Wang_Tianhao_02107061.pdf

by Tianhao Wang

Submission date: 06-Sep-2022 12:30PM (UTC+0100)
Submission ID: 185742912
File name: Wang_Tianhao_02107061.pdf (4.99M)
Word count: 22134
Character count: 104802

Imperial College London

Department of Mathematics

Reinforcement Learning Provides Free-Lunch, but at What Cost?

A Decentralized Mean-Field Games Algorithm for Multi-agent Optimal Liquidation

Author: Tianhao Wang (CID: 02107061)

A thesis submitted for the degree of

MSc in Mathematics and Finance, 2021-2022

Declaration

The work contained in this thesis is my own work unless otherwise stated.

Acknowledgements

I want to thank my supervisor, Paul Bilokon, for his time and effort in my thesis. Further, many thanks to my secondary supervisor Eyal Neuman, who provided solid theoretical support and advice and helped make sense of my confusion in mean-field theory. Additionally, I want to thank the kind guidance from my personal tutor Damiano Brigo when I was frustrated. Lastly, I would like to thank my friend Ben Somner-Bogard and Raisu Shawa for encouraging me mentally and substantially throughout this thesis project. This project would not have been possible without their support.

Abstract

Due to enormous computational costs tremendously rising with the number of traders, multi-agent reinforcement learning algorithms have not been widely adopted in algorithmic and high-frequency trading. However, the widely-used mean-field theory in the literature that aggregates participants makes strong assumptions about a centralized system where all the agents are effectively indistinguishable and can access complete information. Furthermore, the existing literature only focuses on deriving the Nash equilibrium, which may not attain even with provable properties for existence and uniqueness.

To address these issues, we release these assumptions and attempt to propose a decentralized mean-field games (DMFG) model-free learning framework with minimal information available to agents within feasible computational costs. By design, each self-interested agent is myopic and chooses its optimal trading strategy without prior knowledge of the underlying dynamics and its opponent. This simulation-based decentralized learning algorithm offers a more reasonable approximation to the actual financial market. We examine our modified DMFG algorithm with the theoretical Nash equilibrium for three optimal liquidation problems consisting of 1) single agents, 2) trading crowds, and 3) multiple trading crowds distinct in risk aversion and market power.

The difference between the performance of the theoretical Nash equilibrium and DMFG equilibrium offers some interesting intuitions and insights we should be aware of when designing our reinforcement learning algorithms. Besides, we believe this paper also provides some exciting hypotheses to investigate for future research.

Contents

List of Figures

List of Tables

Introduction

Optimal liquidation is one of the most extensively studied problems in algorithmic and high-frequency trading. Such an optimal liquidation problem falls into the category of the classical portfolio optimization by Merton in 1975 [1], where the single agent seeks to maximize the expected profit by trading between a risky asset and a risk-free bank account, with more generalizations in 1992 [2]. Further, Cartea, Jaimungal and Penalva address the issue stemming from the potential price impact and the limited liquidity in the market within a short time interval, which suggests that the agent may need to consider spreading its order over time instead of putting its inventory as the market order all at once as well as a certain sense of urgency to get rid of these shares during the trading process. Many other examples and modifications including liquidation with and without temporary and/or permanent market price impacts as well as with limit and/or market orders can be found in his textbook [3].

While most of the literature on optimal execution focuses on the single agent setting and provides explicit optimal trading strategies, the optimal trading strategies may be deficient in practice as the interaction with other market participants is neglected (e.g., see the evidence of crowding in [4]). To this end, more and more researchers have extended their attention to the multi-agent system over the past couple of decades (e.g., see [5, 6, 7, 8]). In particular, Neuman and Voss studied the phenomenon of crowding in financial markets with exponentially decaying transient market price impact in [9]. An extended but simplified setting of the competition between major-minor agents [1] was investigated by Huang, Jaimungal, and Nourian in [10]. Further, a general form of the mean-field games between K sub-populations with different beliefs was proposed by Casgrain and Jaimungal [11]. Moreover, Casgrain, Ning and Jaimungal provided a Deep Q-learning algorithm to solve the Nash equilibria for the multi-agent setting [12].

Nevertheless, one of the critical limitations of the existing literature on high-frequency trading is that the Nash equilibrium is not necessarily attained even with provable properties for existence and uniqueness [13, 14]. In other words, reaching the Nash equilibrium relies heavily on the introspective-thinking assumption for *each* agent. Besides, strong assumptions including identical and homogeneous agents, global mean-field information (or full information [2]) are usually required to explicitly solve the mean-field games. Sometimes, the individual agents' are assumed to be negligible for simplifying the original problem, as we may see in [11]. These prompted us to shift our attention to reinforcement learning dynamics in which self-interested agents engage with each other and receive feedback to revise their strategies (e.g., see [15, 16, 17]). In particular, these learning dynamics do not assume a centralized system where all the agents learn the same strategy and are indistinguishable from each other.

To address this issue, we present a decentralized mean-field Games (DMFG) model-free learning

[1] The major agents correspond to the financial institutions with relatively larger market powers, lower transaction costs and large inventories to liquidate within some time interval $[0, T]$ based on its fundamental analysis. The minor agents correspond to the individual high-frequency traders with smaller market powers, higher transaction costs and small initial inventories at the time stage 0.

[2] For example, the full information about the dynamics of the underlying and the action of each agent is observable to all the other participants as a result of the introspective-thinking process.

algorithm based on [18] for non-communication Markov multi-agent games over a finite horizon with minimal information available to agents within accessible computational costs. In particular, each agent only has access to their private state information, i.e., their cash and inventory processes, plus the public state information such as the asset price process and current time stage t. Further, model-free means they do not know a model of the underlying state transitions probabilities or the market impacts, even including themselves. We then compare the performance of the DMFG algorithm with the theoretical optimal trading strategies and Nash equilibria. In the context of the generalized optimal liquidation problem, we observe significantly different behaviors and offer some intuitions and explanations to offer some tractability of the DMFG algorithm. These insights suggest some exciting topics for subsequent research.

In chapter 1 of this thesis project, we first introduce Merton's classical optimal liquidation problem in section 1.1 and solve the extended version using the HJB equation in section 1.2. Then, we present one practical extension of the classical problem to show the limitation of the HJB equation method. Afterward, we review the Gateaux derivative in section 1.3 as the general method in the high-frequency trading literature for solving the optimal execution problems in chapter 4.

In the second part of the paper in chapter 2, we start by introducing the stochastic games in section 2.1 as a framework for representing dynamic multi-agent interactions. Then we discuss the existing provably convergent learning algorithms with gradually less and less information observable for the agents. Finally, we would reach the *Decentralized Mean-Field Game* (DMFG) framework proposed by Subramanian et al.[18]

Next, we present our generalized optimal liquidation problem in chapter 3 under the context of the mean-field games based on Casgrain and Jaimungal [11]. Based on this generalization, we propose three examples, including

- optimal liquidation for a single agent,
- trading with crowds,
- (the most general case) mean-field games with the same true belief,

inspired by [19, 9, 11], respectively. We then provide the theoretical and DMFG solutions for each problem presented above in chapter 4 and chapter 5. In particular, we compare and investigate the performance between the theoretical Nash equilibrium and DMFG equilibrium for each example and offer some exciting explanations and topics for future research.

