsymbol{\alpha}$
$p_n^i(\boldsymbol{X}_n^{\boldsymbol{\alpha}})$	probability of agent i to stop at time n
$\mu_n^{N,\boldsymbol{\alpha}}$	proportion of agents at x a time n.
μ_0	initial distribution of the agents
$Be(p)$	Bernoulli distribution of parameter p
$\pi^i(\cdot\|\boldsymbol{X}_n^{\boldsymbol{\alpha}})$	distribution of the control α of the player i
ϵ^i, ϵ^0	random noise of the player i and the common noise
$F(n, x, \mu, \epsilon, \epsilon^0)$	transition function
τ^i	first time for player i at which the decision is to stop
$f(n, x, \mu), \Phi(x, \mu), g(\mu)$	running cost, stopping cost, terminal cost
$\mathcal{P}(X)$	set of probability distributions on $\mathcal{X}$
$J^N(\boldsymbol{\alpha})$	social cost (objective function)
MEAN FIELD SETTING	
A_n^α	"alive" condition of at time n of the representative player
Y_n^α	extended state at time n of the representative player
$\mathcal{S}$	extended state space

Symbol	Description
$J(\alpha)$	mean field social cost
$\mathcal{F}_n^0$	filtration at time n of the common noise process
$\mathcal{L}(Y_n^\alpha \| \mathcal{F}_n^0)$	law of the process Y_n^α conditioned on the filtration $\mathcal{F}_n^0$
$\nu_n^\alpha(x,a)$	proportion of the mass in state x that has "alive" condition equal to a
ν_X^p	first marginal of ν_n^p
$\bar{F}$	mean field dynamics
$\mathcal{H}$	set of all function that represent a stopping probability
B^0	the set of all possible realization of the common noise
$\mathbb{P}^0(z \to x)$	transition probability to go from z to x condition on the common noise
$\mathbb{E}^0$	expectation under $\mathbb{P}^0$
$\mathcal{P}_{n,T}$	set of all policies defined in $[n,T]$
$\|\| \cdot \|\|$	L^1 norm
$\Psi(n,\nu,p)$	represent the combination between the running cost and the stopping cost at time n
$V_n(\nu)$	value function at time n when we start with distribution ν
$\nu_m^{n,\nu,p}$	detailed distribution at time $m > n$ conditioned on initial conditions.

Chapter 1

Introduction

What do a crowd motion, a flock of birds and stock market have in common? All these phenomena can be described by a mathematical theory called mean field theory. Originated in 2006 **?** and **?** to model decision problems in systems of interacting agents, mean field theory describes two types of situations: the first when each agent that is part of the population (each bird in the flock, particles in the ensemble they form etc..) acts to achieve (maximize or minimize) its own gain or cost while the second when a central planner controls each agent with the goal of maximizing/minimizing a social gain/cost. The former is called Mean Field Games while the latter Mean Field Control, which has gained great interest in recent years and this thesis focuses on the latter type of study.

Mean Field Control (MFC): The main idea of that model is that when the agents we consider are defined as indistinguishable, that is, possessing the same characteristics and are interchangeable with each other, then when their number becomes very large, their behavior can be described by a representative agent. Such an agent acts as a function of the macroscopic state of the system and is described by the distribution of the population. Our framework lies on finite state space and discrete time space with finite horizon. Going a little bit deeper in the description, every agent i follows a dynamic

$$X_{n+1}^{i,\boldsymbol{\alpha}} = F(n, X_n^{i,\boldsymbol{\alpha}}, \mu_n^{\boldsymbol{\alpha},N}, \alpha_n^i, \epsilon_{n+1}^i, \epsilon_{n+1}^0) \qquad X_0^{i,\boldsymbol{\alpha}} = \xi^i,$$

where $X_{n+1}^{i,\boldsymbol{\alpha}}$ is the state of agent i at time $n+1$ controlled by α_n^i and the transition that depend on the empirical distribution of the population defined as $\mu_n^{\boldsymbol{\alpha},N} := \frac{1}{N}\sum_{i=1}^N \delta_{X_n^{i,\boldsymbol{\alpha}}}$, a presence of individual noise ϵ_{n+1}^i and the presence of the common noise ϵ_{n+1}^0. While individual noise is different (actually i.i.d) for each agent, common noise is thought of as a random source acting on each component of the population in the same way (for example, the intensity and direction of the wind for each bird within a flock). Note also how the dynamics of each agent at a given instant is affected by the distribution of the entire population at the same instant ($\mu_n^{N,\boldsymbol{\alpha}}$ within the formula) and this term is called mean field interaction. The entire population of N agents want to solve the problem of minimazing the social cost described by

$$\sup_{\alpha\in\mathcal{A}} \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\sum_{n=0}^{T} f^i(n, X_n^{i,\boldsymbol{\alpha}}, \mu_n^{N,\boldsymbol{\alpha}}) + g(\mu_T^{N,\boldsymbol{\alpha}})\right]$$

where f describes a running cost (i.e a cost that it is payed at every time step) and g is the terminal cost. When the number of agents tends to infinity the contribution of each individual agent through the empirical distribution becomes zero, and thus the agents become independent. In this situation, an averaging effect takes place and due to the symmetry of the problem we have the so called *propagation of chaos* phenomenon:

$$\frac{1}{N}\sum_{i=0}^{N}\delta_{X_n^{i,\boldsymbol{\alpha}}} \to \mathcal{L}(X_n^{\alpha}), \qquad \text{as } N \to \infty.$$

In this mean field regime, each agent follows a dynamics of the McKean - Vlasov type described by

$$X_{n+1}^{\alpha} = F(n, X_n^{\alpha}, \mathcal{L}(X_n^{\alpha}|\mathcal{F}_n^0), \alpha_n, \epsilon_{n+1}, \epsilon_{n+1}^0) \qquad X_0^{\alpha} \sim \mu_0,$$

where μ_0 is the initial distribution of the system and $\mathcal{L}(X_n^{\alpha}|\mathcal{F}_n^0)$ is the law of the agent at time n conditioned on the filtration at time n generated by the common noise process (we can think the filtration as the "story", i.e all the realizations, of the common noise process up to time n). The social cost is now formulated as

$$\sup_{\alpha\in\mathcal{A}} \mathbb{E}\left[\sum_{n=0}^{T} f(n, X_n^{\alpha}, \mathcal{L}(X_n^{\alpha}|\mathcal{F}_n^0) + g(\mathcal{L}(X_T^{\alpha}|\mathcal{F}_T^0))\right].$$

The mean field optimal stopping belonges, in some extent, to this type of class. But let us take a step back and briefly describe what the problem of optimal stopping is.

Optimal Stopping: Optimal stopping theory is a field that focuses on determining the most advantageous moment to halt a particular process in order to achieve the greatest expected reward or the smallest expected cost. This concept has widespread applications across various domains, including finance, where it assists in identifying the ideal time to buy or sell assets for maximum profit. In the context of gambling, optimal stopping strategies guide players on when to quit to maximize winnings or minimize losses. Furthermore, it applies to search and matching problems, such as determining the best time to accept a job offer during a job hunt. Beyond these areas, optimal stopping theory is utilized in fields such as healthcare decision-making, quality control processes, and even online algorithms, demonstrating its versatility and importance in making timely and strategic decisions across diverse scenarios. From the probabilistic point a view, given a filtration $\mathcal{F}$ a stopping time τ is a non negative random variable such that $\{\tau = n\} \in \mathcal{F}_n$ for every $n = 0, \dots, T$ (i.e it is possible to know if n it is the right moment to stop only with the information that I have from the beginning up to time n). The optimal stopping problems want to solve the following question: given a dynamical system Z and a reward function Φ the goal is to find the optimal time τ in order to maximize $\mathbb{E}[\Phi(Z_\tau)]$.
This setting can be extended to a multi-agent context in which each agent follows a stopped dynamic, and the cost can also be extended making us find ourselves in a similar situation as

before in the context of mean-field control. In this case the control α is when we decide to stop each agent and unlike most cases of mean filed control this affects the whole trajectory from that moment on. It is therefore referred to as non-Markovian setting. This aspect will play a crucial role in our discussion and we are going to highlight the possible connections and differences to MFC.

Dynamic Programming Principle As written in **?**: *"Life can only be understood going backwards, but it must be lived going forward. - Kierkegaard".* Formalized by Richard Bellman in the 1950s the dynamic programming principle stands that the optimal solution of a problem can be decomposed through the optimal solutions of the subproblems associated to it. A key aspect in applying this method is that the problem we are studying need to verifies the so called "Principle of Optimality" that states that "An optimal policy has the property that whatever the initial state and decisions are, the remaining decisions must constitute an optimal policy with regard to state resulting from the first decision". In shortste path problems (where the goal is to find the shortest path from A to B in a specific graph) it means that the shortest path between two nodes contains the shortest paths between intermediate nodes. A central element of this method is the Bellman's Equation. It is a recursive relationship between the value (expected payoff/cost) of a given time and the value of the next time step. In this way the problem can be decomposed into subproblems and solved by backward induction. In the literature most of the times it is described by the following formula

$$V(s) = \max_a \left\{ r(s,a) + \lambda \sum_{s'} P(s'|s,a)V(s') \right\}$$

where $V(s)$ is the value of the state s, a is the action, $r(s,a)$ is the reward when we are in state s and we use the action a, $P(s'|s,a)$ is the probability to go from s to s' when the action a is used and finally λ is the discount factor that permits us to taking into account more the present than the future reward. In our formulation this formula will have the following form:

$$V_n(s) = \sup_a \left\{ \Psi(n,s,a) + \mathbb{E}\left[V_{n+1}(s')|s,a\right] \right\}$$

Developing a dynamic programming principle (DPP) for a problem gives the opportunity to develop algorithms that perform better than other method.

Deep Learning Methods: Deep learning has become a cornerstone of modern artificial intelligence, powering advancements in diverse fields such as image recognition, natural language processing, and autonomous systems. To tackle the MFOS problem numerically, we consider two deep learning-based approaches grounded in distinct formulations:

1. **Direct Approach (DA):** The first approach attempts to directly minimize the mean-field social cost $J(p)$ by optimizing over all possible stopping probability functions $p : \{0,\ldots,T\} \times \mathcal{X} \to [0,1]$. Here, a time-dependent neural network is employed to parameterize the stopping decision, and training involves iteratively updating the network weights to reduce the total social cost across the population.

2. **Dynamic Programming Approach (DP):** The second, more structured approach leverages the Dynamic Programming Principle (DPP). This method involves solving for the optimal stopping probability via backward induction. For each timestep n, the algorithm learns the true value function $V_n(\nu)$ by solving an optimization problem over one-step stopping probabilities, effectively training a sequence of neural networks backward in time.

Both approaches utilize deep learning's ability to handle high-dimensional spaces and non-linear dependencies. In the DA, the stopping decision is learned holistically across all time steps, which can be seen as a global optimization problem. On the other hand, the Dynamic Programming Approach breaks down the problem into a series of smaller, more manageable optimization tasks, making use of the recursive nature of the problem.
The DA provides a straightforward application of deep learning to the MFOS problem, but it may struggle with the curse of dimensionality and local minima due to the complexity of directly optimizing the entire social cost. The DPP mitigates some of these challenges by focusing on optimizing at each time step independently, though it requires careful coordination across the different time steps to ensure consistency in the learned stopping policies.

Related Works

Optimal stopping. Optimal stopping (OS) problems model situations in which the goal is to stop a dynamical system so as to minimize a cost. To make the models more realistic, randomness is incorporated in the dynamics. A typical optimal stopping problems (in discrete time) is the famous problem of Job search (also called House Selling or Secretary problem) [Lippman and McCall, 1976]. Another important example (in continuous time) is the pricing of American options in finance [Björk, 2009]. More recently, there are also applications in machine learning [Wang et al., 1993]. From the theoretical viewpoint, optimal stopping served as an important step in the development of the optimal control theory, since many ideas have been developed in the former setting before being extended to the more complex optimal control setting. See [Shiryaev, 2007] for more background. In general, there are no explicit solutions, which justified the introduction of various numerical methods.
Computational methods. Typical numerical methods to solve such problems rely on a characterization of the value function through a backward equation. In continuous time and space, it takes the form of a partial differential equation, which can be solved using methods such as finite elements [Achdou and Pironneau, 2005]. The problem can also be solved using probabilistic methods such as [Bally and Pagès, 2003]. However, except in some special cases, the classical methods do not scale well to problems with high dimensional states. In the recent years, deep learning methods have leveraged the power of deep neural networks to tackle optimal problems in high dimension. Becker et al. [2019] proposed to learn the stopping decision at each time step using a deep network by exploiting dynamic programming, and Herrera et al. [2023] extended the approach using randomized neural networks. Several other approaches have been proposed, particularly for continuous time OS problems, such as learning the stopping boundary [Reppen et al., 2022].

Multi-agent optimal stopping. Many real-world scenarios involve several agents and not just one, for instance in finance [Kobylanski et al., 2011], economics [Rosenberg et al., 2007, Kleinberg et al., 2021], and robotics [Crowther, 2023]. The problem's complexity increases with the number of agents and quickly becomes intractable unless simplifying approximations are made. In this work, we will focus on a class of multi-agent optimal stopping (MAOS) problems in which the agents cooperate to minimize a social cost, and both the dynamics and the cost depend only on the individual agent's state and the empirical distribution of agents. This allows us to approximate the solution using a mean field optimal stopping (MFOS) problem. MFOS has recently been studied in continuous time and spaces [Talbi et al., 2023, 2022]. In the present work, we focus on discrete time and finite space models for the agents. However, since the optimal stopping decisions and the value functions are functions of the whole population distribution as we will show, the problem is intrinsically high dimensional, which motivates the use of deep learning methods.
Mean Field Control. In the discrete time mean field control [**?**] describe stochastic optimal control problem of nonlinear mean-field systems and reformulate the problem as a deterministic problem providing a dynamic programming principle in the general form. For other work related of MFC see [Bensoussan et al., 2013, Carmona and Delarue, 2018]. Applications include crowd motion [Achdou and Lasry, 2019], flocking [Fornasier and Solombrino, 2014], finance [Carmona and Laurière, 2021] opinion dynamics [Liang and Wang, 2019], and artificial collective behavior [Gu et al., 2021, Cui et al., 2024], among others.

Main Contributions

Our contribution are the follwing:

- we present a discrete time and space MAOS with common noise and its MFOS limit with the presence of randomized stopping times and we show that MFOS provides an approximately otpimal stopping rule for MAOS, and and the quality of approximation increases with the number of agents in the system.

- we prove a Dynamic Programming Principle for MFOS problems in two different setting: asynchronous stopping decision and asynchronous stopping decision.

- we develop a theoretical analysis for the convergence of the algorithm.

- we propose two deep learning method in order to solve MFOS, by learning the optimal stopping decisions as a function of the whole population distribution.

- we present numerical experiments through three example of increasing complexity in order to validate our model.

Organization

The thesis unfolds as follows. In the next chapter we present the main framework describing the mean field model. Furthermore we discuss the motivation of the mean field setting due to the finite multi agent problem. We prove how the first one approximate well the latter presenting the famous result known as *propagation of chaos*. In Chapter 3 we establish the dynamic programming principle for both synchronous and asynchronous stopping decision. In chapter 4 we present the algorithms that we used for the experiments and we study the convergence. Finally last chapter present the experiments done and we conclude with conclusion, consideration and further works.

Chapter 2

Framework and Motivations

2.1 Motivations: multi agent model with common noise

The mean field problem that we will solve is motivated by the N-agent problem that we are about to describe. Let T be a time horizon and let N be the number of agents that are interacting. Let $\mathcal{X}$ be a finite state space. Each agent i has a state denoted by X_n^i at time n. At time n, each agent stops with probability $p_n^i(\boldsymbol{X}_n^{\boldsymbol{\alpha}})$. We introduce α_n^i a random variable taking value 0 if the agents continue and 1 if it stops. We denote by $\pi^i(\cdot|\boldsymbol{X}_n^{\boldsymbol{\alpha}}) = Be(p_n^i(\boldsymbol{X}_n^{\boldsymbol{\alpha}}))$ its distribution, which is a Bernoulli distribution. We denote by $\boldsymbol{X}_n^{\alpha} = (X_n^1, \dots, X_n^N)$ and $\boldsymbol{\alpha} = (\alpha^1, \dots, \alpha^N)$ the vectors of states and stopping decisions at time n.

Dynamics. We assume that the agents are indistinguishable and interact in a symmetric fashion, i.e. through their empirical distribution $\mu_n^{N,\boldsymbol{\alpha}}(x) := \frac{1}{N}\sum_{i=1}^N \delta_{X_n^{i,\boldsymbol{\alpha}}}(x)$, which is the proportion of agents at x at time n.The states evolve according to: for every $i = 0, \dots, N$,

$$\begin{cases} X_0^{i,\boldsymbol{\alpha}} \sim m_0 \\ \alpha_n^i \sim \pi^i(\cdot|\boldsymbol{X}_n^{\boldsymbol{\alpha}}), \qquad X_{n+1}^{i,\boldsymbol{\alpha}} = \begin{cases} F(n, X_n^{i,\boldsymbol{\alpha}}, \mu_n^{N,\boldsymbol{\alpha}}, \epsilon_{n+1}^i, \epsilon_{n+1}^0), & \text{if } \sum_{m=0}^n \alpha_m^i = 0 \\ X_n^{i,\boldsymbol{\alpha}}, & \text{otherwise,} \end{cases} \end{cases} \tag{2.1}$$

where ϵ_n^i is a random noise impacting the evolution of agent i, m_0 is the initial distribution and ϵ_n^0 is the common noise that affects the dynamics of all agents equally. Let us define the stopping time for agent i: $\tau^i = \inf\{n \geq 0 : \sum_{m=0}^n \alpha_m^i \geq 1\}$, which the first time for player i at which the decision is to stop.

Objective function. Let us consider the running cost $f : [\![0, T]\!] \times \mathcal{X} \times \mathcal{P}(\mathcal{X}) \to \mathbb{R}$, where $\mathcal{P}(\mathcal{X})$ denotes the set of probability distributions on $\mathcal{X}$. Then let us consider $\Phi : \mathcal{X} \times \mathcal{P}(\mathcal{X}) \to \mathbb{R}$, where $\Phi(x, \mu)$ denotes the cost that an agent incurs if she stops at x and the current population distribution is μ. As a terminal cost we are going to define $g : \mathcal{P}(\mathcal{X}) \to \mathbb{R}$ which depends only on the distribution of agents at the terminal stage. The goal for the all the N agents is collectively minimize the following social cost function:

$$J^N(\boldsymbol{\alpha}) = \mathbb{E}\left[\frac{1}{N}\sum_{i=0}^{N}\left(\sum_{n=0}^{\tau^i-1} f(n, X_n^{i,\boldsymbol{\alpha}}, \mu_n^{N,\boldsymbol{\alpha}}) + \Phi(X_{\tau^i}^{i,\boldsymbol{\alpha}}, \mu_{\tau^i}^{N,\boldsymbol{\alpha}}) + g(\mu_T^{N,\boldsymbol{\alpha}})\right)\right], \tag{2.2}$$

with the convention that if $\tau^i = 0$ then the second summation is not computed for that agent i. In other words, the problem consists in finding $(\alpha^1, \dots, \alpha^N) \in \operatorname{argmin} J^N$.

2.2 Mean Field Model

As mentioned earlier, if we let the number of players tend to infinity, we expect, thanks to *conditional* propagation of chaos type results, that the states will become "conditionally" independent and each state will have the same evolution, which will be a non-linear Markov chain. This is the equivalent of *conditional* McKean-Vlasov dynamics. Furthermore, it can be expected that by solving the limiting problem, we obtain an approximate solution for the problem with $N < +\infty$ agents. More precisely, passing formally to the limit in the dynamics (2.1), we obtain the following evolution:

$$\begin{cases} X_0^\alpha \sim \mu_0 \\ \alpha_n \sim \pi(\cdot|X_n^\alpha) = Be(p_n(X_n^\alpha)), \qquad X_{n+1}^\alpha = \begin{cases} F(n, X_n^\alpha, \mu_n^\alpha, \epsilon_{n+1}, \epsilon_{n+1}^0), & \text{if } \sum_{m=0}^n \alpha_m = 0 \\ X_n^\alpha, & \text{otherwise,} \end{cases} \end{cases} \tag{2.3}$$

where $p_n(x)$ denotes the probability with which the agent continues if she is in state x at time n, and μ_n^α is the distribution of X_n^α itself conditioned on the common noise, which we may also denote by $\mathcal{L}(X_n^\alpha|\mathcal{F}_n^0)$, where $\mathcal{F}^0$ contains the realization of the common noise until time n.

We want to emphasize the fact that the introduction of randomized stop times for individual agents is crucial for our purpose and differs from the randomization of the central planner on policies.

We can define, in the same way we did before, the first time in which the control α is 1 as $\tau := \inf\{n \geq 0 : \sum_{m=0}^n \alpha_m \geq 1\}$. Then the social cost function in the mean field problem is defined as:

$$J(\alpha) = \mathbb{E}\left[\sum_{n=0}^{\tau-1} f(n, X_n^\alpha, \mu_n^\alpha) + \Phi(X_\tau^\alpha, \mu_\tau^\alpha) + g(\mu_T^\alpha)\right]. \tag{2.4}$$

Notice that here the expectation has the effect of averaging over the whole population, so there is no counterpart to the empirical average that appears in the finite agent cost (2.2). To stress the dependence on the initial distribution, we will sometimes write $J(\alpha, m_0)$.

We want to show with an example that the extension to randomized stopping times is necessary in the mean-field formulation, because when we try to plug an optimal strategy into the N-agent problem, we notice that the latter is no longer optimal.

Example 1 (Why do we need randomization in the control?)**.** *Let us consider the following scenario: we take the state space $\mathcal{X} = \{D, C\}$ and initial distribution $\mu_0 = 3/4\delta_D + 1/4\delta_C$;*

transition function $F(D, x, \mu, \epsilon) = C$, $F(C, x, \mu, \epsilon) = D$, meaning that the system at any time step, can stop or switch the state. We take as social cost:

$$\Phi(x, \mu) = \begin{cases} 1 & \text{if } \mu(x) \leq 1/2 \\ 5 & \text{if } \mu(x) > 1/2. \end{cases} \tag{2.5}$$

Notice that without allowing the randomized stopping rule the optimal value we can achieve is $V^* = 3/4 \cdot 5 + 1/4 \cdot 1 = 4$, which corresponds to stop all the distribution (in every state) at time $n = 0$. In the end, this formulation cannot reflect the optimum in the association of N agents. Indeed when we plug this policy into the N agent formulation we obtained the value $V^N = 1/N(3N/4 \cdot 5 + N/4 \cdot 1) = 4$, which is not optimal since we can use the strategy (which is going to be optimal for the N-agent problem) to stop, at time 0, only the $1/3$ of players in state D, allowing the others to change state. This leads to a final configuration of $m_1 = 1/2\delta_D + 1/2\delta_C$ and a value of $V^{*,N} = 1/N(N/4 \cdot 5 + 3N/4 \cdot 1) = 2 < V^N = 4$.
In particular, we want to emphasize the fact that, without allowing a randomized stopping time in the MF formulation, we find an optimal state-dependent strategy, which corresponds , in the problem with finite agents, to the fact that every player in the same state will have the same stopping time.

Mean Field model with extended state: A key step towards building efficient algorithms is dynamic programming, which relies on Markovian property. However, in its current form the above problem is not Markovian. In fact a player who stops at time $n = m$ must remain stationary in the interval $[\![m + 1, T]\!]$ and when this information is not contained in the state variable we are unable to take it into account.
To make the system Markovian, we need keep track of the information about whether the player's process has been stopped in the past. This information is not contained in the state so we need to extend the state. Let $A^\alpha = (A^\alpha_n)_{n=0,\ldots,T}$ the process such that $A^\alpha_n = 0$ if the agent has *already* stopped before time n, and 1 otherwise. It is important to notice that if the agent is stopped precisely at time n then, we still have $A^\alpha_n = 1$ but $A^\alpha_m = 0$ for every $m > n$. We define the ***extended state*** as: $Y^\alpha_n = (X^\alpha_n, A^\alpha_n)$ which takes value in the extended state space $\mathcal{S} := \mathcal{X} \times \{0, 1\}$
At this point we introduce the probabilistic framework for a rigorous formulation of our problem.

Probabilistic framework: Let $\mathcal{S}$ be a finite state space and let T be a finite time horizon. Denote by $\mathcal{P}(\mathcal{S})$ the set of probability distributions on $\mathcal{S}$, that we identify with the simplex on $\mathcal{S}$, i.e., $\mathcal{P}(\mathcal{S}) = \{\nu \in [0, 1]^{|\mathcal{S}|} : \sum_{y \in \mathcal{S}} \nu(y) = 1\}$. Let $(\epsilon^0_n)_{n=0,\ldots,T}$ be a stochastic process playing the role of the common noise such that ϵ^0_{n+1} is independent of $\epsilon^0_1, \ldots, \epsilon^0_n$ and $\epsilon_0 = 0$ and let call $(\mathcal{F}^0_n)_{n=0,\ldots,T}$ its canonical completed filtration. However to handle the extra randomness given by probability to stop or not (defined by a Bernoulli distribution) we are going to extend our space. In order to do this we recall the fact that a Bernoulli distribution of a parameter p can be built from a random noise U distributed as a uniform distribution on the interval $[0, 1]$. Indeed if the result of the random variable $U = u$ is less than p then the Bernoulli gives 1 as result, 0 otherwise. Let $(\Omega, \mathcal{G}, \mathbb{P})$ the probability space where $\Omega := \Omega_E \times \Omega_U := \{(\epsilon^0_k, u_k)_{k=0,\ldots,T}\}$ with

$(\epsilon_k^0)_{k=0,\dots,T}$ independent from $(u_k)_{k=0,\dots,T}$ and for every $k = 0, \dots, T$, $u_k \sim \mathcal{U}[0,1]$ (notice that the distribution of ϵ^0 can be arbitrary while er are fixing the uniform distribution in $[0,1]$ for u since we are going to construct a "controlled" Bernoulli distribution starting from this uniform distribution). $\mathbb{P}$ is the probability measure define in this extended space and $\mathcal{G}$ is the σ-algebra that contains all the measurable events.

Then, the dynamics (2.3) of the representative player can be rewritten as:

$$\begin{cases} X_0^\alpha \sim \mu_0, \qquad A_0^\alpha = 1 \\ \alpha_n \sim \pi(\cdot|X_n^\alpha) = Be(p_n(X_n^\alpha)) \\ A_{n+1}^\alpha = A_n^\alpha \cdot (1-\alpha_n) \\ X_{n+1}^\alpha = \begin{cases} F(n, X_n^\alpha, \mu_n^\alpha, \epsilon_{n+1}, \epsilon_{n+1}^0), & \text{if } A_n^\alpha \cdot (1-\alpha_n) = 1 \\ X_n^\alpha, & \text{otherwise.} \end{cases} \end{cases} \tag{2.6}$$

The idea of extending the state using the extra information is similar to [Talbi et al., 2023] in continuous time and space. The mean field social cost (2.4) can rewritten as:

$$J(\alpha) = \mathbb{E}\Bigg[\sum_{n=0}^{T} \left(f(n, X_n^\alpha, \mu_n^\alpha) A_n^\alpha (1-\alpha_n) + \Phi(X_n^\alpha, \mu_n^\alpha) A_n^\alpha \alpha_n\right) + g(\mu_T^\alpha)\Bigg] \tag{2.7}$$

Actually, notice that the part of expectation amounts to taking a sum with respect to the extended state's distribution. Indeed we handle to type of randomness due to the presence of the common noise and the control (α). Let us denote by $\nu_n^p = \mathcal{L}(Y_n^\alpha|\mathcal{F}_n^0)$ the distribution at time n conditioned on the common noise. We are going to denote ν_X^p the first marginal of ν^p (sometimes also denoted by μ). We want to stress the fact that it does not really depend on α but only on the stopping probability p, so we use the superscript p when referring to ν. Note that with the presence of common noise the limit distribution ν is not deterministic, but it is a random variable that evolves conditionally with respect to the common noise. This distribution evolves according to the mean field dynamics:

$$\begin{cases} \nu_0^p(x,0) = 0, \quad \nu_0^p(x,1) = \mu_0(x), \qquad x \in \mathcal{X}, \\ \nu_{n+1}^p = \bar{F}(\nu_n^p, p_n, \epsilon_{n+1}^0), \end{cases} \tag{2.8}$$

where the function $\bar{F}$ is defined as follows. We denote by $\mathcal{H}$ the set of all function $h : \mathcal{X} \to [0,1]$, which represents a stopping probability (for each state) and by B^0 the set of of all possible realization of the common noise. Then, $\bar{F} : \mathcal{P}(\mathcal{S}) \times \mathcal{H} \times B^0 \to \mathcal{P}(\mathcal{S})$ is defined by: for every $x \in \mathcal{X}, a \in \{0,1\}$, $\bar{F}(\nu, h, \epsilon^0)$ is the distribution generated by doing one step, starting from ν, using the stopping probabilities h, according to the realization of the common noise ϵ^0. Mathematically,

$$
\begin{aligned}
(\bar{F}(\nu, h, \epsilon^0))(x,a) &= \Big(\nu(x,0) + \nu(x,1)h(x)\Big)(1-a) + \\
&\quad + \Big(\sum_{z\in\mathcal{X}} \nu(z,1)\Big(\mathbb{P}(z\to x|\epsilon^0)(1-h(z))\Big)\Big)a \\
&= \Big(\nu(x,0) + \nu(x,1)h(x)\Big)(1-a) + \Big(\sum_{z\in\mathcal{X}} \nu(z,1)\Big(\mathbb{P}^0(z\to x)(1-h(z))\Big)\Big)a
\end{aligned}
\tag{2.9}
$$

where $\mathbb{P}(z \to x|\epsilon^0)$ is the transition matrix associated to the unstopped process X conditionally to the common noise, i.e. it is the probability to go from the state z to the state x knowing that we are not going to stop in x and we can observe the realization of ϵ^0. Notice that in general the transitions may depend on ν itself. So the last equation can be written more succinctly in a matrix-vector product but the transition matrix depends on ν itself, which is why this type of dynamics is sometimes referred to a *non-linear* dynamics.
The mean field social cost can be rewritten purely in terms of the distribution as follows:

$$
J(p) = \mathbb{E}^0\left[\sum_{n=0}^{T}\sum_{(x,a)\in\mathcal{S}}\Big(f(n,x,\nu^p_{X,n})a(1-p_n(x))\nu^p_n(x,a) + \nu^p_n(x,a)\Phi(x,\nu^p_{X,n})ap_n(x)\Big) + g(\nu^p_{X,T})\right]
\tag{2.10}
$$

where the only randomness that we have to handle with $\mathbb{E}^0$ is the realization of the common noise. Furthermore $p : \{0,\dots,T\}\times\mathcal{X}\to[0,1]$ is the function that associates at every time step and state the probability to stop (in that state at that time). Let us define $\mathcal{P}_{0,T}$ the set of all such functions.
The link with the above formulation is that $\alpha_n(x)$ is distributed according to $Be(p_n(x))$, and $\nu^p_n := \mathcal{L}(Y^\alpha_n|\mathcal{F}^0_n)$ is the extended state's distribution. Moreover, $\nu^p_n(x,0)$ is the mass in x that has stopped. Last, $\mathcal{L}(X^\alpha_n|\mathcal{F}^0_n) = \nu^p_{X,n} = \sum_{a\in\{0,1\}}\nu^p_n(\cdot,a)$ is the first marginal of this distribution.