Chapter 1

High-Frequency Trading and Mean Field Game

High-frequency trading is an essential class of algorithmic trading, which refers to any form of quantitative trading strategy that computer programs execute automatically. In recent years, automated algorithmic trading algorithms across electronic trading platforms have developed rapidly and attracted the attention of numerous researchers. In particular, optimal liquidation is one of the most extensively studied problems in high-frequency trading.

1.1 Introduction to Optimal Liquidation

Suppose an agent holding a large amount of stock for a risky asset in the financial market decides to liquidate these shares as suggested by its fundamental portfolio analysis. Therefore, the agent needs to find an optimal liquidating strategy to address this target by the suggested time stage T to maximize its expected return with a sense of urgency of getting rid of these shares during the time interval $[0, T]$. Given the fact that the market does not always provide infinite liquidity to absorb a large market order at the best available price, the faster the agent liquidates its inventory, the more significantly negative market price impact it would incur on the market, and hence the poorer prices it would obtain at execution.

To model the trade-off between the market impact (and thus execution cost) and the risk of future price uncertainty, let $\nu_t > 0$ (or < 0) denotes the rate at which the agent buys (or sells) its shares at time t, and thus the agent buys $\nu_t dt$ (or sells $-\nu_t dt$) amount of shares within the infinitesimal time interval dt, the agent seeks to maximize its *objective function*

$$\mathcal{R}_0(\nu) = \mathbb{E}\left[X_T^\nu + q_T^\nu\left(S_T^\nu - \Phi q_T^\nu\right) - \phi\int_0^T (q_t^\nu)^2 dt\right] \tag{1.1.1}$$

given some strategies $\nu := (\nu_t)_{t\in[0,T]} \in \mathcal{A}$ and the dynamic processes

$$\begin{cases} dq_t^\nu = \nu_t dt, \quad q_0^\nu = q \\ dS_t^\nu = -\lambda\nu_t dt + \sigma dW_t, \quad S_0^\nu = S \\ dX_t^\nu = \nu_t(S_t^\nu - c\nu_t)dt, \quad X_0^\nu = X_0 \end{cases}, \tag{1.1.2}$$

where

- $\nu := (\nu_t)_{t\in[0,T]}$ represents the (positive) rate at which the agent trades (liquidation rate) and is what the agent can control,

- $q^\nu := (q_t^\nu)_{t\in[0,T]}$ represents the agent's inventory process controlled by ν.
- $S^\nu := (S_t^\nu)_{t\in[0,T]}$ represents the fundamental price process controlled by ν.
- $X^\nu := (X_t^\nu)_{t\in[0,T]}$ represents the agents cash process controlled by ν.
- $\mathcal{A}$ represents the admissible set of strategies consists of $\mathcal{F}$-predictable non-negative bounded strategies to exclude repurchasing of shares and keeps the liquidation rate finite.

In the objective function $\mathcal{R}_0$, $\Phi, \phi > 0$ implement a penalty on the agent's terminal and running inventory, respectively. In the price process S^ν, $\lambda > 0$ models a linear transient price impact. Whereas in the cash process X^ν, $c > 0$ models a instantaneous transaction cost of the agent's trading strategy ν on its execution price.

1.2 Hamilton-Jacobi-Bellman Equation

1.2.1 General Hamilton-Jacobi-Bellman Equation

One of the most well-known algorithmic trading methods to solve this optimization problem is to apply the Hamilton-Jacobi-Bellman (HJB) equation. To do so, we first define the general time-indexed *performance criteria* by

$$\mathcal{P}^\nu(x_t) = \mathbb{E}_t\left[G(x_T) + \int_t^T F(x_s,\nu_s)ds\right], \tag{1.2.1}$$

where $x := (x_t)_{t\in[0,T]}$ is the vectorized state process such that $x_t := (t, q_t^\nu, S_t^\nu, X_t^\nu)$ represents the information observable to the agent by time stage t. Without loss generality, we denote $\mathcal{F}_t$ as the filtration generated by the vectorized state process observable to the agent x_t throughout this paper. Formally,

$$\mathcal{F}_t := \sigma\left((x_u)_{u\in[0,t]}\right). \tag{1.2.2}$$

In the context of the previous optimal liquidation example, function G and F correspond to

$$G(x_T) := X_T^\nu + q_T^\nu\left(S_T^\nu - \Phi q_T^\nu\right), \tag{1.2.3}$$
$$F(x_t,\nu_t) := (q_t^\nu)^2, \tag{1.2.4}$$

as we defined within the objective function 1.1.1. Intuitively, the performance criteria provides a *score* of the current state x_t for the agent given the filtration $\mathcal{F}_t$.

Secondly, by defining the time-indexed *optimal performance criteria*

$$\mathcal{P}(x_t) = \sup_{\nu\in\mathcal{A}} \mathcal{P}^\nu(x_t), \tag{1.2.5}$$

we obtain the time indexed analog of the original optimization problem in section 1.1. In particular, $\mathcal{P}(x_t)$ would provide the score of the current vectorized state x_t if the optimal admissible strategy is executed since time stage t.

Notice that the optimal performance criteria definition implicitly indicates the overlapping sub-problem feature and optimal sub-structure property. More precisely, the optimal performance criteria $\mathcal{P}(x_t)$ at current time stage t would equal to the performance that optimally executed between $(t, t+dt)$ plus the expected optimal performance criteria $\mathcal{P}(x_{t+dt})$ conditional on $\mathcal{F}_t$ at the future time stage $t+dt$ given x_{t+dt} from x_t. Formally, one may obtain the dynamic programming principle stated as follows.

Theorem 1.2.1 (Dynamic Programming Principle)**.** *The optimal performance criteria in equation* (1.2.5) *satisfies*

$$\mathcal{P}(x_t) = \sup_{\nu \in \mathcal{A}} \mathbb{E}_t \left[\mathcal{P}(x_s)\right], \tag{1.2.6}$$

for any $0 < t < s < T$.

Remark 1.2.2. Note that even though we denote $\sup_{\nu \in \mathcal{A}}$ with a slight abuse of notation, the selected $\nu = (\nu_t)_{t \in [0,T]}$ would only affect the performance between (t, s) as $\mathcal{R}(x_s)$ represents the optimal performance criteria of the state x_s *given the optimal admissible strategy executed since time stage s.*

Then, take the infinitesimal perspective of the dynamic programming principle in theorem 1.2.1 using Itô's lemma at t, we obtain that

$$d\mathcal{P}(x_t) = (\partial_t + \mathcal{L}_t^\nu)\mathcal{P}(x_t)dt + diag(\mathcal{D}_x\mathcal{P}(x_t))\boldsymbol{\sigma}_x^\nu d\boldsymbol{W}_t, \tag{1.2.7}$$

where $\mathcal{L}_t^\nu$ represents the infinitesimal generator of x_t, $\mathcal{D}_x\mathcal{P}(x_t) := (\partial_i \mathcal{P}(x_t))$, with $diag(\boldsymbol{v})$ represents the diagonalized matrix for some vector $\boldsymbol{v} = (v^1, \cdots, v^n)$ such that

$$diag(\mathcal{D}_x\mathcal{P}(x_t)) := \begin{bmatrix} \partial_1\mathcal{P}(x_t) & & \\ & \ddots & \\ & & \partial_n\mathcal{P}(x_t) \end{bmatrix},$$

and $\boldsymbol{\sigma}_x^\nu$ represents the vectorized volatility part of the corresponding stochastic differential equations. In the context of the previous optimal liquidation example in section 1.1, the volatility term in equation (1.2.7) can be simplified as

$$\left(\mathcal{D}_x\mathcal{P}(x_t)\right)^T \boldsymbol{\sigma}_x^\nu = (0, 0, \sigma\partial_s\mathcal{P}(x_t), 0), \tag{1.2.8}$$

since only the price process S^ν's dynamic involves non-zero volatility part as shown in the equation (1.1.2).

Combining the equation (1.1.2) and theorem 1.2.1, we would conclude the the Hamilton-Jacobi-Bellman (HJB) equation (also known as the Dynamic Programming Equation) as follows.

Theorem 1.2.3 (Hamilton-Jacobi-Bellman equation)**.** *The optimal performance criteria in equation* (1.2.5) *satisfies*

$$\begin{cases} \partial_t\mathcal{P}(x_t) + \sup_{\nu \in \mathcal{A}} \{\mathcal{L}_t^\nu \mathcal{P}(x_t) + F^\nu(x_t)\} & = 0 \\ \mathcal{P}(x_T) & = G(x_T) \end{cases} \tag{1.2.9}$$

Remark 1.2.4. In this subsection, we only provide the intuition behind the proofs of theorem 1.2.1 and theorem 1.2.3. Interested readers are highly recommended to consult the textbook [3, chap 6] written by Cartea, Jaimungal and Penalva.

1.2.2 Solve the Classical Optimal Liquidation Problem

Given the general form of the HJB equation in theorem 1.2.3, we may solve the optimal liquidation problem defined in section 1.1. In fact, this problem is an extended exercise in [3, E.6.1]. In this subsection, we will provide a sketched solution to attain the optimal trading strategy ν^* using the HJB equation.

First, given the objective function defined in equation (1.1.1), we can write down the HJB equation in the context of this example as

$$\begin{cases} \partial_t \mathcal{P} + \sup_{\nu \in \mathcal{A}} \left\{ \nu(S - c\nu)\partial_X \mathcal{P} + \nu \partial_q \mathcal{P} - \lambda \nu \partial_S \mathcal{P} + \frac{1}{2}\sigma^2 \partial_{SS}\mathcal{P} \right\} & = 0 \\ \mathcal{P}(T, X, q, S) & = X + q(S - \Phi q) \end{cases} . \tag{1.2.10}$$

Regard the performance criteria $\mathcal{P}$ as given, we may solve the optimal trading strategy ν^* as

$$\nu^* = \frac{S\partial_X \mathcal{P} + \partial_q \mathcal{P} - \lambda \partial_S \mathcal{P}}{2c\partial_X \mathcal{P}}. \tag{1.2.11}$$

Hence, by plugging equation (1.2.11) back into equation (1.2.10), one can remove the $\sup_{\nu \in \mathcal{A}}$ term of the first equation and attain a partial differential equation (PDE) bound with the terminal condition by the second equation. That is, the key to solve the optimal trading strategy ν^* is to solve the optimal performance criteria $\mathcal{P}$, which requires to provide an ansatz (i.e., guess) to solve the PDE. In this example, we will make the ansatz as

$$\mathcal{P}(t, X, q, S) = X + qS + h(t, q), \tag{1.2.12}$$

for some function $h \in \mathbb{C}^2(\mathbb{R})$.

Remark 1.2.5. This example is a classical optimal execution problem with regularly defined objective functional and dynamics. But in general, making up the ansatz would be one of the main obstacles to solve the single-agent optimal execution problem using the HJB equation. The limitation will mainly be discussed in section 1.2.3.

Plugging the equation (1.2.12) into equation (1.2.11), we obtain

$$\nu^* = \frac{2S - \lambda q + \partial_q h}{2c}, \tag{1.2.13}$$

and hence the equation (1.2.10) can be simplified as

$$\begin{cases} (-2S + \lambda q)^2 + 4c\partial_t h + (4S - 2\lambda q)\partial_q h + (\partial_q h)^2 & = 0 \\ h(T, q) & = -\Phi q^2 \end{cases} . \tag{1.2.14}$$

Again, we need to make the ansatz for the function h as

$$h(t, q) = f_0(t) + q f_1(t) + q^2 f_2(t), \tag{1.2.15}$$

so that the equation (1.2.14) can be simplified as

$$\begin{cases} \left[(2S - \lambda q) + (f_1(t) + 2q f_2(t))\right]^2 + 4c(f_0^{'}(t) + q f_1^{'}(t) + q^2 f_2^{'}(t)) = 0 \\ f_0(T) = 0, \quad f_1(T) = 0, \quad f_2(T) = -\Phi \end{cases} , \tag{1.2.16}$$

for any q. Via grouping the first equation in 1.2.16 by q and realizing that each coefficients terms of q need to be zero, we attain the following ODE system

$$\begin{cases} \lambda^2 - 4\lambda f_2(t) + 4 f_2(t)^2 + 4c f_2^{'}(t) & = 0, \quad f_2(T) = -\Phi \\ -4S\lambda - 2\lambda f_1(t) + 8S f_2(t) + 4 f_1(t) f_2(t) + 4c f_1^{'}(t) & = 0, \quad f_1(T) = 0 \\ 4S^2 + 4S f_0(t) + f_1(t)^2 + 4c f_0^{'}(t) & = 0, \quad f_0(T) = 0 \end{cases} , \tag{1.2.17}$$

which solves

$$f_2(t) = \frac{4c\Phi + \lambda(2\Phi+\lambda)(t-T)}{4c - 2\lambda(2\Phi+\lambda)(t-T)} \tag{1.2.18}$$

$$f_1(t) = \frac{2S(t-T)(2\Phi+\lambda)}{2c-(t-T)(2\Phi+\lambda)} \tag{1.2.19}$$

$$f_0(t) = \frac{2S^2(t-T)}{-2c+(t-T)(2\Phi+\lambda)} \tag{1.2.20}$$

for any t. It follows that

$$\nu^*(t,X,q,S) = \frac{-2S+q(2\Phi+\lambda)}{-2c+(t-T)(2\Phi+\lambda)}, \quad \forall (t,X,q,S) \in [0,T]\times\mathbb{R}^2\times\mathbb{R}_+. \tag{1.2.21}$$

Remark 1.2.6. The restriction over S in equation (1.2.21) results from the restriction of the admissible trading strategy space $\mathcal{A}$. As the HJB equation is not the main focus of this paper, we would not dive into the technical details. Interested readers may check the convexity in the objective function (or the performance criteria) if the price process goes to negative for verification purposes.