2.3 Convergence of the measure and ϵ-optimality approximation

In this section we want to provide two main results for our discussion. The first regard the convergence of the measures, also called *conditional propagation of chaos*, while the second is to recover the approximate solution for the finite agent model.
These two results are classics in the literature and are fundamental to solving the mean-field control problem to retrieve finite-agent solutions.
Let us recall the extended-state dynamics and the cost for the N-agent problem. Let us fix the following notation $\nu^{N,p}_m := \frac{1}{N}\sum_{i=1}^N \delta_{Y^{i,\alpha}_m}$ and $\nu^p_m := \mathcal{L}(Y^\alpha_m|\mathcal{F}^0_m)$.

$$\begin{cases} X_0^{i,\boldsymbol{\alpha}} \sim \mu_0, \qquad A_0^{i,\boldsymbol{\alpha}} = 1 \\ \alpha_n^i \sim \pi_n^i(\cdot|X_n^{i,\boldsymbol{\alpha}}) = Be(p_n(X_n^{i,\boldsymbol{\alpha}})) \\ A_{n+1}^{i,\boldsymbol{\alpha}} = A_n^{i,\boldsymbol{\alpha}} \cdot (1-\alpha_n^i) \\ X_{n+1}^{i,\boldsymbol{\alpha}} = \begin{cases} F(n, X_n^{i,\boldsymbol{\alpha}}, \frac{1}{N}\sum_{j=0}^N \delta_{X_n^{j,\boldsymbol{\alpha}}}, \epsilon_{n+1}^i, \epsilon_{n+1}^0), & \text{if } A_n^{i,\boldsymbol{\alpha}} \cdot (1-\alpha_n^i) = 1 \\ X_n^{i,\boldsymbol{\alpha}}, & \text{otherwise.} \end{cases} \end{cases} \tag{2.11}$$

The social cost is defined as:

$$\begin{aligned} J^N(\alpha^1, \dots, \alpha^N) = \mathbb{E}\Big[\frac{1}{N}\sum_{i=0}^N \Big(\sum_{n=0}^T f(n, X_n^{i,\boldsymbol{\alpha}}, \mu_n^{N,\boldsymbol{\alpha}}) A_n^{i,\boldsymbol{\alpha}}(1-\alpha_n^i) + \\ + \; \Phi(X_n^{i,\boldsymbol{\alpha}}, \mu_n^{N,\boldsymbol{\alpha}}) A_n^{i,\boldsymbol{\alpha}} \alpha_n^i + g(\mu_T^{N,\boldsymbol{\alpha}})\Big)\Big], \end{aligned}$$

that we can rewrite only in terms of the distribution and using the control p instead of α as

$$\begin{aligned} J^N(p, \dots, p) = \mathbb{E}\Big[\sum_{(x,a)\in\mathcal{S}} \Big(\sum_{n=0}^T \nu_n^{N,p}(x,a) f(n, x, \mu_n^{N,\boldsymbol{\alpha}}) a(1-p_n(x)) + \\ + \; \nu_n^{N,p}(x,a)\Phi(x, \mu_n^{N,\boldsymbol{\alpha}}) a p_n(x) + g(\mu_T^{N,\boldsymbol{\alpha}})\Big)\Big] = \\ = \mathbb{E}\Big[\sum_{n=0}^T \Psi(n, \nu_n^{N,p}, p_n) + g(\mu_T^{N,\boldsymbol{\alpha}})\Big], \end{aligned} \tag{2.12}$$

with Ψ defined as $\Psi(n,\nu,p) := \sum_{(x,a)\in\mathcal{S}} f(n,x,\nu_{X,n})a(1-p_n(x))\nu_n(x,a) + \nu_n(x,a)\Phi(x,\nu_{X,n})ap_n(x)$. Let us define $||\mu-\nu|| := \sum_{y\in\mathcal{S}} |\mu(y)-\nu(y)|$.

Assumption 1. *Let $L_p > 0$ and let us define $\mathcal{P} := \{p : [0,T] \times \mathcal{X} \times \mathcal{P}(\mathcal{S}) \to [0,1] : p$ is L_p-Lipschitz$\}$, the set of all possible admissible policies. Let $K > 0$ and let us assume that the mean field dynamics satisifies: $|\bar{F}(\nu,p,e) - \bar{F}(\nu',p',e)| \le C(e)(\|\nu-\nu'\| + |p-p'|)$ with $C(e) \le K$ for $e \in B^0$ where B^0 is the set of all possible realization of the common noise. Assume also that the function $\Psi : [\![0,T]\!] \times \mathcal{P}(\mathcal{X} \times \{0,1\}) \times \mathcal{P}(\mathcal{X}) \to \mathbb{R}$ defined as $\Psi(n,\nu,p) := \sum_{(x,a)\in\mathcal{S}} f(n,x,\nu_{X,n})a(1-p_n(x))\nu_n(x,a) + \nu_n(x,a)\Phi(x,\nu_{X,n})ap_n(x)$ is L_Ψ-Lipschitz uniformly with respect to time and the terminal cost g is L_g-Lipschitz.*

Summing up, these hypotheses allow us to say that if we start from two distributions that are close to each other (or converge to each other) then the images through these functions will remain close to each other (or converge to each other).

Lemma 1 (Conditional propagation of chaos)**.** *Suppose Assumption 1 holds. Given the dynamics* (2.11)*,* (2.6) *it holds for every $n = 0, \dots, T$:*

$$\sup_{p\in\mathcal{P}_{n,T}} \mathbb{E}\left[\|\nu_n^{N,p} - \nu_n^p\|\right] \le (K(1+L_p)^n + 1)\frac{\sqrt{2|\mathcal{X}|}}{\sqrt{N}} \tag{2.13}$$

Proof. We are going to follow an induction argument over the time steps:
Initialization: for $n = 0$, since the samples are iid at time 0, by the law of large numbers (LLN) we have:

$$\sup_{p\in\mathcal{P}} \mathbb{E}\left[\|\nu_0^{N,p} - \nu_0^p\|\right] \leq \frac{\sqrt{2|\mathcal{X}|}}{\sqrt{N}}$$

using that $\mathbb{E}[||\mu - \nu||_1] \leq |\mathcal{S}|\mathbb{E}[||\mu - \nu||_2]$.
Induction step: assume now that (2.13) holds at time n. Using triangle inequality, at time $n+1$ we have, for any $p \in \mathcal{P}$,

$$\begin{aligned}
&\mathbb{E}\left[\|\nu_{n+1}^{N,p} - \nu_{n+1}^p\|\right] \leq \\
&\leq \mathbb{E}\left[\|\nu_{n+1}^{N,p} - \bar{F}(\nu_n^{N,p}, p_n(\nu_n^{N,p}), \epsilon_{n+1}^0)\|\right] + \mathbb{E}\left[\|\bar{F}(\nu_n^{N,p}, p_n(\nu_n^{N,p}), \epsilon_{n+1}^0) - \nu_{n+1}^p\|\right]
\end{aligned}$$

where we recall the expression of $\bar{F}$ described by (2.9).
For the second term, by continuity property of $\bar{F}$and $p(\nu)$ we can write :

$$\begin{aligned}
&\mathbb{E}\left[\|\bar{F}(\nu_n^{N,p}, p_n(\nu_n^{N,p})) - \nu_{n+1}^p\|\right] = \\
&= \mathbb{E}\left[\|\bar{F}(\nu_n^{N,p}, p_n(\nu_n^{N,p}), \epsilon_{n+1}^0) - \bar{F}(\nu_n^p, p_n(\nu_n^p), \epsilon_{n+1}^0)\|\right] \leq \\
&\leq \mathbb{E}\left[C(\epsilon_{n+1}^0)(\|\nu_n^{N,p} - \nu_n^p\| + |p_n(\nu_n^{N,p}) - p_n(\nu_n^p)|)\right] \\
&\leq \mathbb{E}\left[K(\|\nu_n^{N,p} - \nu_n^p\| + L_p\|\nu_n^{N,p} - \nu_n^p\|)\right] \\
&= K(1+L_p)\mathbb{E}\left[\|\nu_n^{N,p} - \nu_n^p\|\right] \leq K(1+L_p)^{n+1}\mathbb{E}[||\nu_0^{N,p} - \nu_0^p||] \\
&\leq K(1+L_p)^{n+1}\frac{\sqrt{2|X|}}{\sqrt{N}}
\end{aligned}$$

where we used that by induction step assuming that for every n, $C(\epsilon_{n+1}^0)$ is bounded by the constant K independent of $p \in \mathcal{P}$.
For the first term we have:

$$\begin{aligned}
&\mathbb{E}\left[\left\|\nu_{n+1}^{N,p} - \bar{F}(\nu_n^{N,p}, p_n(\nu_n^{N,p}), \epsilon_{n+1}^0)\right\|\right] = \\
&= \mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^N \delta_{Y_{n+1}^{i,\boldsymbol{\alpha}}} - \bar{F}(\nu_n^{N,p}, p_n(\nu_n^{N,p}), \epsilon_{n+1}^0)\right\|\right] = \\
&= \mathbb{E}\left[\left|\sum_{y\in\mathcal{S}}\frac{1}{N}\sum_{i=1}^N \delta_{Y_{n+1}^{i,\boldsymbol{\alpha}}}(y) - \bar{F}(\nu_n^{N,p}, p_n(\nu_n^{N,p}), \epsilon_{n+1}^0)(y)\right|\right] = \\
&= \sum_{y\in\mathcal{S}}\mathbb{E}\left[\left|\frac{1}{N}\sum_{i=1}^N \delta_{Y_{n+1}^{i,\boldsymbol{\alpha}}}(y) - \bar{F}(\nu_n^{N,p}, p_n(\nu_n^{N,p}), \epsilon_{n+1}^0)(y)\right|\right] = \\
&= \sum_{y\in\mathcal{S}}\mathbb{E}\left[\mathbb{E}\left[\left|\frac{1}{N}\sum_{i=1}^N \delta_{Y_{n+1}^{i,\boldsymbol{\alpha}}}(y) - \bar{F}(\nu_n^{N,p}, p_n(\nu_n^{N,p}), \epsilon_{n+1}^0)(y)\right|\middle|\boldsymbol{Y}_n^p\right]\right] \\
&= \sum_{y\in\mathcal{S}}\mathbb{E}\left[\mathbb{E}\left[\left|\frac{1}{N}\sum_{i=1}^N \delta_{Y_{n+1}^{i,\boldsymbol{\alpha}}}(y) - \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^N \delta_{Y_{n+1}^{i,\boldsymbol{\alpha}}}(y)\middle|\boldsymbol{Y}_n^p\right]\right|\middle|\boldsymbol{Y}_n^p\right]\right] \leq \frac{\sqrt{2|X|}}{\sqrt{N}}
\end{aligned}$$

by the LLN, where again the bound is independent of $p \in \mathcal{P}$. So we have obtained by induction that:

$$\sup_{p\in\mathcal{P}} \mathbb{E}\left[\|\nu_n^{N,p} - \nu_n^p\|\right] \leq (K(1+L_p)^n + 1)\frac{\sqrt{2|X|}}{\sqrt{N}}$$

for every time step $n = 0, \dots, T$. □

Then we are ready to state the following theorem:

Theorem 2 (ε-approximate optimality for finite agent model)**.** *If p^* is the optimal policy for the MFOS problem and $\hat{p}$ is the optimal policy for the N-agent problem (when all the agents have to use the same policy), then: as $N \to +\infty$,*

$$J^N(p^*, \dots, p^*) - J^N(\hat{p}, \dots, \hat{p}) \to 0.$$

with rate of convergence $\mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$ (the precise bound is given in the proof).

Proof. We can write:

$$J^N(p^*, \dots, p^*) - J^N(\hat{p}, \dots, \hat{p}) = \Big(J^N(p^*, \dots, p^*) - J(p^*)\Big) + \Big(J(p^*) - J(\hat{p})\Big) + \Big(J(\hat{p}) - J^N(\hat{p})\Big)$$

Notice first that we can bound this term simply deleting the second term in the r.h.s noticing $J(p^*) - J(\hat{p}) \leq 0$ since p^* is optimal for the *mean field* cost $J(p)$. For the first term we can write:

$$\begin{aligned}
&J^N(p^*, \dots, p^*) - J(p^*) = \\
&= \mathbb{E}\left[\sum_{n=0}^{T} \Psi(n, \nu_n^{N,p^*}, p_n^*(\nu_n^{N,p^*})) + g(\mu_T^{N,p^*})\right] - \sum_{n=0}^{T} \Psi(n, \nu_n^{p^*}, p_n^*(\nu_n^{p^*})) + g(\mu_T^{p^*}) = \\
&= \sum_{n=0}^{T} \mathbb{E}\left[\Psi(n, \nu_n^{N,p^*}, p_n^*(\nu_n^{N,p^*})) - \Psi(n, \nu_n^{p^*}, p_n^*(\nu_n^{p^*})) + g(\mu_T^{N,p^*}) - g(\mu_T^{p^*})\right] \leq \\
&\leq \sum_{n=0}^{T} \mathbb{E}\left[L_\Psi(\left\|\nu_n^{N,p^*} - \nu_n^{p^*}\right\| + \left|p_n^*(\nu^{N,p^*}) - p_n^*(\nu_n^{p^*})\right|) + L_g\|\mu_T^{N,p^*} - \mu_T^{p^*}\|\right] \leq \\
&\leq \sum_{n=0}^{T} \mathbb{E}\left[L_\Psi(1+L_p)\left\|\nu_n^{N,p^*} - \nu_n^{p^*}\right\| + L_g\|\mu_T^{N,p^*} - \mu_T^{p^*}\|\right] \leq \\
&\leq (L_\Psi(1+L_p) + L_g)T \sup_{n\in\{0,\dots,T\}} \sup_{p\in\mathcal{P}} \mathbb{E}\left[\left\|\nu_n^{N,p^*} - \nu_n^{p^*}\right\|\right]
\end{aligned}$$

by Lemma 1. For the last term $J(\hat{p}) - J^N(\hat{p})$ we can apply the same argument that we just described. In the following way we obtain:

$$J^N(p^*, \dots, p^*) - J^N(\hat{p}, \dots, \hat{p}) \leq (L_\Psi(1+L_p) + L_g)T\left[1 + K^T(1+L_p)^T\right]\frac{\sqrt{2|X|}}{\sqrt{N}}$$

□

Chapter 3

Dynamic Programming

Our motivation for developing a dynamic programming principle (DPP) for our formulation comes from both the literature and numerical purposes. Dynamic programming (DP) appears very often in the literature, encompassing fields such as economics, control theory, finance, development of computer programs to the ability of a computer to master the game of chess, Go, and many others. In the control theory of a dynamic system in particular, it has been studied and used very often to find solutions to a given optimization problem. Moreover, implementing an algorithm that founds on DPP often leads to precise optimal solutions that perform better than other methods. The main idea of the DPP is to decompose a multi-level defined problem to a smaller sub-problem. Specifically, in our setting, the decision to stop the agent at a time interval $[0, T]$ will be decomposed into a binary decision at each time recursively: "if we know an optimal decision from time $m = 1$ onward, then the problem reduces to the decision of whether or not to stop the agent at time $m = 0$."

3.1 Generale case

We introduce the dynamical form of the social cost (2.10) as:

$$
\begin{aligned}
V_n(\nu) &:= \inf_{p\in\mathcal{P}_{n,T}} J(p(x),\nu) \\
&:= \inf_{p\in\mathcal{P}_{n,T}} \mathbb{E}^0\Bigg[\sum_{(x,a)\in\mathcal{S}}\sum_{n=0}^{T} f(n,x,\nu^{n,\nu,p}_{X,n})a(1-p_n(x))\nu^{n,\nu,p}_n(x,a) + \\
&+\ \nu^{n,\nu,p}_n(x,a)\Phi(x,\nu^{n,\nu,p}_{X,n})ap_n(x) + g(\nu^{n,\nu,p}_{X,T})\Bigg],
\end{aligned}
\tag{3.1}
$$

where $\mathcal{P}_{n,T}$ is the set of all possible function $p : \{n, \dots, T\} \times \mathcal{X} \to [0, 1]$ and $\nu^{p,\nu,n}$ denotes the distribution of the process that starts at time n with a given distribution ν; it satisfies (2.8) but starting at time n instead of 0 with $\nu^{p,\nu,n}_n = \nu$.
The optimal value at time 0 will be denoted: $V^*(\nu) = V_0(\nu)$, which is also equal to $\inf_p J(p, \nu)$. With this definition we can now state and prove the following DPP.

Theorem 3 (Dynamic Programming Principle)**.** *For the dynamics given by* (2.6) *and the value function given by* (3.1) *the following dynamic programming principle holds:*

$$\begin{cases} V_n(\nu) = \inf_{h\in\mathcal{H}} \sum_{(x,a)\in\mathcal{S}} \nu(x,a)\Phi(x,\nu_X)ah(x) + \nu(x,a)f(n,x,\nu_X)a(1-h(x)) \\ \qquad\qquad + \mathbb{E}^0\left[V_{n+1}(\bar{F}(\nu,h,\epsilon^0_{n+1}))\right]. \\ V_T(\nu) = \sum_{(x,a)\in\mathcal{S}} \nu^p_T(x,a)\Phi(x,\nu^p_{X,T})a + g(\nu^p_{X,T}) \end{cases} \tag{3.2}$$

where ν_X is the first marginal of the distribution ν, i.e., $\nu_X(x) = \nu(x,0)+\nu(x,1)$ and $\mathbb{E}^0$ is the expectation over the realization of the common noise . The sequence of optimizers defines an optimal stopping decision that we will denote by $h^* : \{0,\ldots,T-1\}\times\mathcal{X}\times\mathcal{P}(\mathcal{S}) \to [0,1]$ and satisfies: for every $n \in \{0,\ldots,T-1\}$ and every $\nu \in \mathcal{P}(\mathcal{S})$, $V_n(\nu) = \sum_{(x,a)\in\mathcal{S}} \nu(x,a)\Phi(x,\nu_X)ah^*_n(x,\nu) + \nu(x,a)f(n,x,\nu_X)a(1-h^*_n(x,\nu)) + \mathbb{E}^0\left[V_{n+1}(\bar{F}(\nu,h^*_n(x,\nu),\epsilon^0_{n+1}))\right]$.