1.2.3 Limitation of Hamilton-Jacobi-Bellman Equation

In this subsection, we will attempt to solve an extended optimal liquidation problem which is the simplified version of [19] by Neuman and Voss in 2022.

consider the optimal liquidation problem described in section 1.1, but with persistent transient price impact decaying gradually over time at some exponential resilience rate $\rho > 0$. Formally, we have the same objective function as in equation (1.1.1) but the dynamic processes becomes

$$\begin{cases} dq_t^\nu = \nu_t dt, \quad q_0^\nu = q_0 \\ dY_t^\nu = -\rho Y_t^\nu dt + \lambda \nu_t dt, \quad Y_0^\nu = y_0 \\ dS_t^\nu = -dY_t^\nu + \sigma dW_t, \quad S_0^\nu = S_0 \\ dX_t^\nu = \nu_t(S_t^\nu - c\nu_t)dt, \quad X_0^\nu = X_0 \end{cases}, \tag{1.2.22}$$

where Y_t^ν corresponds to the transient price impact that persists and decays gradually overtime at some exponential resilience rate $\rho > 0$. In particular, the transient price impact process Y can be derived as

$$Y_t^\nu = e^{-\rho t} y + \gamma \int_0^t e^{-\rho(t-u)} \nu_u du, \quad 0 \le t \le T. \tag{1.2.23}$$

for some initial value $y \in \mathbb{R}$. Note that the extra factor Y raises an extra argument for the performance criteria $\mathcal{P}$ because it cannot be canceled in the SDE system. That is where the complexity is introduced.

Following the similar process in section 1.2.2, we obtain the HJB equation as

$$\begin{cases} \partial_t\mathcal{P} + \sup_{\nu\in\mathcal{A}}\left\{\nu(S-c\nu)\partial_X\mathcal{P} + \nu\partial_q\mathcal{P} + (Y-\lambda\nu)(\partial_S\mathcal{P}+\partial_Y\mathcal{P}) + \frac{1}{2}\sigma^2\partial_{SS}\mathcal{P}\right\} & = 0 \\ \mathcal{P}(T,X,q,Y,S) & = X + q(S-\Phi q) \end{cases}. \tag{1.2.24}$$

and thus

$$\nu^* = \frac{S\partial_X\mathcal{P} + \partial_q\mathcal{P} - \lambda(\partial_Y\mathcal{P}+\partial_S\mathcal{P})}{2c\partial_X\mathcal{P}}. \tag{1.2.25}$$

However, notice that the key difficulty to solving the equation (1.2.24) is that Y is implicitly related to S but does not explicitly affect $\mathcal{P}$ at terminal time stage T. Intuitively, the agent should not care about the market impact process Y by terminal stage T either, as the agent will exit the market by that time. Still, the HJB equation is solvable in theory, but because of this feature, making up the ansatz for the $\mathcal{P}$ becomes exceptionally challenging. [1]

Remark 1.2.7. Note that this extension is only a simplified version of [19], let along the multi-agent version of the optimal liquidation problem as we may see in [9, 11] [2].

1.3 Gateaux Derivative

While the HJB equation provides a convenient framework to solve a massive class of single-agent optimal execution problems, making up the correct ansatz is one of its main limitations, as we indicated in remark 1.2.5. Further, the complexity of solving the PDE would massively grow when we extend the classical optimal liquidation problem, which brings us incapable of solving it. In contrast, the Gateaux derivative offers a more general methodology to solve the most regularly defined algorithmic and high-frequency trading problems (e.g., see [19, 9, 11]). Therefore, in this section we would provide a review for the Gateaux derivative.

Definition 1.3.1. Given a function $F : U \to \mathbb{R}$ for some locally convex topological vector spaces U, the Gateaux derivative $\langle F(\nu), \alpha \rangle$ at ν in direction $\alpha \in U$ is defined as

$$\langle F(\nu), \alpha \rangle := \lim_{\epsilon \to 0} \frac{F(\nu + \epsilon\alpha) - F(\nu)}{\epsilon}. \tag{1.3.1}$$

If the limit exists for all $\alpha \in U$, then we say that F is Gateaux differentiable at ν.

Intuitively, consider the objective functional $\mathcal{R}_0(\nu)$ for the action process $\nu := (\nu_t)_{t\in[0,T]}$. Then the Gateaux derivative $\langle \mathcal{R}_0(\nu), \alpha \rangle$ describes the change of the value if the agent tries slightly different action $\nu_t + \epsilon\alpha_t$ for certain time stages between $[0, T]$. Therefore, following the analog of the derivative in the Euclidean space, the agent should attain the *critical* strategy ν^* when $\langle \mathcal{R}_0(\nu^*), \alpha \rangle = 0$.

Based on the Gateaux derivative, one would now be equipped with the tool to solve the most optimal execution problems in the high-frequency trading literature. In chapter 4, we will present the details of using the Gateaux derivative to solve the the generalized optimal liquidation problem.

[1] Alternatively, one may still solve the PDE of the $\mathcal{P}$ numerically. However, this has been out of the scope of this paper. Thus, we would not investigate it.

[2] Similarly, one may still solve the multi-agent optimal liquidation problem by extending the HJB equation to the so-called Nash-HJB equation. However, this has been out of the scope of this paper. Thus, we would not investigate it either.

Chapter 2

Stochastic Game and Reinforcement Learning

While optimal liquidation problems have arisen in financial modeling, we have witnessed the exciting development of reinforcement learning in solving general multi-agent stochastic games in recent years (e.g., see [20, 21]). Meanwhile, we also discussed the limitation of the existing algorithmic trading methodologies in the previous chapter. Therefore, we are motivated to extend our attention to reinforcement learning, focusing on multi-agent stochastic games.

In the following of this chapter, we will start by introducing the general framework for reinforcement learning and then discussing the existing provably convergent learning algorithms with gradually less and less information observable for the agents. These additional restrictions provide a better approximation for the financial market in reality, where trading agents are only offered minimal information.

2.1 Introduction of Stochastic Game

Stochastic games (also known as Markov Games) have been widely studied and used (e.g., see [22, 23, 24]) for solving multi-agent interaction problems since 1953 by Shapley [25]. For the setting of this paper, an N-player stochastic game can be characterized formally by $\langle \mathcal{X}, \mathcal{X}^j, \mathcal{A}^j, r^j, p, \gamma \rangle_{j \in \mathbb{N}}$, where

1. The set of *private states* $\mathcal{X}^j$ consists of the *private state* $x_t^j \in \mathcal{X}^j$ containing the information that is observable for the agent j at time stage t, for any $t \in [0, T]$,

2. The set of *global states* $\mathcal{X}$ consists of the *global state* $x_t \in \mathcal{X}$ containing the information of all agents in the system at time stage t, for any $t \in [0, T]$,

3. The set of *admissible actions* $\mathcal{A}^j := (\mathcal{A}_t^j)_{t \in [0,T]}$ consists of the action $\nu^j = (\nu_t^j)_{t \in [0,T]} \in \mathcal{A}^j$ that the agent j can choose, where $\mathcal{A}_t^j$ represents the set of admissible actions for the agent j at time stage t,

4. The *immediate reward function* $r^j : \mathcal{X} \times \boldsymbol{\mathcal{A}} \to \mathcal{R}$, where $\boldsymbol{\mathcal{A}} := \prod_{j=1}^N \mathcal{A}^j$, corresponds to the immediate reward received for the agent j at each global state x_t after the system executes the actions of all agents $\boldsymbol{\nu}_t := (\nu_t^1, \cdots, \nu_t^N)$ at time stage t, for any $t \in [0, T]$,

5. The *transition probability* $p(x_{t+1} | x_t, \boldsymbol{\nu}_t)$ corresponds to the probability of transitioning from global state x_t to x_{t+1} given the joint action $\boldsymbol{\nu}_t$ executed at time stage t, for any $t \in [0, T]$,

6. The *discount factor* $\gamma^j \in [0, 1]$ corresponds to the discounted accumulation of the future reward to the current utility of the agent j.

It then follows that the target of the agent j is to find the strategy $\nu^{j,*} = (\nu_t^{j,*})_{t\in[0,T]}$ that maximizes its *objective function*, i.e., the expected sum of discounted future immediate reward, at time stage $t = 0$, given by

$$\mathcal{R}_0^j(x_t, \nu^j) := \mathbb{E}\left[\sum_{t=0}^{T} (\gamma^j)^t \cdot r^j(x_t; \nu^j, \boldsymbol{\nu}_t^{-j})\right] \tag{2.1.1}$$

for some partition of the time interval $[0, T]$. Here $\boldsymbol{\nu}^{-j} := (\nu^1, \cdots, \nu^{j-1}, \nu^{j+1}, \cdots, \nu^N)$ denotes the *joint opponent action* of all the agents in the system except for the agent j, and $\{x_0 \sim p_0, x_{t+1} \sim p(\cdot|x_t, \boldsymbol{\nu}_t), t \in [0, T]\}$ denotes the *state process* at each time stage t with $p_0 \in \Delta(\mathcal{X})$ as the initial state distribution.

Remark 2.1.1. The discount factor γ is usually restricted within $[0, 1)$ in the literature when $T = \infty$ to avoid the potential divergence of $\mathcal{R}_0^j$. But as we only focus on the optimal liquidation problem within finite terminal time $T < \infty$, this hyper-parameter γ has no particular restriction.

Remark 2.1.2. Unlike the optimal liquidation problem defined in section 1.1, the "optimal" is actually ill-defined in the context of multi-agent stochastic games. Precisely, the "optimal" outcome of the agent j might not follow its opponent's optimal wishes, especially in zero-sum games. To reconcile this conflict, we will introduce two related but distinct lines of research in the following sections.

2.2 Centralized Q-learning

While a Q-learning based algorithm for attaining Nash equilibria in general stochastic games was firstly introduced by Hu and Wellman [26], one of the first attempts to apply Deep Q-learning in solving such a multi-agent optimal liquidation problem is Jaimungal's Nash-DQN [12]. The primary focus is to solve the Nash equilibrium and study the properties without a prior knowledge of its dynamics.

In order to do this, they first define the *objective function* for each agent j as

$$\mathcal{R}_0^j(x_0, \nu^j; \boldsymbol{\nu}^{-j}) = \mathbb{E}\left[\sum_{t=0}^{T} (\gamma_j)^t r^j(x_t, \nu_t^j; \boldsymbol{\nu}_t^{-j})\right], \tag{2.2.1}$$

following the stochastic process $\{x_0 \sim p_0, x_{t+1} \sim p(\cdot|x_t, \boldsymbol{\nu}_t), t \in [0, T]\}$ as defined in 2.1.1 so that each agent j can attain its utility $\mathcal{R}_0^j(x_0, \nu^j; \boldsymbol{\nu}^{-j})$ at time stage $t = 0$, given their opponent joint action $\boldsymbol{\nu}^{-j}$. In other words, the target of agent j is now to obtain an action $\nu^{j,*}$ that optimizes their objective function, but also as a function of $\boldsymbol{\nu}^{-j}$, for all $j \in \mathfrak{N}$. In the end, those agents' action processes would form a Nash equilibrium, which can be formally defined as follows.

Definition 2.2.1. A collection of admissible actions $\boldsymbol{\nu}^* := (\nu^{1,*}, \cdots, \nu^{N,*})$ forms a Nash Equilibrium if

$$\mathcal{R}_0^j(x_0, \nu^j; \boldsymbol{\nu}^{-j,*}) \leq \mathcal{R}^j(x_0, \nu^j; \boldsymbol{\nu}^{-j,*}) \tag{2.2.2}$$

for all admissible strategies ν^j and for all $j \in \mathfrak{N}$.

Next, they extended the Bellman equation for Nash equilibria to fit into the reinforcement learning framework. To do this, they applied the dynamic programming principle to the agent j's objective function $\mathcal{R}_0^j(x_0, \nu^j; \boldsymbol{\nu}^{-j,*})$, which results in

$$\mathcal{R}_t^j(x_t, \nu^{j,*}; \boldsymbol{\nu}^{-j,*}) = \max_{\nu_t^j \in \mathcal{A}_t^j} \left\{ r^j(x_t, \nu_t^j; \boldsymbol{\nu}_t^{-j,*}) + \mathbb{E}_{x_{t+1} \sim p(\cdot | x_t, \nu_t^j, \boldsymbol{\nu}_t^{-j,*})} \left[\mathcal{R}^i(x_{t+1}, \nu^{j,*}; \boldsymbol{\nu}^{-j,*}) \right] \right\}, \tag{2.2.3}$$

for all $t \in [0, T]$. Without loss generality, here we abuse the notation $\mathcal{R}_t^j$ as the expected total reward from time stage t to the terminal time stage T.

That is, for any time stage t, the agent j's objective function $\mathcal{R}_t^j$ in the Nash equilibrium at time stage t is equal to the maximum of all possible summations of the current immediate reward r^j and the expected value of the future total reward $\mathcal{R}_t^j$ among all admissible actions $u_t \in \mathcal{A}_t^j$, which follows the Nash equilibrium $\boldsymbol{\nu}$ afterward until terminal time stage T. Here, the next state x_t is influenced by the current action ν_t^j via the transition probability $p(\cdot | x_t, \nu_t^j, \boldsymbol{\nu}_t^{-j})$, and hence the future total reward $\mathcal{R}_t^j$ would also vary depending on the selected current action ν_t^j. Therefore, this new Bellman equation for Nash equilibria is satisfied simultaneously at Nash equilibria for all $j \in \mathfrak{N}$ by definition.