Proof. To prove this result, we will show that actually the mean field optimal stopping problem can be reduce to a mean field optimal control problem in discrete time and continuous space. Then we can apply the well-studied dynamic programming principle for mean field Markov decision processes (MFMDPs). We have:

$$\begin{aligned} V_n(\nu) := \inf_{p\in\mathcal{P}_{n,T}} \mathbb{E}^0\Bigg[&\sum_{(x,a)\in\mathcal{S}} \sum_{m=n}^{T} f(m,x,\nu^{n,\nu,p}_{X,m})a(1-p_m(x))\nu^{n,\nu,p}_m(x,a) + \\ &+ \nu^{n,\nu,p}_m(x,a)\Phi(x,\nu^{n,\nu,p}_{X,m})ap_m(x) + g(\nu^{n,\nu,p}_{X,T})\Bigg] = \\ = \inf_{p\in\mathcal{P}_{n,T}} \mathbb{E}^0 &\left[\sum_{m=n}^{T} \Psi(m,\nu^{n,\nu,p}_m,p_m) + g(\nu^{n,\nu,p}_{X,T})\right], \end{aligned}$$

where $\Psi : [|n,T|] \times \mathcal{P}(\mathcal{S}) \times \mathcal{P}_{n,T} \to \mathbb{R}$ and it is defined by,

$$\Psi(n,\nu,p) := \sum_{(x,a)\in\mathcal{S}} f(n,x,\nu_{X,n})a(1-p_n(x))\nu_n(x,a) + \nu_n(x,a)\Phi(x,\nu_{X,n})ap_n(x). \tag{3.3}$$

We can then define the process $Z^\nu := (Z^p_n)_{n=0,\ldots T}$ as $Z^p_0 = z := \nu$ and $Z^p_{n+1} := \bar{F}(Z^p_n, p_n, \epsilon^0_{n+1})$. We also denote as $Z^{p,1}_m := \nu^{n,\nu,p}_{X,m}$ the first marginal of this process. With this notation our value function can be written as:

$$V_n(\nu) := V_n(z) = \inf_{p\in\mathcal{P}_{n+1}} \mathbb{E}^0\left[\sum_{m=n}^{T} \Psi(m,Z^p_m,p_m) + g(Z^{p,1}_T)\right].$$

and we recognize a well studied control problem for which the DPP is:

$$V_n(z) = \inf_{h\in\mathcal{H}} \Psi(n,z,h) + \mathbb{E}^0\left[V_{n+1}(\bar{F}(z,h,\epsilon^0_{n+1})\right],$$

where $\mathcal{H}$ is the set of all functions $h : \mathcal{X} \to [0,1]$. Eventually we can conclude by proceeding backward, getting our initial notation. □

3.2 Synchronous Stopping Times

Actually we can show that this DPP still holds for a restricted class of randomized stopping times in which all the agents (regardless of their own state) have the same probability of stopping. Let $\tilde{\mathcal{P}}_{n,T}$ be the set of $p : \{0, \dots, T\} \to [0, 1]$. Notice that here p_n does not depend on the individual state x. At every time step $n = m$ every agent has the same probability to stop p_m, i.e for every $x \in \mathcal{X}$ at time $n = m$, $p_n(x) = p_n$. We call this set as *synchronous* stopping times. Let us define:

$$\tilde{V}_n(\nu) := \inf_{p \in \tilde{\mathcal{P}}_{n,T}} J(p, \nu) := \inf_{p \in \tilde{\mathcal{P}}_{n,T}} \mathbb{E}^0 \Bigg[\sum_{(x,a) \in \mathcal{S}} \sum_{m=n}^{T} f(m, x, \nu^{n,\nu,p}_{X,m}) a (1 - p_m) \nu^{n,\nu,p}_m (x, a) + \\ + \ \nu^{n,\nu,p}_m (x, a) \Phi(x, \nu^{n,\nu,p}_{X,m}) a p_m + g(\nu^{n,\nu,p}_{X,T}) \Bigg]$$

Then it is easy to extend our previous result described in Theorem 3.

Theorem 4 (Dynamic Programming Principle for Synchronous Stopping Times)**.** *For the setting of synchronous stopping times, the value function satisfies:*

$$\begin{cases} \tilde{V}_n(\nu) = \inf_{h \in [0,1]} \sum_{(x,a) \in \mathcal{S}} \nu(x, a) \Phi(x, \nu_X) a h + \nu(x, a) f(n, x, \nu_X) a (1 - h) + \mathbb{E}^0 \left[\tilde{V}_{n+1}(\bar{F}(\nu, h, \epsilon^0_{n+1})) \right] . \\ \tilde{V}_T(\nu) = \sum_{(x,a) \in \mathcal{S}} \nu^p_T (x, a) \Phi(x, \nu^p_{X,T}) a + g(\nu^p_{X,T}) \end{cases} \tag{3.4}$$

The proof follows the same argument as the one of Theorem 3 so we omit it.

Chapter 4

Deep Learning Methods: Theoretical Analysis

4.1 Algorithms

To address the MFOS problem numerically, we have two approaches based on two different formulations. As the most naive approach, we can attempt to directly minimize the mean-field social cost $J(p)$ stated in (2.10), where we optimize over all the possible stopping probability functions $p : \{0, \ldots, T\} \times \mathcal{X} \to [0, 1]$. A more ideal treatment is to leverage the Dynamic Programming Principle (DPP) discussed in Theorem 3 and solve for the optimal stopping probability using induction backward in time. For each of the timestep n, we implicitly learn the true value function $V_n(\nu)$ by solving the optimization problem in (3.2), where we search over all possible one-step stopping probability function $h : \mathcal{X} \to [0, 1]$ for each time n. We refer to the method of directly optimizing mean-field social cost as the direct approach (DA) and the attempt to solve MFOS via backward induction of the DPP approach.To alleviate the notations, we denote: $\bar{\Psi}(\nu, h) = \sum_{x \in \mathcal{X}} \nu(x, 1)\Psi(x, \nu_X)h(x)$, which represents the one-step mean field cost. In the code, `optim_up` denotes one update performed by the optimizer (e.g. Adam in our simulations). Pseudocodes are shown in algorithms 1 and 2

4.2 Convergence analysis of the algorithms

In this section we want to provide a convergence analysis of the estimator V_0^M of the value function V_n, where M represent the training sample size. Note that we are going to study the algorithm at the *mean field* level. In particular we are going to analyse the convergence of the DP-algorithm where at each step t we are going to approximate the optimal policy by our neural network keeping track of all the approximations we have already made in the interval $[t + 1, T]$ (we proceed backward in time). Our interest relies on how the finite size M of the training set affect the convergence of our algorithm, so we are going to assume that the optimizer find the exact argmin at each iteration. Our goal is to show that as the size of our training set goes to

Algorithm 1 Direct Approach for MFOS

Require: Time-dependent stopping decision neural network: $\psi_\theta : \{0, \dots, T\} \times \mathcal{X} \times \mathcal{P}(\mathcal{S}) \to [0, 1]$, cost function Ψ, mean-field dynamic transition $\bar{F}$, time horizon T, max training iteration $N_{\texttt{iter}}$.

for $k = 0, \dots, N_{\texttt{iter}} - 1$ **do**
    Uniformly sample initial distribution ν_0^p from the probability simplex on $\mathbb{R}^{2|\mathcal{X}|}$ and a common noise trajectory $(e_n^0)_{n=0}^{T-1}$
    for $n = 0, \dots, T - 1$ **do**
        $p_n(x) = \psi_\theta(x, \nu_n^p, n; \theta_k)$ for any $x \in \mathcal{X}$ ▷ Compute stopping probability
        $\ell_n = \sum_{x \in \mathcal{X}} \Psi(n, \nu_n^p, p_n)(x)$ ▷ Compute loss at time n
        $\nu_{n+1}^p = \bar{F}(\nu_n^p, p_n, e_n^0)$ ▷ Simulate MF dynamic
    end for
    Compute $\ell_T = g(\nu_{X,T}^p)$
    $\ell = \sum_{n=0}^{T} \ell_n$ ▷ Compute the total loss
    $\theta_{k+1} = \texttt{optimizer\_up}(\theta_k, \ell(\theta_k))$ ▷ AdamW optimizer step
end for
return $\psi_{\theta_{N_{\texttt{iter}}}}$

infinity our neural network can compute the exact optimal value.
Let us recall our mean field setting: our process is the distribution of the population at each time step $(\nu_n^p)_{n=0,\dots,T}$ (when we measure not only the proportion of people in a given state but also the proportion among them that has stopped or not) and it is controlled by the policy p that represent at each time step and each state the probability to stop in that state at that time; this process follow the dynamics given by (2.9) so, $\nu_{n+1}^p = \bar{F}(\nu_n^p, p_n, \epsilon_{n+1}^0)$; the social cost associated to this dynamics is $J(p) := \mathbb{E}^0 \left[\sum_{n=0}^{T} \Psi(n, \nu_n^p, p_n) + g(\nu_{X,T}^p) \right]$ where Ψ is defined in (3.3). We define the class of one-layer neural networks that are going to approximate the optimal policy as:

$$
\begin{aligned}
\mathcal{A}_M := \Bigg\{ & \nu \in \mathcal{P}(\mathcal{S}) \to A(\nu; \beta) = (A_1(\nu; \beta), \dots, A_q(\nu; \beta)) \in [0, 1]^{\mathcal{X}}, |\mathcal{X}| = q, \\
& A_i(\nu; \beta) = \sigma \left(\sum_{j=1}^{K} c_{ij} (a_{ij} \cdot \nu + b_{ij})_+ + c_{0j} \right), \qquad i = 1, \dots, q \\
& \beta = (a_{ij}, b_{ij}, c_{ij})_{i,j}, a_{i,j} \in \mathbb{R}^d, ||a_{ij}|| \le \eta, b_{ij}, c_{ij} \in \mathbb{R}, \sum_{i=0}^{K} c_{ij} \le \gamma \Bigg\}
\end{aligned}
\tag{4.1}
$$

where K represent the neurons and γ and η are usually referred to in the literature as respectively *total variation* and *kernel*. In this way we are working with a neural networks that have one hidden layer, rectified linear activation function and σ as output layer.
Using a one layer Neural Network (NN) denoted by $A(\nu; \beta) \in \mathcal{A}_M$ (where M is the size of the training sample) we are going approximate our optimal policy learning backward in time

Algorithm 2 Dynamic Programming Approach for MFOS

Require: A sequence of stopping decision neural network: $\psi_\theta^n : \mathcal{X} \times \mathcal{P}(\mathcal{S}) \to [0,1]$ for $n \in \{0, \ldots, T-1\}$, cost function Ψ, mean-field dynamic transition $\bar{F}$, time horizon T, max training iteration $N_{\texttt{iter}}$.

Set $\psi_\theta^T = 1$ since all distribution stopped at time T.
for $n = T-1, \ldots, 0$ **do** ▷ Train backward in time
  for $k = 0, \ldots, N_{\texttt{iter}} - 1$ **do**
    Uniformly sample initial distribution ν_n^p from the probability simplex on $\mathbb{R}^{2|\mathcal{X}|}$
    for $m = n, \ldots, T$ **do**
      if $m = n$ **then**
        $p_m(x) = \psi_\theta^m(x, \nu_m^p; \theta_k^n)$ ▷ Compute with NN for current time
      else
        $p_m(x) = \psi_\theta^m(x, \nu_m^p; \theta^{m,*})$ ▷ Compute with trained NN from future time
      end if
      $\ell_m = \sum_{x \in \mathcal{X}} \Psi(m, \nu_m^p, p_m)(x)$ ▷ Compute loss at time m
      $\nu_{m+1}^p = \bar{F}(\nu_m^p, p_m, e_n^0)$ ▷ Simulate MF dynamic
    end for
    Compute $\ell_T = g(\nu_{X,T}^p)$
    $\ell = \sum_{m=n}^T \ell_m$ ▷ Compute the total loss from time n to T
    $\theta_{k+1}^n = \texttt{optimizer\_up}(\theta_k^m, \ell(\theta_k^n))$ ▷ AdamW optimizer step
  end for
  Set $\theta^{n,*} = \theta_{N_{\texttt{iter}}}^n$ ▷ Stored trained weight
end for
return $(\psi_{\theta_{N_{\texttt{iter}}}^n}^n)_{n=0}^{T-1}$

our parameter β, in the following sense: at every time step n we keep track of all approximated optimal policies $\hat{p}_k$, that we have already approximated, for $k = n+1, \dots, T-1$ and we search for

$$\hat{\beta}_n \in \operatorname*{argmin}_{\beta} \mathbb{E}^0 \left[\Psi(n, \nu_n, A(\nu_n; \beta)) + \hat{Z}^A_{n+1} \right], \tag{4.2}$$

where $\nu_n \sim \rho_n$ (sample training distribution) and we denoted by $\hat{Z}^A_{n+1} := \sum_{k=n+1}^{T-1} \Psi(k, \hat{\nu}^\beta_k, \hat{p}^\beta_k(\hat{\nu}^\beta_k)) + g(\hat{\nu}^\beta_{X,T})$ with $\hat{\nu}^\beta_{n+1} = \bar{F}(\nu_n, A(\nu_n; \beta), \epsilon^0_{n+1})$ and $\hat{\nu}^\beta_{k+1} = \bar{F}(\hat{\nu}^\beta_k, \hat{p}^\beta_k(\hat{\nu}^\beta_n), \epsilon^0_{n+1})$ for every $k = n+1, \dots, T-1$.

Given an estimate $\hat{p}_k$, $k = n+1, \dots, T-1$ the approximated policy $\hat{p}_n$ is estimated by using a training sample $(\nu^{(m)}_n, (\epsilon^{(m),0}_{k+1})^{T-1}_{k=n})$, for $m = 0, \dots, M$ for simulating the trajectory and optimizing β of $\mathcal{A}(\cdot, \beta)$ by a stochastic Gradient Descent method (ADAMW). We are going to denote $(\hat{p}^M_k)^{T-1}_{k=n}$ to underline the fact that we are approximating the optimal policy using a training sample of size M. The estimated value function is defined by:

$$\hat{V_n}^M := \mathbb{E}_M \left[\sum_{k=n}^{T-1} \Psi(k, \hat{\nu}^{n,\nu}_k, \hat{p}^M_k(\hat{\nu}^{n,\nu}_k)) + g(\hat{\nu}^{n,\nu}_{X,T}) \right] =: J_n^{(\hat{p}^M_k)^{T-1}_{k=n}}, \tag{4.3}$$

where $\mathbb{E}_M$ is the expectation conditioned on the training set used for computing $(\hat{p}^M_k)_k$, and $(\hat{\nu}^{n,\nu}_k)_k$ is driven by the estimated optimal controls.

In practice at every time steps we generate a training sample for $\nu^{(m)}_n$, $m = 1, \dots, M$ and samples for the common noise $(\epsilon^{(m)}_k)^T_{k=n+1}$ for $m = 1, \dots, M$, then we consider the policy

$$\hat{p}_n \in \operatorname*{argmin}_{A \in \mathcal{A}_M} \hat{J}^{A,(\hat{p}^M_k)}_{n,M} := \operatorname*{argmin}_{A \in \mathcal{A}_M} \frac{1}{M} \sum_{m=1}^{M} \left[\Psi(n, \nu^{(m)}_n, A(\nu^{(m)}_n)) + \hat{Z}^{(m),A}_{n+1} \right]$$

where $\hat{J}^{A,(\hat{p}^M_k)}_{n,M}$ is the empirical cost function and $\hat{p}_n := A(\cdot; \hat{\beta}_n)$. So we are computing and approximation of $\hat{p}$ using Stochastic Gradient Descent (SGD) method.

The analysis of the convergence of a similar algorithm has been studied by **?**. Their framework is not mean field and therefore differs form ours in several aspects. For that reason we cannot apply their result as a "black-box" but we are going to prove it from scratch. Note that our initial purpose is to solve the N-agent problem so finally we want to ensure that the assumptions on it bring to "good" formulation of the mean field problem.

Assumption 5. *We are going to make assumptions on the transition probability, the dependence on the initial distribution, the cost function, the dynamics and the neural network.*

(H-1) *- Dynamics: for the convergence of the algorithm we introduce a perturbed dynamics, defining the role of the common noise in the following sense: at every time step the system evolve according to $F(n, x, \mu, e)$, a transition function without the presence of the common noise. After that the whole mass (stopped and non stopped) is perturbed by the common noise. Another property that we want to achieve a continuity property of the mean filed dynamics*

$\bar{F}$ and this derive from a corresponding property of the N-agent setting in which $Y^{p,i}_{n+1} = D(n, Y^{p,i}_n, \nu^{N,p}_n, p_n, \epsilon^i_{n+1}, \epsilon^0_{n+1})$ as described in (2.11). We assume that

$$|D(n, y, \nu^N, p, e, e^0) - D(n, y^1, \nu^{1,N}, p^1, e, e^0)| \leq \tilde{C}(e)C(e^0)(|y-y^1| + \|\nu^N - \nu^{1,N}\| + \|p-p^1\|)$$

with $\tilde{C}(e) \leq \tilde{K}$ for every e and $C(e^0) \leq K$ for every e^0.
This property brings to the continuity property of $\bar{F}$ described in Assumptions 1.