Furthermore, the authors considered vectorizing the notation for conciseness and consistency in the reinforcement learning literature. That is, by denoting the vectorized expected total reward function $\boldsymbol{\mathcal{R}}_t(x_t, \boldsymbol{\nu}^*) := (\mathcal{R}_t^j(x_t, \boldsymbol{\nu}^*))_{j \in \mathfrak{N}}$, they defined the stacked *Nash state value function* $\boldsymbol{V}(x_t)$ as

$$\boldsymbol{V}(x_t) := (V^j(x_t))_{j \in \mathfrak{N}} := (\mathcal{R}_t^j(x_t, \boldsymbol{\nu}^*))_{j \in \mathfrak{N}} \tag{2.2.4}$$

as well as the stacked *Nash Q-function* as

$$\boldsymbol{Q}(x_t, \boldsymbol{\nu}_t^*) := \boldsymbol{r}(x_t, \boldsymbol{\nu}_t^*) + diag(\boldsymbol{\gamma}) \mathbb{E}_{x_{t+1} \sim p(\cdot | x_t, \boldsymbol{\nu}_t^*)} \left[\boldsymbol{V}(x_{t+1}) \right], \tag{2.2.5}$$

with $\boldsymbol{r} := (r^j)_{j \in \mathfrak{N}}$, $\boldsymbol{\gamma} := (\gamma^j)_{j \in \mathfrak{N}}$, and $diag(\boldsymbol{v})$ denotes as the diagonalized matrix for some vector $\boldsymbol{v} = (v^1, \cdots, v^n)$ such that

$$diag(\boldsymbol{v}) := \begin{bmatrix} v_1 & & \\ & \ddots & \\ & & v_n \end{bmatrix}.$$

By definition, the Nash Q-function $\boldsymbol{Q}$ represents the expected value of the objective function may take, given the current state x_t and *arbitrary* current action $\boldsymbol{\nu}_t$, but following the Nash equilibrium action $\boldsymbol{\nu}^*$ since after time stage t. Therefore, the reinforcement learning problem would be well-defined once the meaning of "optimal" in the context is well-defined.

Definition 2.2.2 (*Nash Operator* [12])**.** Consider a collection of N concave real-valued functions $\boldsymbol{f}(\boldsymbol{u}) := (f^j(u^j, \boldsymbol{u})^{-j})$, where $f^j : \mathcal{U}^j \times \mathcal{U}^{-j} \to \mathbb{R}$. The Nash operator $\boldsymbol{\mathcal{N}}_{\boldsymbol{u} \in \boldsymbol{\mathcal{U}}} : \boldsymbol{f}(\boldsymbol{u}) \mapsto \boldsymbol{f}(\boldsymbol{u}^*)$, where the Nash equilibrium value $\boldsymbol{u}^* := \arg \boldsymbol{\mathcal{N}}_{\boldsymbol{u} \in \boldsymbol{\mathcal{U}}} \boldsymbol{f}(\boldsymbol{u})$ such that

$$f^j(u^j, \boldsymbol{u}^{-j,*}) \leq f^j(u^{j,*}, \boldsymbol{u}^{-j}), \quad \forall u^j \in \mathcal{U}^j, \forall j \in \mathcal{N}. \tag{2.2.6}$$

That is, the Nash operator corresponds to simultaneously maximizing each of the f^j in their first argument u^j for a sufficiently regular collection of functions $\boldsymbol{f}$.

Therefore, we obtain a relationship between the Nash value function $\boldsymbol{V}$ and the Nash Q-function $\boldsymbol{Q}$ inborn from the definition of the Nash operator as $\boldsymbol{V}(x_t) = \boldsymbol{\mathcal{N}}_{\boldsymbol{\nu}_t \in \boldsymbol{\mathcal{A}}_t} \boldsymbol{Q}(x_t, \boldsymbol{\nu}_t)$, hence the Bellman equation for Nash equilibria in equation (2.2.3) can be rewritten as

$$\boldsymbol{V}(x_t) = \boldsymbol{\mathcal{N}}_{\boldsymbol{\nu}_t \in \boldsymbol{\mathcal{A}}_t} \boldsymbol{Q}(x_t, \boldsymbol{\nu}_t) = \boldsymbol{\mathcal{N}}_{\boldsymbol{\nu}_t \in \boldsymbol{\mathcal{A}}_t} \left\{ \boldsymbol{r}(x_t, \boldsymbol{\nu}_t^*) + diag(\boldsymbol{\gamma}) \mathbb{E}_{x_{t+1} \sim p(\cdot | x_t, \boldsymbol{\nu}_t)} \left[\boldsymbol{V}(x_{t+1}) \right] \right\}, \quad (2.2.7)$$

and hence $\boldsymbol{\nu}^* = (\boldsymbol{\nu}_t)_{t \in [0,T]}$, where $\boldsymbol{\nu}_t^* = \arg \boldsymbol{\mathcal{N}}_{\boldsymbol{\nu}_t \in \boldsymbol{\mathcal{A}}_t} \boldsymbol{Q}(x_t, \boldsymbol{\nu}_t)$. In other words, it is sufficient to obtain the Nash Q-function to obtain the Nash equilibrium $\boldsymbol{\nu}^*$, which is sufficient to provide the update function and loss function as we will see in chapter 5 for details.

Remark 2.2.3. Note that the main difference in the intuition between equation (2.2.3) and equation (2.2.7) is that the previous Bellman equation selects the current *optimal* action ν_t^j for each agent $j \in \mathfrak{N}$ with the opponent's joint action $\boldsymbol{\nu}^{-j}$ regarded as given, whereas the latter includes both the agent j's current action ν_t^j and the opponent's joint action $\boldsymbol{\nu}_t^{-j}$ into the consideration and regards the joint "optimal" action $\boldsymbol{\nu}_t$ as the *Nash equilibrium* at each time stage t.

In other words, the Nash Bellman equation in equation (2.2.7) does not distinguish the action of the agent j or the joint action of the opponent $-j$ anymore. The Nash equilibrium $\boldsymbol{\nu}$ is obtained directly given the Nash Q-function $(\boldsymbol{Q}(x_t, \boldsymbol{\nu}_t))_{t \in [0,T]}$. That is, the Nash equilibrium $\boldsymbol{\nu}$ is governed centrally by the *global* Q-function that regards the global state x_t and the global action $\boldsymbol{\nu}$ as its arguments as a whole. Intuitively, a global processor controls all the agent's actions in the system.

2.3 Independent and Decentralized Q-Learning

Although the Nash Deep Q-Learning presented in section 2.2 provides a concise learning algorithm to obtain the Nash equilibrium, one of the critical questions is whether a Nash equilibrium can be realized or not during the interacting process between these *self-interested* agents in the multi-agent stochastic games. This reflects the obstacle revealed in equation (2.1.1) that agents cannot attain their "optimal" strategies beforehand without a prior opinion (or model) on their opponent's strategy.

When there exists a Nash equilibrium, the centralized learning algorithm would identify and provide the equilibrium strategies relying on an introspective decision-making process. However, many empirical studies suggest that such a thinking process is not sometimes realized, even repeatedly played in games (e.g., [22, 14]). Nagel in 1995, for instance, presented a well-known empirical experiment played among a class of students for submitting an integer between 1 and 100 independently, and the student who chose the number closest to the 2/3rd of the average of the whole class would be the winner. The introspective thinking process that controls the actions of the whole class globally would expect the students to pick 0. However, the empirical results show that students would choose non-zero numbers, and not even close to zero, such that their average ended up around 20 to 30, even if the game was played repeatedly many times [13]. That is to say, the introspective thinker would always lose the game. In contrast, players who observed the chosen numbers and reacted by picking numbers closer to the winning number would achieve convergence to equilibrium along the process (also see in [22]).

Notice that one of the key differences is that the students have not engaged in any forward-looking strategy, but *engaged with each other and received feedback to revise their strategies during the repeated game process* [14]. In particular in equation (2.1.1), each agent j has and keeps updating the prior assumption about their opponent's strategies $\hat{\boldsymbol{\pi}}_t^{-j}$ over time $t \in [0, T]$. Hence, the target of each agent j in this context is to find its optimal action $\nu^{j,*}$ such that its *objective function*

$$\mathcal{R}_0^j(x_t, \nu^j) := \mathbb{E}_{\hat{\boldsymbol{\nu}}^{-j} \sim \hat{\boldsymbol{\pi}}_t^{-j}} \left[\sum_{t=0}^{T} (\gamma^j)^t \cdot r^j(x_t, \nu_t^j; \boldsymbol{\nu}_t^{-j}) \right] \quad (2.3.1)$$

is maximized given their prior assumption. Nevertheless, during the repeated game process, the agents would observe their opponents' actions at time stage t and re-estimate their models for estimating what their opponents will act in the next time stage $t+1$. Such a non-stationary feature increases the complexity of solving such a problem.

2.3.1 Fictitious Play

Since it was introduced by Brown in 1951 [27], fictitious play provides a class of stylist independent learning dynamics that proved to be convergent in repeated zero-sum stochastic games [22, 28, 29]. This type of learning dynamics requires that each agent in the system *erroneously* assume that its opponent plays according to a stationary strategy only depends on the current state x_t. In fact, those agents are essentially playing an *auxiliary stage-game* at each stage t.

Formally, each agent j involves in the auxiliary stage-game which can be characterized as $\langle \mathcal{X}_t, \mathcal{X}_t^j, \mathcal{A}_t^j, Q^j(x_t, \cdot)\rangle_{j\in\mathbb{N}}$ at each stage $t \in [0,T]$. The fundamental difference from the original stochastic game described in section 2.1 is that the reward function in this context is the Q-function $Q^j(x_t,\cdot): \mathcal{A} \to \mathbb{R}$ at current state x_t. That is, each agent j is making their decision at each stage time t directly with the discounted future reward, but conditioned on their own *belief* (or, model) $\hat{\boldsymbol{\pi}}_t^{-j}$ on the opponents' strategy for current state x_t as a *weighted empirical average*. In particular,

$$\nu_t^{j,*} = \underset{\nu_t^j \in \mathcal{A}_t^j}{\operatorname{argmax}}\, \mathbb{E}_{\hat{\boldsymbol{\nu}}^{-j}\sim\hat{\boldsymbol{\pi}}_t^{-j}}\left[Q^j(s_t, \nu_t^j; \hat{\boldsymbol{\nu}}_t^{-j}),\right] \tag{2.3.2}$$

where the Q-function $Q^j(x_t,\cdot;\hat{\boldsymbol{\nu}}_t^{-j})$ under the belief $\hat{\boldsymbol{\pi}}_t^{-j}$ following the dynamic programming principle can be rewritten as

$$Q^j(x_t,\nu_t^j;\hat{\boldsymbol{\nu}}_t^{-j}) = r^j(x_t,\nu_t^j;\boldsymbol{\nu}_t^{-j}) + \gamma^j \max_{\nu_{t+1}^j\in\mathcal{A}_{t+1}^j}\left\{\mathbb{E}_{x_{t+1}\sim p(x_t|\nu_t^j,\hat{\boldsymbol{\nu}}_t^{-j})}\left[Q^j(x_{t+1},\nu_{t+1}^j;\hat{\boldsymbol{\nu}}_{t+1}^{-j})\right]\right\} \tag{2.3.3}$$

and the belief of the agent j follows the incremental update rule as in [22] via

$$\hat{\boldsymbol{\pi}}_{t+1}^{-j}(x) = \hat{\boldsymbol{\pi}}_t^{-j}(x) + \alpha_{c_t(x)}\left(\boldsymbol{\nu}_{t+1}^{-j} - \hat{\boldsymbol{\nu}}_t^{-j}(x)\right) \tag{2.3.4}$$

for any $t\in[0,T]$. Here, $\alpha_{c_t}\in[0,T], \forall t$ is some vanishing learning rate with $c_t(x)$ indicating the number of visits to state x by t. Such a vanishing learning rate indicates that the agent j would weight less to their current belief than the observed action if the state has not been visited many times.

Here, one can quickly notice that the update rule assumes that the private state space $\mathcal{X}^j = \mathcal{X}, \forall j$, and even identical over all time stage t, as well as that the state space $\mathcal{X}$ is finite for any $t\in[0,T]$. Further, while the identicality and finiteness of the state space $\mathcal{X}$ might be acceptable when modeling the financial market, such an updating rule may be problematic when the action space $\mathcal{A}^{-j}$ is contiguous in our optimal liquidation problem. Mainly, the practitioner would run into a dilemma: either the search space is too large to ensure visiting all state-action pairs at least once if we discrete both the state space and action space according to the market unit, or this independent learning dynamics cannot be applied in the financial market.

Inspired by the intuition of the weighted-average updating rule and the potential computational issue, here we suggest modelling the estimated opponents' strategies via a neural network π_θ^j, which will provide a distribution of opponents' actions given the state x. Since such a neural network is trained by observing the historical opponents' action, the neural network can produce a quasi-weighted-average distribution even over the contiguous action space, with some modification over either the network structure or training process or historical data. Further, we can set the neural network to be updated at each time stage t so that it fits into the fictitious play set-up.

2.3.2 Decentralized Q-learning

Even though fictitious play provides a simple and stylist learning algorithm with minimal information available to agents, the requirement that each agent erroneously assumes that the opponents play according to stationary strategies is sometimes unrealistic, especially in the financial market. Another way to address the non-stationarity issue, on the other hand, was proposed by Leslie, Basar and Ozdaglar in 2021 [30]. Instead of assuming fictitious stationary strategies, they suggested a two-timescale adaptation Q-learning dynamic for each agent, introduced by Leslie and Collins in 2005 [14, 31] and originating in Fudenberg and Levine in 1998 [14]. In particular, agents would infer their opponents' actions through an estimate of their opponents' local Q-function, thus generating an estimate of the value function to infer their own continuation reward afterward.

Formally, the learning algorithm in this context can be characterized as follows. Instead of forming a belief about the opponents' strategies, the agents can *estimate* their *local Q-function*

$$q^j_{\boldsymbol{\pi}^{-j}}(x, \nu^j) := \mathbb{E}_{\boldsymbol{\nu}^{-j} \sim \boldsymbol{\pi}^{-j}} \left[Q^j_{\boldsymbol{\pi}^{-j}}(x, \nu^j, \boldsymbol{\nu}^{-j}) \right] \tag{2.3.5}$$

to infer their opponents' actions. Note that here the opponents' strategies $\boldsymbol{\pi}^{-j}$ is the *true* strategies which are non-stationary over the time stage t. As a consequence, the agent j in this context is not able to form about $\boldsymbol{\pi}^{-j}$. Correspondingly, Leslie et al. suggested a updating algorithm in [22, Table 1] that has shown to be provably convergent. In particular, the local Q-function is updated incrementally as

$$\hat{q}^j_{t+1}(x_t, \nu^j_t) = \hat{q}^j_t(x_t, \nu^j_t) + \alpha^j_{c(x_t)}(r^j_k + \gamma^j \hat{v}^j_t(x_t) - \hat{q}^j_t(x_t, \nu^j_t)), \tag{2.3.6}$$

with the auxiliary value function estimates as

$$\hat{v}^j_{t+1}(x_t) = \hat{v}^j_t(x_t) + \beta^j_{c(x_t)} \hat{q}^j_v(x_t)(\pi^j_t \hat{q}^j_t(x_t) - \hat{v}^j_t(x_t)) \tag{2.3.7}$$

for all agent $j \in \mathfrak{N}$ and any time stage $t \in [0, T]$. Here $\alpha^j(x_t), \beta^j(x_t) \in (0, 1)$ are some diminishing learning rates as defined in section 2.3.1 for updating the local Q-function and value function.