***(H-2)** - Transition probability: In the setting of assumption **H-1**, . In particular :*

$$P^p(\nu, d\nu') = r(\nu, p; \nu')\rho(d\nu')$$

where $P^p(\nu, d\nu')$ is the transition kernel of our system and

$$\begin{aligned} & r(\nu, p; \nu') \leq \|r\|_\infty \leq \infty \qquad \forall \nu, \nu' \in \mathcal{P}(\mathcal{S}), \forall p \in \mathcal{P} \\ & |r(\nu_1, p_1; \nu') - r(\nu_2, p_2; \nu')| \leq L_r(\|\nu_1 - \nu_2\| + |p_1 - p_2|) \qquad \forall \nu_1, \nu_2, \nu' \in \mathcal{P}(\mathcal{S}), \forall p_1, p_2 \in \mathcal{P}_{0,T}. \end{aligned} \tag{4.4}$$

Then we want that our sampling distribution has a bigger support than ρ in order to have for every $f : \mathcal{P}(\mathcal{S}) \to \mathbb{R}$ bounded and Lipschitz function and for every control $p \in \mathcal{P}$ the following inequality:

$$\int \int f(\nu') P^p(\nu, d\nu') \eta_{sampl}(d\nu') \leq K \int f(\nu') \eta_{sampl}(d\nu')$$

where η_{sampl} is the training distribution used to sample.

***(H-3)** - Cost function: as mentioned in Assumption 1 we have the Lipschitz property and the boundness derive from the compactness of the state space and the control space. Notice that these properties are not strictly related to the mean field model since the cost function are the same as in the finite agent framework.*

***(H-4)** - Neural network : Recalling the definition of the class of neural network that we are going to work with, defined in (4.1) we assume the following:*

$$K, \eta, \gamma \xrightarrow[M \to \infty]{} \infty \qquad \gamma^{T-1}\eta^{T-2}\sqrt{\frac{\log(M)}{M}} \xrightarrow[M \to \infty]{} 0$$

Remark 1. *Assumption **H-3** permits us to have a proper random dynamics for at least every initial distribution. In particular, this means that it is possible for an agent who has stopped to be "reborn" and continue to spread. Thus, it is possible for an agent to receive multiple stop costs each time it decides to stop.*

Remark 2. *Assumption **H-3** assumes that our transition probability (at mean field level) has a density with respect to a distribution ρ with bounded and L_r-Lipschitz density (notice that, at every time step we have a sample distribution for $\nu^{(m)}_n$ and a sample distribution for $(\epsilon^{(m)}_{k+1})^{T-1}_{k=n}$).*

Before state the main theorem of this section regarding the convergence of the value we now give a regularity property of the value. In particular we have,

Proposition 1. *Under the Assumptions 5, for every $n = 0, \dots, T$ the value function V_n is bounded and Lipschitz continuous and for every $n = 0, \dots, T$ the optimal control $p_n^* \in \mathbb{L}^1(\rho)$, where ρ is the training sample distribution.*

Proof. Due to the assumption on the running and the terminal cost we can write:

$$
\begin{aligned}
V_n(\nu) := \inf_{p \in \mathcal{P}_{n,T}} \mathbb{E}^0 \left[\sum_{m=n}^{T} \Psi(m, \nu_m^{n,\nu,p}, p_m) + g(\nu_{X,T}^{n,\nu,p}) \right] \leq \\
\leq (T-n)\|\Psi\|_\infty + \|g\|_\infty
\end{aligned}
$$

for every $n = 0, \dots, N$ and for every initial distribution ν. For the Lipschitz property, for every initial distirbution $\nu, \mu \in \mathcal{P}(\mathcal{X})$ and for every control p we have,

$$
\begin{aligned}
|J_n(p,\nu) - J_n(p,\mu)| &= \left| \mathbb{E}^0 \left[\sum_{m=n}^{T} \Psi(m, \nu_m^{n,\nu,p}, p_m) + g(\nu_{X,T}^{n,\nu,p}) \right] - \mathbb{E}^0 \left[\sum_{m=n}^{T} \Psi(m, \nu_m^{n,\mu,p}, p_m) + g(\mu_{X,T}^{n,\mu,p}) \right] \right| \\
&\leq \left| \mathbb{E}^0 \left[L_\Psi \sum_{m=n}^{T} \left(\|\nu_m^{n,\nu,p} - \mu_m^{n,\mu,p}\| + |p_m(\nu_m^{n,\nu,p}) - p_m(\nu_m^{n,\nu,p})| \right) + L_g \|\nu_{X,T}^{n,\nu,p} - \mu_{X,T}^{n,\mu,p}\| \right] \right| \\
&\leq \left| \mathbb{E}^0 \left[L_\Psi \sum_{m=n}^{T} \left(\sup_{l \in [|n,T|]} \|\nu_l^{n,\nu,p} - \mu_l^{n,\mu,p}\| + L_p \sup_{l \in [|n,T|]} \|\nu_l^{n,\nu,p} - \mu_l^{n,\nu,p}\| \right) + L_g \sup_{l \in [|n,T|]} \|\nu_{X,l}^{n,\nu,p} - \mu_{X,l}^{n,\mu,p}\| \right] \right| \\
&\leq \left| L_\Psi (T-n) \Big(C\|\nu - \mu\| + L_p C \|\nu - \mu\| \Big) + L_g C \|\nu - \mu\| \right| \\
&\leq \Big(L_\psi (T-n) C (1 + L_p) + L_g C \Big) \|\nu - \mu\|.
\end{aligned}
$$

Notice that here we used the Lipschitz property of the cost functions and the control function with respect to the measure and then the dependency of the initial value express by:

$$
\mathbb{E}^0 \left[\sup_{l \in [|n,T|]} \|\nu_l^{n,\nu,p} - \mu_l^{n,\mu,p}\| \right] \leq C\|\nu - \mu\|
$$

where μ_l and ν_l are driven by the same dynamics $\bar{F}$ but with different initial distribution. This last inequality can be easily proved by induction noticing that

$$
\mathbb{E}^0 \left[\|\nu_1^{n,\nu,p} - \mu_1^{n,\mu,p}\| \right] = \mathbb{E}^0 \left[\|\bar{F}(n,\nu,p,\epsilon^0) - \bar{F}(n,\mu,p,\epsilon^0)\| \right] \leq C(\epsilon^0)||\mu - \nu||.
$$

Taking the infumum over $\mathcal{P}_{n,T}$ we end up with the Lipschitz property for the value function. Regarding the L^1 property of the optimal control we can first argue by a measurable selection theorem that for every $n = 0, \dots, T, p_n^* : \mathcal{P}(\mathcal{S}) \to [0,1]^{\mathcal{X}}$ can be chosen measurable with respect to ρ. Finally, from the finiteness of space

$$
\int_{\mathcal{P}(S)} \|p_n^*(\nu)\| d\rho(\nu) < \infty.
$$

□

Theorem 6 (Convergence of the value). *When assumptions 5 holds we obtain the following result on the convergence of the value:*

$$\mathbb{E}\left[\hat{V}_n^M(\nu_n) - V_n(\nu_n)\right] = \mathcal{O}\Bigg(\frac{\gamma^{T-n-1}\eta^{T-n-2}}{\sqrt{M}} + \sup_{n\le k\le T-1}\inf_{A\in\mathcal{A}_M}\mathbb{E}\left[|A(\nu_k) - p_k^*(\nu_k)|\right]\Bigg) \tag{4.5}$$

where $\mathbb{E}$ stands for the expectation over the training set used to evaluate the approximated optimal policies $(\hat{p}_k^M)_{n\le k\le T-1}$, as well as the path $(\nu_n)_{n\le k\le T}$ controlled by the latter.

Proof. We can write,

$$\begin{aligned}\mathbb{E}_M\left[\hat{V}_n^M(\nu_n) - V_n(\nu_n)\right] &= \mathbb{E}_M\left[\hat{V}_n^M(\nu_n)\right] - \inf_{A\in\mathcal{A}_M}\mathbb{E}_M\left[J_n^{A,(\hat{p}_k^M)_{k=n+1}^{T-1}}(\nu_n)\right]\\ &+ \inf_{A\in\mathcal{A}_M}\mathbb{E}_M\left[J_n^{A,(\hat{p}_k^M)_{k=n+1}^{T-1}}(\nu_n)\right] - \mathbb{E}_M\left[V_n(\nu_n)\right]\\ &= \mathbb{E}_M\left[\hat{V}_n^M(\nu_n)\right] - \hat{J}_{n,M}^{(\hat{p}_k^M)_{k=n}^{T-1}} + \hat{J}_{n,M}^{(\hat{p}_k^M)_{k=n}^{T-1}} - \inf_{A\in\mathcal{A}_M}\mathbb{E}_M\left[J_n^{A,(\hat{p}_k^M)_{k=n+1}^{T-1}}(\nu_n)\right]\\ &+ \inf_{A\in\mathcal{A}_M}\mathbb{E}_M\left[J_n^{A,(\hat{p}_k^M)_{k=n+1}^{T-1}}(\nu_n)\right] - \mathbb{E}_M\left[V_n(\nu_n)\right]\end{aligned} \tag{4.6}$$

where $\hat{J}_{n,M}^{A,(\hat{p}_k^M)_{k=n+1}^{T-1}} := \frac{1}{M}\sum_{m=1}^M[\psi(n,\nu_n^{(m)}, A(\nu_n^{(m)})) + \hat{Z}_{n+1}^{(m),A}]$, the empirical cost function from n to T, associated with the sequence of controls $(A, (\hat{p}_K^M)_{k=n+1}^{T-1})$ and training set, where $\hat{Z}_{n+1}^A$ is defined in (4.2).

Step 1: let us analyse the first therm of the r.h.s of this inequality. We have,

$$\begin{aligned}\mathbb{E}_M\left[\hat{V}_n^M(\nu_n)\right] - \hat{J}_{n,M}^{(\hat{p}_k^M)_{k=n}^{T-1}} &= \mathbb{E}_M\left[J_n^{(\hat{p}_k^M)_{k=n}^{T-1}}(\nu_n)\right] - \hat{J}_{n,M}^{(\hat{p}_k^M)_{k=n}^{T-1}}\\ &\le \epsilon_n^{\text{esti}}.\end{aligned} \tag{4.7}$$

where

$$\epsilon_n^{\text{esti}} := \sup_{A\in\mathcal{A}_M}\left|\hat{J}_{n,M}^{A,(\hat{p}_k^M)_{k=n+1}^{T-1}} - \mathbb{E}_M\left[J_n^{A,(\hat{p}_k^M)_{k=n+1}^{T-1}}(\nu_n)\right]\right| \tag{4.8}$$

is the *estimation error* at time n associated with the algorithm.
Moreover, for any $A\in\mathcal{A}_M$,

$$\hat{J}_{n,M}^{A,(\hat{p}_k^M)_{k=n+1}^{T-1}} - \mathbb{E}_M\left[J_n^{A,(\hat{a}_k^M)_{k=n+1}^{T-1}}(\nu_n)\right] \le \epsilon_n^{\text{esti}}$$

Recalling that $\hat{p}_n^M = \text{argmin}_{A\in\mathcal{A}_M}\hat{J}_{n,M}^{A,(\hat{p}_k^M)_{k=n+1}^{T-1}}$ we can take the infimum over $\mathcal{A}_M$ we get,

$$\hat{J}_{n,M}^{(\hat{p}_k^M)_{k=n}^{T-1}} - \inf_{A\in\mathcal{A}_M}\mathbb{E}_M\left[J_n^{A,(\hat{p}_k^M)_{k=n+1}^{T-1}}(\nu_n)\right] \le \epsilon_n^{\text{esti}}$$

Pluggin this inequality into (4.7) we obtain the following:

$$\mathbb{E}_M\left[\hat{V}_n^M(\nu_n)\right] - \inf_{A\in\mathcal{A}_M}\mathbb{E}_M\left[J_n^{A,(\hat{p}_k^M)_{k=n+1}^{T-1}}(\nu_n)\right] \le 2\epsilon_n^{\text{esti}}. \tag{4.9}$$

Step 2: let us proceed to the analysis of the second term of the equality (4.6). Using the tower property for J_n and the dynamic programming principle for V_n, as stated in theorem 3, with the optimal control p_n^* at time n, we can write

$$\begin{aligned}
&\inf_{A\in\mathcal{A}_M}\mathbb{E}_M\left[J_n^{A,(\hat{p}_k^M)_{k=n+1}^{T-1}}(\nu_n)\right] - \mathbb{E}_M[V_n(\nu_n)] \\
&\qquad = \epsilon_n^{\text{approx}} + \inf_{A\in\mathbb{A}^{\mathcal{X}}}\mathbb{E}_M\left\{\Psi(\nu_n, A(\nu_n)) + \mathbb{E}_n^A\left[J_{n+1}^{(\hat{p}_k^M)_{k=n+1}^{T-1}}(\nu_{n+1})\right]\right\} \\
&\qquad - \mathbb{E}_M\left[\Psi(\nu_n, p_n^*(\nu_n)) + \mathbb{E}_n^{p_n^*}[V_{n+1}(\nu_{n+1})]\right] \\
&\qquad \le \epsilon_n^{\text{approx}} + \mathbb{E}_M\mathbb{E}_n^{p_n^*}\left[J_{n+1}^{(\hat{p}_k^M)_{k=n+1}^{T-1}}(\nu_{n+1}) - V_{n+1}(\nu_{n+1})\right],
\end{aligned}$$

where

$$\epsilon_n^{\text{approx}} := \inf_{A\in\mathcal{A}_M}\mathbb{E}_M\left[J_n^{A,(\hat{p}_k^M)_{k=n+1}^{T-1}}(\nu_n)\right] - \inf_{A\in\mathbb{A}^{\mathcal{X}}}\mathbb{E}_M\left[J_n^{A,(\hat{p}_k^M)_{k=n+1}^{T-1}}(\nu_n)\right] \tag{4.10}$$

is the *approximation error* at time n (notice that the first term is always bigger than the second one since $\mathcal{A}_M \subset \mathbb{A}^{\mathcal{X}}$). It measures how well the regression function can be approximated by means of neural networks function in $\mathcal{A}_M$ ($\mathbb{A}^{\mathcal{X}}$ is the set of Borelian functions from the state space $\mathcal{X}$ into the control space $\mathbb{A} := [0,1]^{\mathcal{X}}$; note that the class of neural network is not dense in the set $\mathbb{A}^{\mathcal{X}}$). The notation $\mathbb{E}_n^A[\cdot]$ stands for the expectation conditioned by ν_n at time n and the training set, when the action A is followed at time n. Now let us assume that $\nu_n \in \mathcal{P}(\mathcal{S}^*)$ and $p_n^* \in \mathcal{P}_{n,T}^*$, i.e we are in the framework where the assumptions on the transition probability (4.4) are satisfied and therefore we can write,

$$\begin{aligned}
&\inf_{A\in\mathcal{A}_M}\mathbb{E}_M\left[J_n^{A,(\hat{p}_k^M)_{k=n+1}^{T-1}}(\nu_n)\right] - \mathbb{E}_M[V_n(\nu_n)] \\
&\qquad \le \epsilon_n^{\text{approx}} + \|r\|_\infty \int [\hat{J}_{n+1}^{(\hat{p}_k^M)_{k=n+1}^{T-1}}(\nu') - V_{n+1}(\nu')]\rho(d\nu') \\
&\qquad = \epsilon_n^{\text{approx}} + \|r\|_\infty\mathbb{E}_M\left[\hat{V}_{n+1}^M(\nu_{n+1}) - V_{n+1}(\nu_{n+1})\right]
\end{aligned}$$

with $\nu_{n+1} \sim \rho$.
Step 3: using the previous steps and the first decomposition (4.6), we have

$$\begin{aligned}
&\mathbb{E}_M\left[\hat{V}_n^M(\nu_n) - V_n(\nu_n)\right] = \mathbb{E}_M\left[\hat{V}_n^M(\nu_n)\right] - \inf_{A\in\mathcal{A}_M}\mathbb{E}_M\left[J_n^{A,(\hat{p}_k^M)_{k=n}^{T-1}}(\nu_n)\right] \\
&+ \inf_{A\in\mathcal{A}_M}\mathbb{E}_M\left[J_n^{A,(\hat{p}_k^M)_{k=n}^{T-1}}(\nu_n)\right] - \mathbb{E}_M\left[V_n(\nu_n)\right] \\
&\le 2\epsilon_n^{\text{estim}} + \epsilon_n^{\text{approx}} + \|r\|_\infty\mathbb{E}_M\left[\hat{V}_{n+1}^M(\nu_{n+1} - V_{n+1}(\nu_{n+1})\right]
\end{aligned}$$

By induction we simply have

$$\mathbb{E}_M\left[\hat{V}_n^M(\nu_n) - V_n(\nu_n)\right] \leq \sum_{k=n}^{T-1}(2\epsilon_k^{\text{esti}} + \epsilon_k^{\text{approx}}). \tag{4.11}$$

Our goal now is to derive the behaviour of these two errors when the sample size goes to infinity. In particular we rely on the following lemma:

Lemma 2 (Convergence of the errors)**.** *For $n = 0, \ldots, T$ the following holds as $M \to \infty$*

$$\mathbb{E}[\epsilon_n^{esti}] = \mathcal{O}\left(\frac{\gamma^{T-n-1}\eta^{T-n-2}}{\sqrt{M}}\right) \tag{4.12}$$

$$\mathbb{E}[\epsilon_n^{approx}] = \mathcal{O}\left(\frac{\gamma^{T-n-1}\eta^{T-n-2}}{\sqrt{M}} + \sup_{n\leq k\leq T-1}\inf_{A\in\mathcal{A}_M}\mathbb{E}[|A(\nu_k) - p_k^*(\nu_k)|]\right) \tag{4.13}$$

Proof. First result: the order of convergence of the estimation error can be prove the same as Lemma 4.10 of Hure et. al (2021). The main idea is to use a copy of the process ν_n and an additional randomness using random signs. *Second result:* For the second result we can follow Hure et. al paying attention when our framework slightly differ from their formulation. Here we are going to give all the main ideas highlighting the differences with their method. In particular let $(\hat{p_k}^M)_{k=n+1}^{T-1}$ be the sequence of the estimated controls at time $k = n+1, \ldots, T-1$. The cost function associated with A is characterized by the following Bellman equation:

$$\begin{cases} J_N^{A,(\hat{p}_k^M)_{k=n+1}^{T-1}}(\nu) = g(\nu) \\ J_n^{A,(\hat{p}_k^M)_{k=n+1}^{T-1}}(\nu) = \Psi(n,\nu,A(\nu)) + \mathbb{E}_{n,\nu}^A\left[J_{n+1}^{A,(\hat{p}_k^M)_{k=n+1}^{T-1}}(\nu_{n+1})\right]. \end{cases} \tag{4.14}$$

Adding and subtracting $\mathbb{E}[V_n(\nu_n)]$ we can show that

$$\epsilon_n^{\text{approx}} \leq \inf_{A\in\mathcal{A}_M}\mathbb{E}_M[J_n^{A,(\hat{p}_k^M)_{k=n+1}^{T-1}}(\nu_n)] - \mathbb{E}[V_n(\nu_n)],$$

and then we can apply DPP to obtain

$$\begin{aligned} &\min_{A\in\mathcal{A}_M}\mathbb{E}_M[J_n^{A,(\hat{p}_k^M)_{k=n+1}^{T-1}}(\nu_n)] - \mathbb{E}[V_n(\nu_n)] \\ &\leq \inf_{A\in\mathcal{A}_M}\mathbb{E}_M\left[\Psi(n,\nu_n,A(\nu_n)) + \mathbb{E}_n^A\left[J_{n+1}^{(\hat{p}_k^M)_{k=n+1}^{T-1}}(\nu_{n+1})\right]\right] - \\ &- \mathbb{E}\left[\Psi(n,\nu_n,p^*(\nu_n)) + \mathbb{E}_n^{p^*}[V_{n+1}(\nu_{n+1})]\right] \end{aligned}$$

Then, for all the distributions and the controls, using assumption (4.4), we can write

$$\begin{aligned} &\mathbb{E}_M\left[\Psi(n,\nu_n,A(\nu_n)) + \mathbb{E}_n^A\left[J_{n+1}^{(\hat{p}_k^M)_{k=n+1}^{T-1}}(\nu_{n+1})\right]\right] - \mathbb{E}\left[\Psi(n,\nu_n,p^*(\nu_n)) + \mathbb{E}_n^{p^*}[V_{n+1}(\nu_{n+1})]\right] \\ &\leq (L_\Psi + \|V_{n+1}\|L_r)\mathbb{E}[\|A(\nu_n) - p^*(\nu_n)] + \\ &\|r\|_\infty \inf_{A\in\mathcal{A}_M}\mathbb{E}_M\left[J_{n+1}^{A,(\hat{p}_k^M)_{k=n+2}^{T-1}}(\nu_{n+1}) - V_{n+1}(\nu_{n+1})\right] + 2\|r\|_\infty\epsilon_{n+1}^{\text{esti}}. \end{aligned}$$

Pluggin this last inequality in the previous we obtain

$$\mathbb{E}\left[\inf_{A\in\mathcal{A}_M}\mathbb{E}_M\left[J_n^{A,(\hat{p}_k^M)_{k=n+1}^{T-1}}(\nu_n)\right]-\mathbb{E}[V_n(\nu_n)]\right]$$
$$=\mathcal{O}\Bigg(\sup_{n+1\le k\le T-1}\mathbb{E}[\epsilon_k^{\text{esti}}]+\sup_{n\le k\le T-1}\inf_{A\in\mathcal{A}_M}\mathbb{E}[|A(\nu_n)-p^*(\nu_n)|]\Bigg).$$

Now we use the first result of this lemma regarding the order of convergence of the expected value of the estimation error to complete the proof. □

Putting all the elements together, we have shown that

$$\mathbb{E}_M\left[\hat{V}_n^M(\nu_n)-V_n(\nu_n)\right]\le\sum_{k=n}^{T-1}(2\epsilon_k^{\text{esti}}+\epsilon_k^{\text{approx}})=$$
$$=\mathcal{O}\Bigg(\frac{\gamma^{T-n-1}\eta^{T-n-2}}{\sqrt{M}}+\sup_{n\le k\le T-1}\inf_{A\in\mathcal{A}_M}\mathbb{E}\left[|A(\nu_k)-p_k^*(\nu_k)|\right]\Bigg).$$

□

The first of the rate of convergence should be seen as the estimation error due to the approximation of the optimal controls by means of neural networks in $\mathcal{A}_M$ using the empirical cost functional $J_{n,M}^{A,(\hat{p}_k^M)}$. Notice that we can conclude that our algorithm converge when the second term of the convergence in (4.5) goes to 0 as M goes to infinity.

Proposition 2. *In the previous setting we have:*

$$\sup_{n\le k\le T-1}\inf_{A\in\mathcal{A}_M}\mathbb{E}\left[|A(\nu_k)-p^*(\nu_k)|\right]\xrightarrow{M\to\infty}0 \tag{4.15}$$

Furthermore if we assume the optimal control p_k^* is $c-Lipschitz$ for $k=n,\dots,T-1$, then we also have the following rate of convergence,

$$\sup_{n\le k\le T-1}\inf_{A\in\mathcal{A}_M}\mathbb{E}\left[|A(\nu_k)-p^*(\nu_k)|\right]\le c\left(\frac{\gamma}{c}\right)^{-2d/(d+1)}\log\left(\frac{\gamma}{c}\right)+\gamma K^{-(d+3)/(2d)}, \tag{4.16}$$

where $d:=|\mathcal{P}(\mathcal{X})|$, the dimension of the mean field state space and γ is defined is (4.1).

Proof. The proof rely on Proposition 4.1 Hure et al (2021) but, despite their approach, we have proved a regularity property of the optimal control (proposition 1) instead of assuming it. In particular, in order to apply their result, we need at least $p^*\in\mathbb{L}^1(\rho)$ where ρ is the distribution defined in 5 and we have proven it in 1. □

Chapter 5

Experiments

In this section we provide three examples of increasing complexity. We want to give value to our methods on situations that increasingly approach a real-world dimension. The first example serves as a test to show the validity of our algorithms and in doing so solves a task that at this level can still be solved by the human mind quite easily. The second example is a two-dimensional extension of the previous one and tests the response of the algorithms when the dimensionality of the problem increases. These finally lead to the main example we tested in this thesis, which simulates the "intelligent" formation of a flock of drones. Starting from a chaotic situation, the set of robots must form a target distribution decided to their own. The goal is then to stop the correct portion of the population at the right time. Such a task is not easily solved by the human mind and as the size increases solving such a problem becomes prohibitive. We then demonstrate the great impact our method can have for this type of problem. Each example is solved by both methods described in the previous section, and the results show the evolution of the stoppered and unstopped mass at each instant, the strategy (the probability of stoppering agents in each state) at each instant by the central planner, and the results after training the training error and testing error.

5.1 Architecture

In this section, we will described the architecture of our Neural Network. For the direct approach, the neural network takes an input time t, while for the DPP approach, the neural network does not need time input.

In general, our neural network has the following structure. Our neural network takes an input pair (x, t), where x is the spatial input, t is the time. If t is a needed input, then it is passed through a module to generate a standard sinusoidal embedding and then fed to 2 fully connected layers with Sigmoid Linear Unit (SiLU) and generate an output t_{out}. Spatial input x is passed through an MLP with k residual blocks, each containing 4 linear layers with hidden dimension D and SiLU activation. This generates an output y_{out}. Our final output out is computed through,

$$\text{out} = \text{Outmod}(\text{GroupNorm}(y_{\text{out}} + t_{\text{out}}))$$

where Outmod is an out module that consists of 3 fully connected layers with hidden dimension D and SiLU activation, $\mathrm{GroupNorm}$ stands for group normalization. If t is not a needed input, then set $t_{\text{out}} = 0$. For all the test cases we have experimented with, we use $k = 3$, $D = 128$ for all the 1D experiments and $k = 5, D = 256$ for the 2D experiments.

5.2 Experiments

5.2.1 Towards the Uniform 1D

In this first example we are going to consider only a stopping cost, i.e every agent pays a cost only when he decided to stop. In particular we are going to solve the following example: there are 5 stations and all the agents start from the first one; at every time step they move to the right unless they are in the last station where they must stay there; the goal of this example is that stopping an agent has a cost related to how much people are in the same station (avoiding stooping in a crowd station). The mathematical details are expressed below. We take state space $\mathcal{X} = \{0, 1, 2, 3, 4\}$, time horizon $T = 4$, transition function $F(n, x, \mu, \epsilon) = x + 1$, with absorbing boundary at $x = 4$ (meaning that once at 4, the agent does not move anymore), and as stopping cost function $\Phi(x, \mu) = \mu(x)$ which depends on the mean field only through the state of the agent (this is sometimes called local dependence). For the testing distribution, we take a distribution concentrated on state $x = 0$, denoted as $\mu_0 = \delta_0$. It can be seen that the optimal strategy consists in spreading the mass to make it as close as uniform as possible (hence the name of this example). First, we explain how the optimal value is computed. Since the agents move deterministically to the right, the only option to freeze some mass at a state x is to do it at time n. It can be seen that: for every $n = 0, \ldots, T$ and for every $x \in \mathcal{X}$, we want to have $p_n(x = n) = \frac{1}{T+1-n}\mathbb{1}_{x=n}$ for $n < T$ and $p_n(x) = 1$ for $n = T$. Actually notice that for all $x \neq n$ the choice of p_n is arbitrary so, at every time-step n we can apply the same p_n for every state x. This brings us to optimize over the set of synchronous stopping times. Indeed it is optimal to stop someone before the time horizon T, otherwise we are going to pay $\mu(x) = 1$ at the end and the best way to do it is to spread the distribution over the state space. Then we can compute the optimal value and obtain: $V^{*,\delta_0} := \frac{T+2}{2(T+1)}$.

Direct Approach Results: Figure 5.1 shows the results of the experiments for the Direct Approach. The first set of images shows the evolution of the mass (the red one describes the mass that was stopped while the blue one refers to the mass that was not stopped) from the beginning of the problem to the end, and the final state is shown as the last image. It can be seen that a uniform distribution of states was achieved at the end. Also note that, as shown in the second set of images, the decision to stop a certain percentage of mass is independent of whether or not there is mass in that specific state. In fact, at time 0 the optimal decision is to stop all agents in state 4 even if there is no mass. This ensures robustness with respect to the initial distribution. In the last images the training loss and the testing loss are shown. The red line in the second image is the optimal loss computed theoretically.

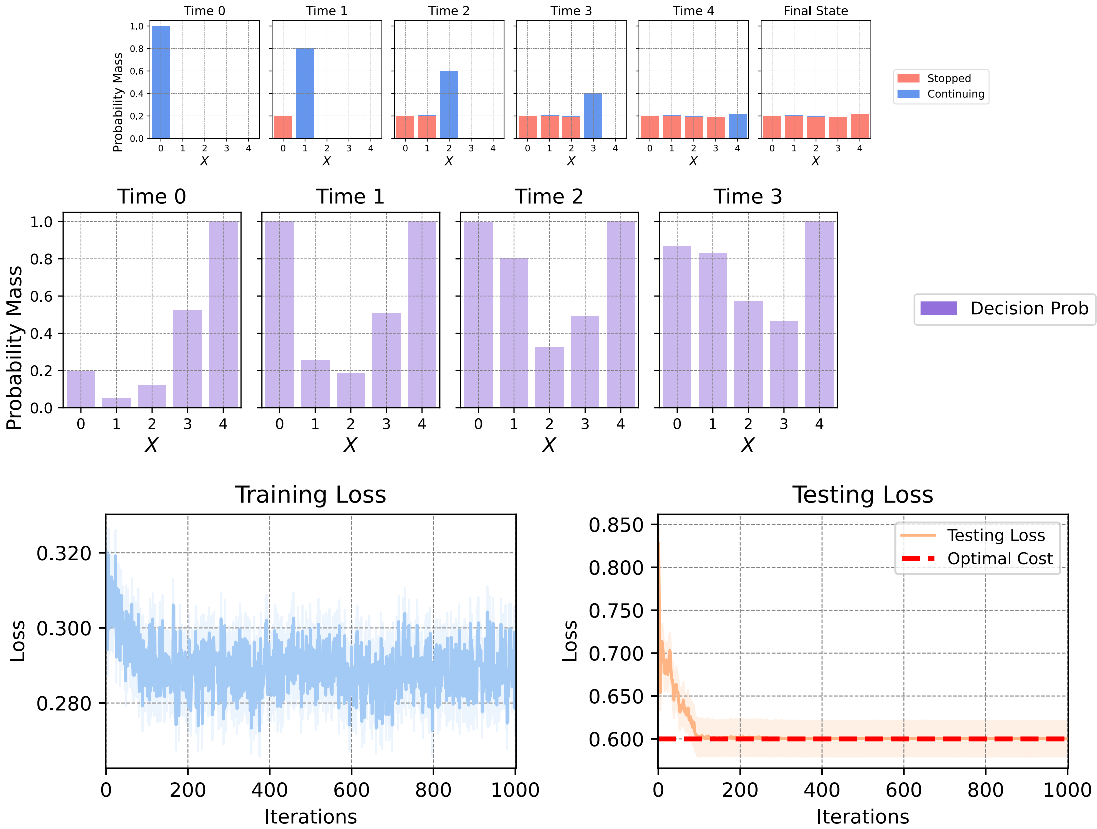

Figure 5.1: Example 1. DA results. Top: Evolution of the distribution and stopping probability at every time step. Bottom: training and testing losses after training.

Dynamic Programming Results: Figure 5.2 shows the results of the experiments when using the DPP approach for our algorithm. The results regarding the evolution of the mass are similar while we can see a difference in the decision probability. Then we plot the testing and the training loss for every time step.

The two methods in this example are comparable and both obtained excellent performance comparing with the benchmark value computed theoretically. This first example is fundamental to show how the neural network learn the optimal policies and obtain the optimal value.

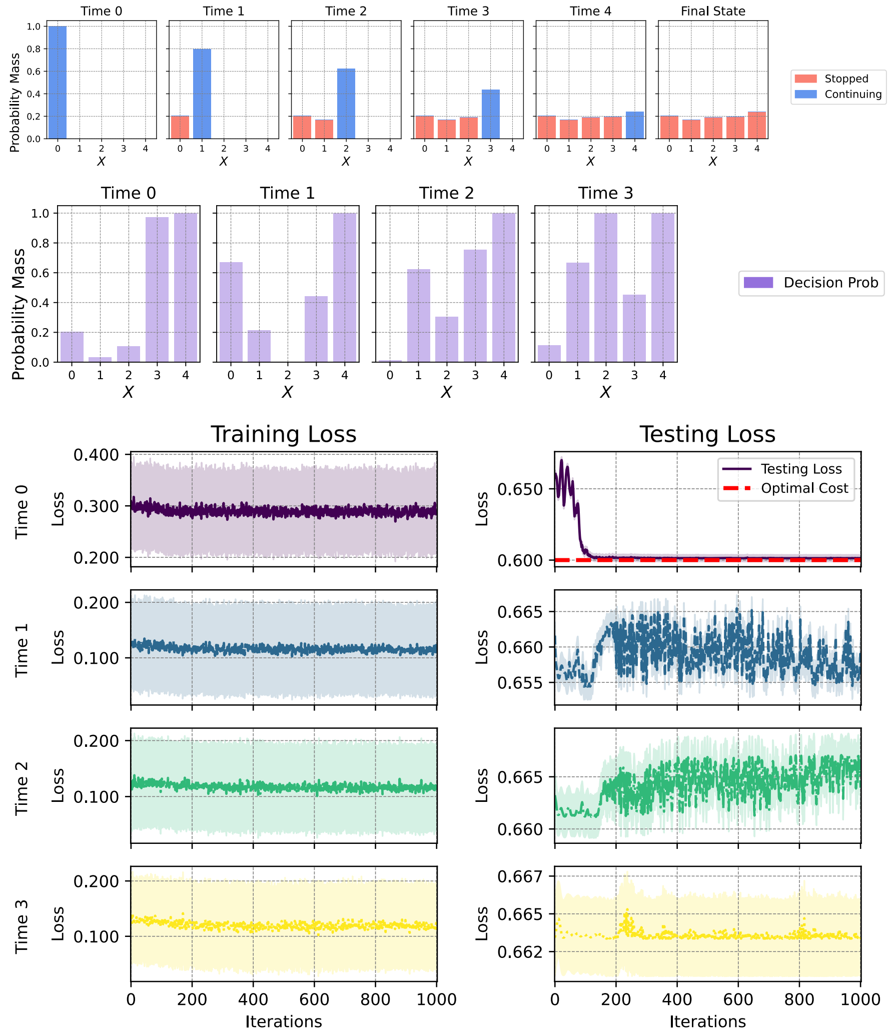

Figure 5.2: Example 1. DPP results. Top:Evolution of the distribution and stopping probability at every time step. Bottom: training and testing losses after training.

5.2.2 Towards the uniform 2D

In this example we extend the previous framework to a 2-dimension grid. We take state space $\mathcal{X} = \{0,1,2,3,4\} \times \{0,1,2,3,4\}$, time horizon $T = 4$, transition function $F(n, x, \mu, \epsilon) = x + (1, 0)$ which means that the agent deterministically moves to the state on the right on the same row, with absorbing boundary at $x = 4$, and cost function $\Phi(x, \mu) = \mu(x)$ which depends on the mean field only through the state of the agent (this is sometimes called local dependence). For the testing distribution, we take a distribution concentrated on state $x = 0$, denoted as $\mu_0 = 1/2\delta_{(0,0)} + 1/3\delta_{(0,2)} + 1/6\delta_{(0,1)}$. As in the 1D case, it can be seen that the optimal strategy consists in spreading the mass to make it as close as uniform as possible. However, the mass will not be uniform over the grid since rows have different masses.

Direct Approach Results: Figure 5.3 shows the results for the experiments conducted by the direct approach. Differently from before, we can see that the evolution of mass is described by a 2D heat map divided into two sectors. The top one shows the arrested mass in each grid state, while the bottom one describes the mass that has not yet been stopped. Also the optimal policy is described by a 2D heat map. No benchmark value is compute here.

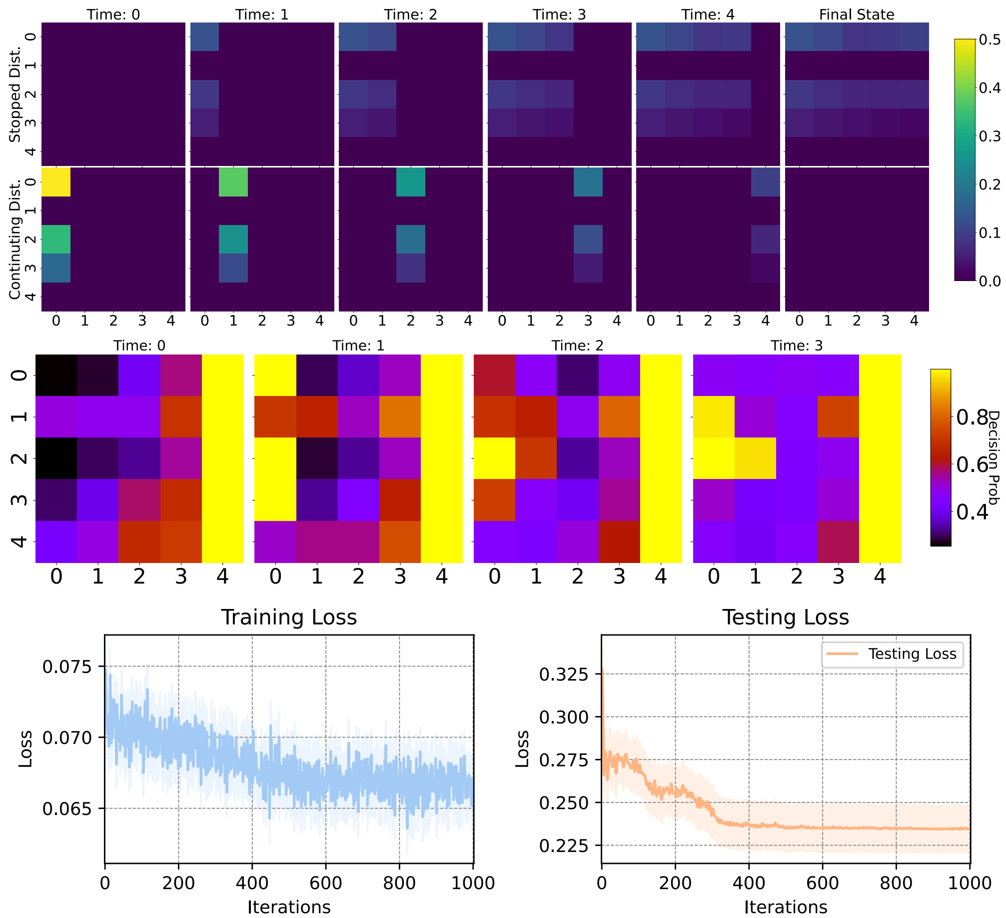

Figure 5.3: Example 2. DA results: Top -Evolution of the distribution and stopping probability at every time step. Bottom- training and testing losses after training.

Dynamics Programming Approach Results: Figure 5.4 shows the results for the experiments conducted by the dynamics programming approach. We can see that the final state is slightly different and it seems that the direct approach perform better on the task (getting towards a uniform distribution along the rows that has positive mass in the initial step). However, as you can see, in the last set of images of both the results the testing loss and the training loss are comparable meaning that even if the final configuration in the DPP approach seem to be far from the goal the cost is still close to the optimal one.

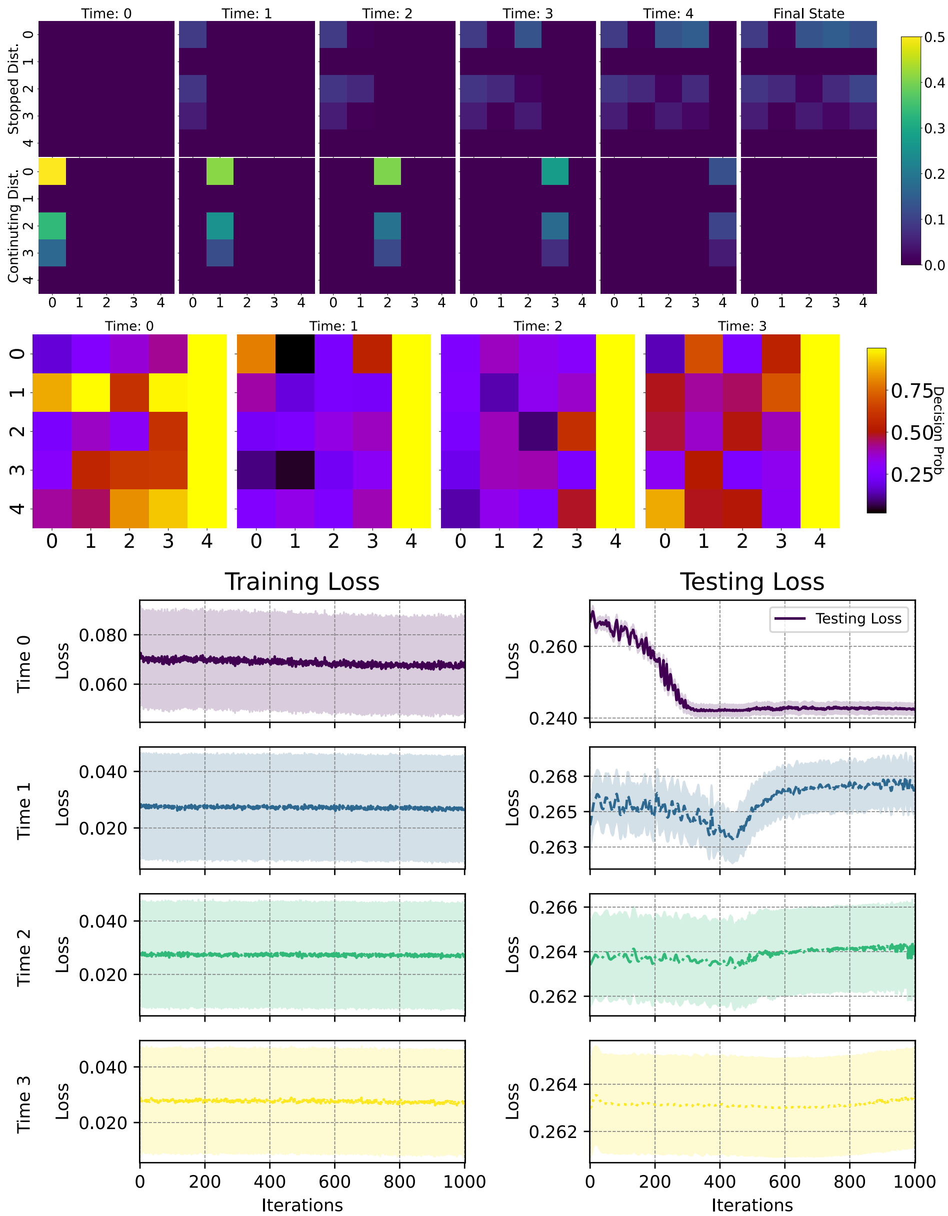

Figure 5.4: Example 2. DPP results: Top -Evolution of the distribution and stopping probability at every time step. Bottom- training and testing losses after training.

5.2.3 Drones Match the Target Distribution

This example want to explore a little bit more the potential of our method and try to model a real world scenario. We want to get able to let our drones start with a noisy random initial configuration and then during time reach the final desired configuration. In order to do this the cost at every time step will be represented by the distance between the detailed distribution at that time step and the target distribution that we want to achieve. Furthermore we implement a common noise dynamics. The introduction of the common noise let the problem be more realistic to real - world scenarios. For example we can think this common noise as an obstacle positioned in the second row of our grid that change position in the row at every time step. Then the neural network will be take that into account when learn the optimal policy. In details we take as a state space a 4×4 square grid $\mathcal{X} := \{0, 1, 2, 3\} \times \{0, 1, 2, 3\}$. The time horizon is set to $T = 20$. As a transition function, we take the uniform dynamic distribution over the neighbors in the following sense: given a state $x = (x_1, x_2)$ the transition to the next state is chosen with equal probability from the set $\{(x_1 + 1, x_2), (x_1 - 1, x_2), (x_1, x_2 + 1), (x_1, x_2 - 1)\}$ whenever all these states exist. When we consider a state on the boundary, the distribution is distributed over the possible neighbors. The common noise is taken uniformly in the second line of the grid at every time step. In other words it eliminate a possible neighbour in the second line at every time step. As mentioned above, we want to reach a desired target configuration, which we denote by ρ. We take as target distribution the letter "O" defined by $\rho = \frac{1}{8}(\delta_{0,1} + \delta_{0,2} + \delta_{1,0} + \delta_{2,0} + \delta_{3,1} + \delta_{3,2} + \delta_{2,3} + \delta_{1,3})$ (Figures 5.5 and 5.6) . To do this we construct as a social cost function the terminal cost $g_\rho(\mu) = ||\mu - \rho||_2^2$. So far, running and stopping costs have not been considered. As first step we introduce as initial distribution the uniform distribution over the simplex, i.e $\mu_0 \sim \mathcal{U}(\mathcal{P}(\mathcal{X}))$. This task is by no means trivial if you want to solve it "by hand," and this therefore shows the effectiveness of our method when scaling to very complex situations.

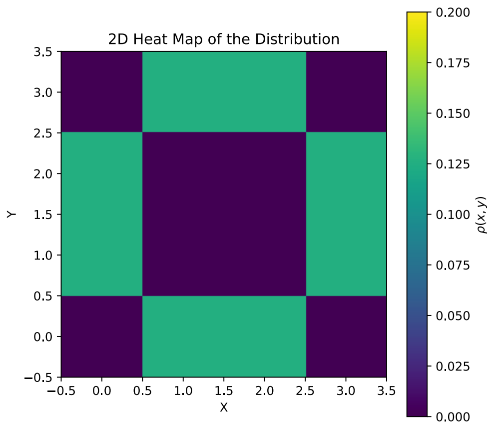

Figure 5.5: Heat map of the target distribution ρ

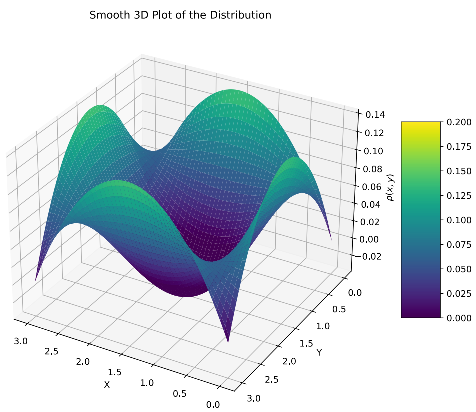

Figure 5.6: 3D plot of the target distribution ρ

Direct Approach Results: Figure 5.7 shows the results for the experiments conducted by the direct approach. We are showing only some steps of the evolution due to the limit space. It can be seen that in the second row we have a 0 mass square for every time step and this is linked to a presence of an obstacle (common noise). From a random initial distribution it can be seen that the algorithm perform well on creating the target distribution. Differently from before we now give also a view on the evolution of the 3D mass over time without separate form the stopped mass and the mass that has not been stopped yet. Note that since the state space is discrete we use a spline interpolation to give a better idea of what is happening. For a precise evolution of the mass we can see at the 2D heatmap. Note that the testing and the stopping loss are very close to zero since the mass is spread over several states and we have only a terminal cost that is defined by a L^2 norm.

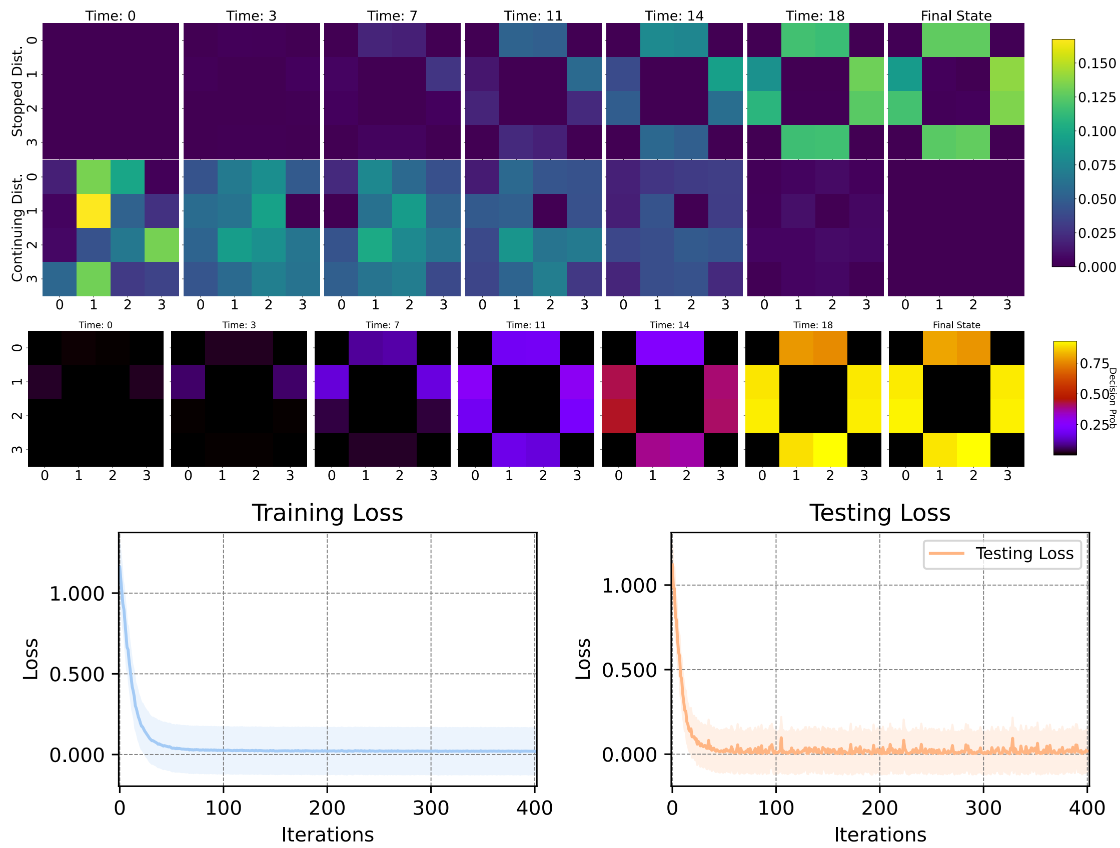

Figure 5.7: Example 3. DA results. Top: Evolution of the distribution and stopping probability at every time step. Bottom - training and testing losses after training.

Dynamic programming Results: Figure 5.9 shows the results for the experiments conducted by the dynamic programming approach. Due to the computational complexity and limited computational resources developing an efficient dynamics programming principle is not trivial. However we can see that our algorithm performs well on the task.

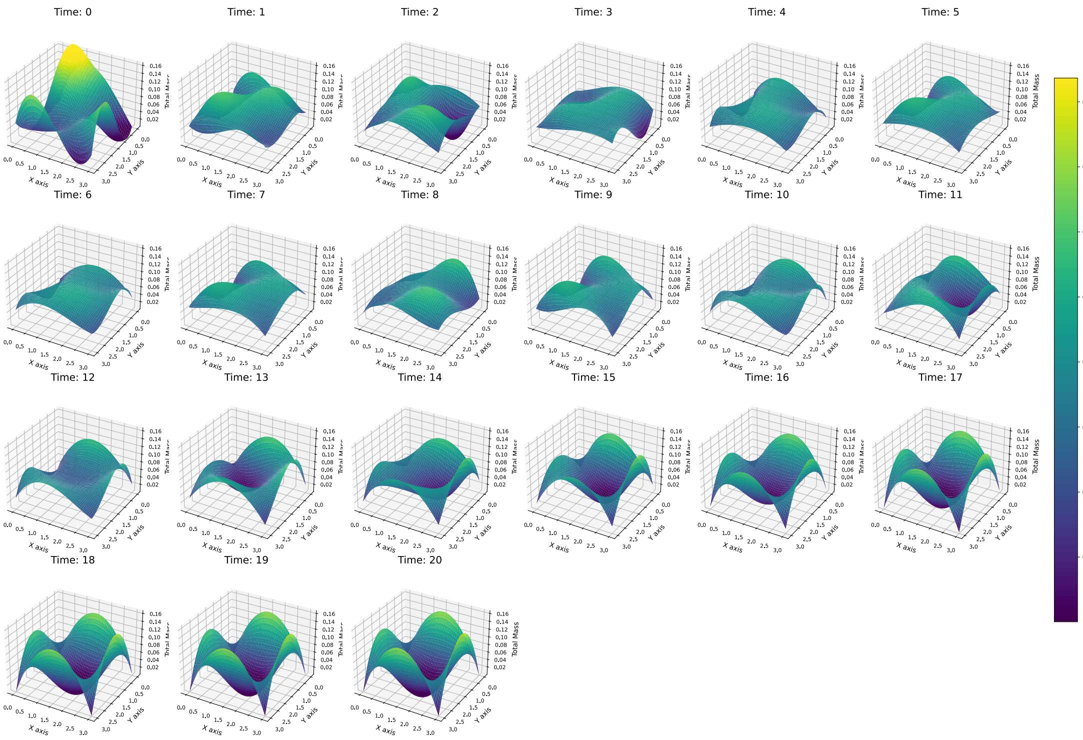

Figure 5.8: Example 3. Direct Approach: 3D Evolution of the mass

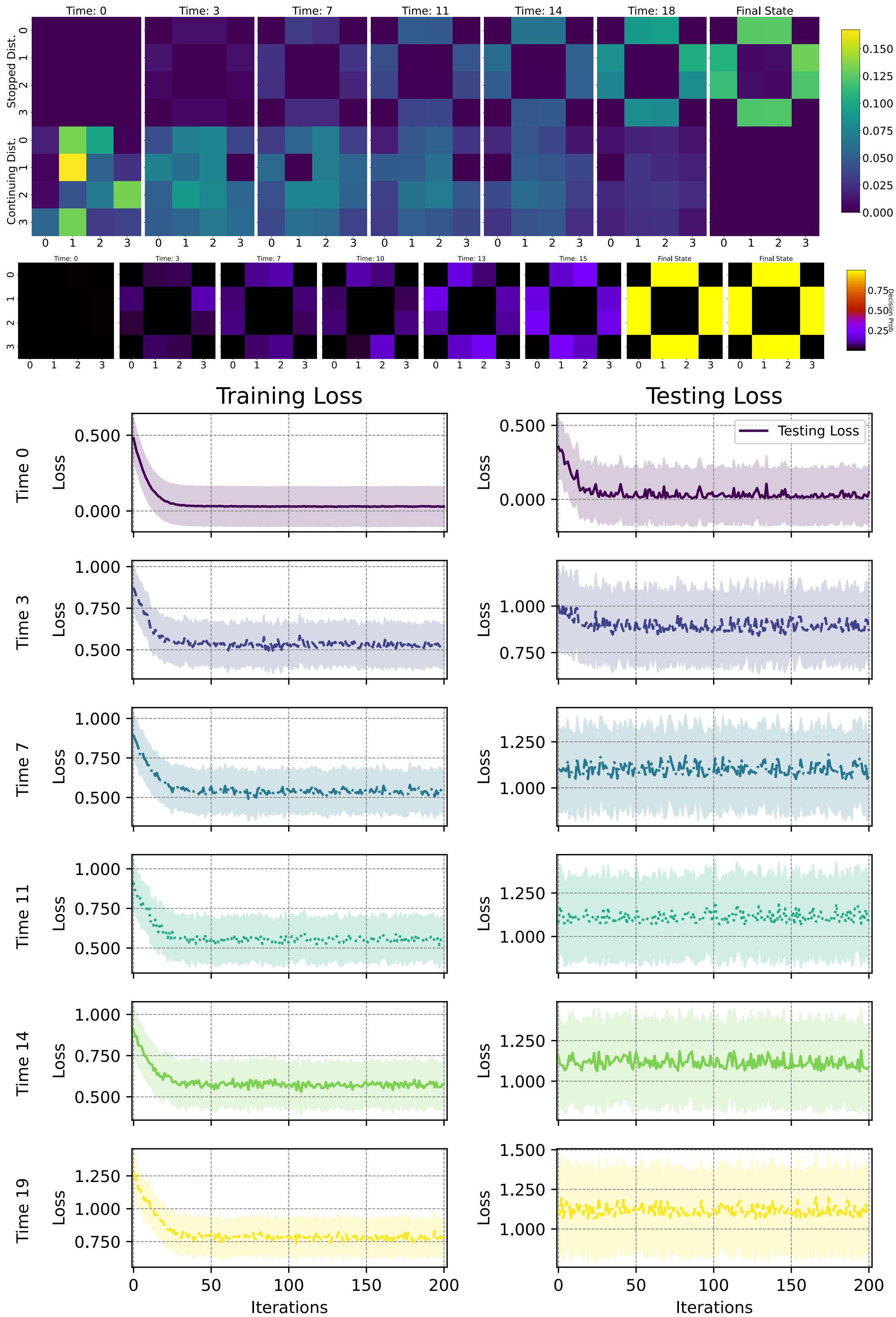

Figure 5.9: Example 3. DPP results. Top: Evolution of the distribution and stopping probability at every time step. Bottom- training and testing losses after training.

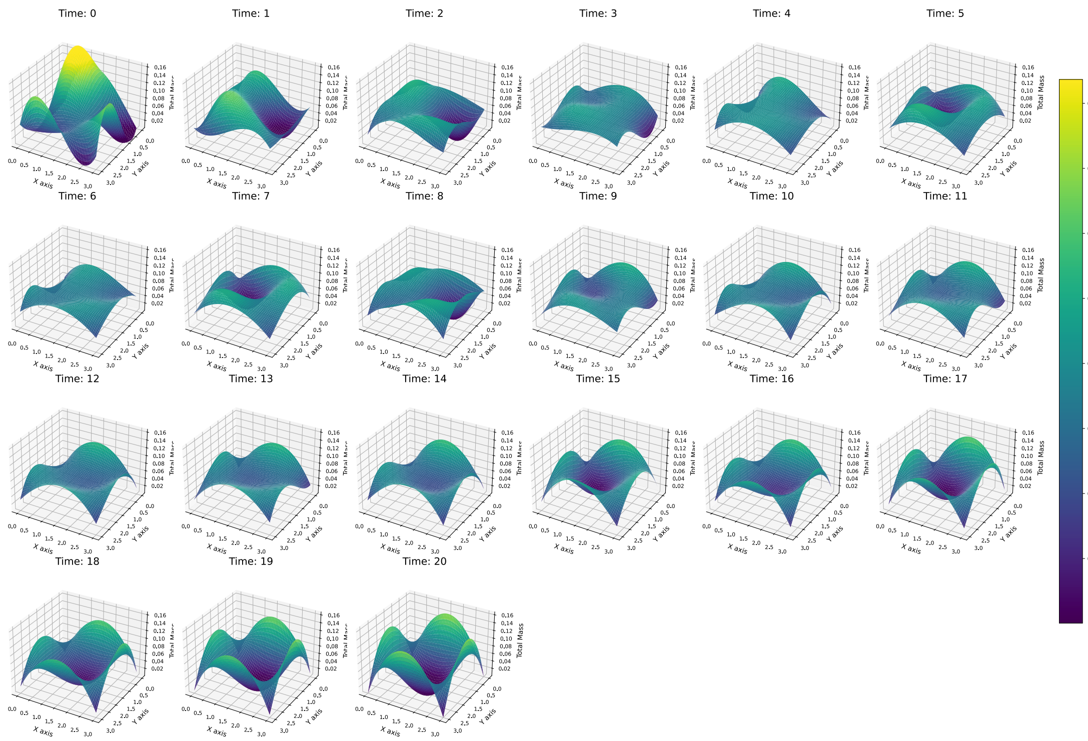

Figure 5.10: Example 3. DPP 3D Evolution of the mass

Conclusion

In this thesis we presented the problem of optimal stopping a regime mean field in discrete time and space. We generalized the pure decision space to randomized decisions. From what we know this is the first work in which such a framework is presented.

We motivated our mean field model by the finite agent environment and proved that an optimal solution in the former is approximation of the optimal solution of the latter (propagation of chaos and ϵ optimality approximation). This is a fundamental aspect since it allows, when the number of agents tends to infinity, to analyse the solution of the model at the mean-field regime

We also extended the cost function by developing our model with the presence of a running cost and a terminal cost in addition to the usual stopping cost present in optimal stopping problems. Of fundamental importance to this work is the introduction of an additional random variable that described the state (stopped or not) of each agent. With such an extended space our model becomes Markovian, a crucial aspect for the formulation of a dynamic programming principle. To demonstrate this principle, a pivotal aspect that should be emphasized was the reduction of our problem to a Mean Field Control problem. Such a reduction is not trivial but allows us to extend the techniques that can be used considering that the literature of MFC is more developed and richer than the Mean Field optimal stopping problems.

We have thus provided a dynamic programming principle for two classes of stopping times: asynchronous stopping and synchronous stopping. In the former case the central planner can decide to stop each individual agent at different times while in the latter the stopping of the entire population occurs at a single instant .

We furthermore provided two deep learning algorithms to solve several examples of increasing complexity. The first one builds on the principle of dynamic programming demonstrated in this paper while the second one attempts to directly minimize the mean field social cost sampling the whole trajectories and optimizing over all possible stopping probability functions. We performed an in-depth theoretical analysis of the algorithm, demonstrating its convergence in a simpler environment aware of the difficulties of having a general and detailed theoretical analysis of the algorithm.

Eventually we described three scenarios through three different examples of increasing complexity.

The last it meant to be a simulation of a real world application with the intelligent formation of a target distribution by a flock of drones

Future Works: There are several aspects that can be improved and be studied deeply.

- The different classes of stopping times analyzed can be extended
- Multiple real-world finite-agent examples can be studied and implemented in order to have a real application of our model
- The theoretical analysis of the algorithm can be done in a more general framework
- Different algorithms can be implemented and tested

However, we believe that this work opens horizons for the development of optimal stopping problems, which have gained increasing importance in recent years by modeling different situations and real-world applications.

Bibliography

Yves Achdou and Jean-Michel Lasry. Mean field games for modeling crowd motion. *Contributions to partial differential equations and applications*, pages 17–42, 2019.

Yves Achdou and Olivier Pironneau. *Computational methods for option pricing*. SIAM, 2005.

Vlad Bally and Gilles Pagès. A quantization algorithm for solving multidimensional discrete-time optimal stopping problems. *Bernoulli*, 9(6):1003–1049, 2003.

Sebastian Becker, Patrick Cheridito, and Arnulf Jentzen. Deep optimal stopping. *Journal of Machine Learning Research*, 20(74):1–25, 2019.

Alain Bensoussan, Jens Frehse, and Sheung Chi Phillip Yam. *Mean field games and mean field type control theory*. Springer Briefs in Mathematics. Springer, New York, 2013. ISBN 978-1-4614-8507-0; 978-1-4614-8508-7.

Tomas Björk. *Arbitrage theory in continuous time*. Oxford university press, 2009.

René Carmona and François Delarue. *Probabilistic theory of mean field games with applications. I*, volume 83 of *Probability Theory and Stochastic Modelling*. Springer, Cham, 2018. ISBN 978-3-319-56437-1; 978-3-319-58920-6. Mean field FBSDEs, control, and games.

René Carmona and Mathieu Laurière. Deep learning for mean field games and mean field control with applications to finance. *arXiv preprint arXiv:2107.04568*, 2021.

Trey D Crowther. Optimal stopping of multi-robot exploration for unknown, bounded environments. 2023.

Kai Cui, Sascha H Hauck, Christian Fabian, and Heinz Koeppl. Learning decentralized partially observable mean field control for artificial collective behavior. In *The Twelfth International Conference on Learning Representations*, 2024.

Massimo Fornasier and Francesco Solombrino. Mean-field optimal control. *ESAIM: Control, Optimisation and Calculus of Variations*, 20(4):1123–1152, 2014.

Haotian Gu, Xin Guo, Xiaoli Wei, and Renyuan Xu. Mean-field controls with Q-learning for cooperative MARL: convergence and complexity analysis. *SIAM Journal on Mathematics of Data Science*, 3(4):1168–1196, 2021.

Calypso Herrera, Florian Krach, Pierre Ruyssen, and Josef Teichmann. Optimal stopping via randomized neural networks. *Frontiers of Mathematical Finance*, pages 0–0, 2023.

Jon Kleinberg, Robert Kleinberg, and Sigal Oren. Optimal stopping with behaviorally biased agents: The role of loss aversion and changing reference points. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 681–682, 2021.

Magdalena Kobylanski, Marie-Claire Quenez, and Elisabeth Rouy-Mironescu. Optimal multiple stopping time problem. 2011.

Yong Liang and Bingchang Wang. Robust mean field social optimal control with applications to opinion dynamics. In *2019 IEEE 15th International Conference on Control and Automation (ICCA)*, pages 1079–1084, 2019. doi: 10.1109/ICCA.2019.8899655.

Steven A Lippman and John J McCall. The economics of job search: A survey. *Economic inquiry*, 14(2):155–189, 1976.

A Max Reppen, H Mete Soner, and Valentin Tissot-Daguette. Neural optimal stopping boundary. *arXiv preprint arXiv:2205.04595*, 2022.

Dinah Rosenberg, Eilon Solan, and Nicolas Vieille. Social learning in one-arm bandit problems. *Econometrica*, 75(6):1591–1611, 2007.

Albert N Shiryaev. *Optimal stopping rules*, volume 8. Springer Science & Business Media, 2007.

Mehdi Talbi, Nizar Touzi, and Jianfeng Zhang. From finite population optimal stopping to mean field optimal stopping. *arXiv preprint arXiv:2210.16004*, 2022.

Mehdi Talbi, Nizar Touzi, and Jianfeng Zhang. Dynamic programming equation for the mean field optimal stopping problem. *SIAM Journal on Control and Optimization*, 61(4):2140–2164, 2023.

Changfeng Wang, Santosh Venkatesh, and J Judd. Optimal stopping and effective machine complexity in learning. *Advances in neural information processing systems*, 6, 1993.