Remark 2.3.1. Though the updating rule in equation (2.3.6) is not explicitly revealed from equation (2.3.5), the agents in this context can infers their opponents' strategies and establish provably convergent Nash equilibrium. Surprisingly, the critical intuition and reasoning behind this are exactly where the issue of non-stationarity comes from. In other words, it is because the true local-Q function involves the unknown true opponents' strategies $\boldsymbol{\pi}^{-j}_t$ that the estimated local-Q function can reveal such a true $\boldsymbol{\pi}^{-j}_t$ from the estimation over time $t \in [0, T]$. Despite the fact that this would slow down the update of the value function and Q-function estimate in nature as in general [32, 33], this does help weaken the dependence between the configuration of the stage games.

2.3.3 Decentralized Mean Field Game

Independent learning (e.g., fictitious play) and decentralized Q-learning provide a form to solve the multi-agent stochastic games instead of a centralized system where all the agents are effectively indistinguishable and learn the same policy. However, they either require prior knowledge (or assumption) about each opponent's strategies or an extremely long time to converge, even for a simple two-player zero-sum discrete game. In particular, the fictitious play also strictly specified the learning pattern (i.e., erroneously assume each opponent's strategy is stationary, for each agent j). Besides, the independent learning in general demands the complete information of the system to be accessible to each agent. That is, each agent will be able to observe the individual action of its opponents. Even if we modify the learning pattern as a function of the mean-field action μ only, it still requires to obtain all the agents action first to compute the current mean-field action. All of the limitations above are not practical in the financial market.

Notice that obtaining the exact equilibrium action ν_t^j at time stage t in the independent and decentralized learning requires the complete information of its opponent's current action $\boldsymbol{\nu}_t^{-j}$ as well. This arises the so-called 'chicken-and-egg' problem as described in [18]. Motivated to address this problem, Subramanian, Taylor, Crowley and Poupart in 2022 proposed a new mean-field system called *Decentralized Mean-Field Games (DMFG)* in that paper. They relaxed the strong assumptions of the centralized system where agents are indistinguishable and learn the same strategy, and obtained a stronger performances compared to common baselines with minimal information playing in several well-known multi-agent playground in reinforcement learning literature.

Formally, the learning algorithm in the DMFG framework can be characterized by the updating rules we presented in the previous subsections. Neither focusing on the global nor local Q-functions, the new updating equation emphasizes the *mean-field action* $\tilde{Q}^j(x_t, \nu_t^j, \mu_t)$, where $\mu := (\mu_t)_{t\in[0,T]}$ corresponds to the mean-field actions as we indicated in the first paragraph.

$$\tilde{Q}_{t+1}^j(x_t, \nu_t^j, \hat{\mu}_t^j) = \tilde{Q}_t^j(x_t, \nu_t^j, \hat{\mu}_t^j) + \alpha_{c(x_t)}^j(r_k^j + \gamma^j \hat{v}_t^j(x_t) - \tilde{Q}_t^j(x_t, \nu_t^j, \hat{\mu}_t^j)), \tag{2.3.8}$$

with similar setting as we have described in equation (2.3.6), except that here, the mean-field action μ_t is not observable until the time stage $t+1$. To this end, each agent needs to *predict* the current mean-field action μ_t by

$$\hat{\mu}_t^j(x_t, \mu_{t-1}) = f^j(x_t, \hat{\mu}_t^j), \quad \forall i \in \mathfrak{N}. \tag{2.3.9}$$

Remark 2.3.2. The theoretical support behind the approximation of the mean-field Q-function to the global-Q function via the local-Q functions can be found in section 5.1 and for details in [34].

The DMFG algorithm provides a more practical framework than most existing centralized and decentralized learning methods, especially in the financial market setting where the market participants are neither identical nor simply predictable and observe minimal information (public market information plus the turnover and cash inflows and Outflows inferring the last period mean-field action). Further, the actions of the market participants, particularly some large financial institutions, are sometimes not negligent. Those advantages inspired us to implement the new mean-field system by Subramanian et al. to solve our optimal liquidation problem.

However, note that this DMFG algorithm equips a pair of $(\hat{Q}^j, \hat{v}^j, f^j)$ for each agent $j \in \mathfrak{N}$ to learn during the training process. This feature does incur tremendous computational costs as the amount of the population size N increases, especially if these estimations are trained given a pair of neural networks. This computation cost is both out of the thesis's physical capability and unfavourable for financial practice. To address this issue, we proposed a modified DMFG learning algorithm in chapter 5 with acceptable computational costs that mainly related to $K \ll N$ instead of N, and provided a comparison with the theoretical Nash equilibrium derived using Gateaux derivative in chapter 4.

Chapter 3

Generalized Optimal Liquidation Model Set-Up

3.1 Population of Agents

Consider the mean-field game described in [11] where the financial market consists of a population $\mathfrak{N}$ of $N > 0$ *rational heterogeneous* agents trading a *single* asset S. The total population $\mathfrak{N}$ can be split into $K < N$ disjoint sub-populations $\{\mathcal{K}_k\}_{k\in\mathfrak{K}}$, each of which has N_k agents with a specific investment objective and a particular market impact on the traded asset.

Each of the agents j in the k-th sub-population $\mathcal{K}_k$ starts with a random amount $\mathfrak{Q}_0^{j,k}$ of asset following a certain distribution $F^k(\cdot)$ varies according to their sub-population, and only controls the amount purchased or sold of the underlying asset at a continuous rate $\nu^j = (\nu_t^j)_{t\in[0,T]}$ over a fixed trading period $[0, T]$ from a set of admissible strategies

$$\mathcal{A}^j := \left\{ \nu^j : \nu^j \text{progressively measurable s.t., } \mathbb{E}\left[\int_0^T (\nu_s^j)^2 ds\right] < \infty \right\}. \tag{3.1.1}$$

Here $\nu^j > 0$ ($\nu^j < 0$) indicates the rate of buy (sell) orders the agent places in the market and we denote $\boldsymbol{\mathcal{A}} := \prod_{j=1}^N \mathcal{A}^j$ and $\boldsymbol{\nu} := (\nu^1, \cdots, \nu^N) \in \boldsymbol{\mathcal{A}}$ for notation convenience.

Different from Casgrain and Jaimungal, we assume that every agent in the market holds the identical and true belief over the asset price process $S^{\boldsymbol{\nu}}$ to simplify our analysis. In particular, every agent in the market knows that the risky asset $S^{\boldsymbol{\nu}} = (S_t^{\boldsymbol{\nu}})_{t\in[0,T]}$ is defined as

$$S_t^{\boldsymbol{\nu}} := F_t - Y_t^{\boldsymbol{\nu}}, \quad 0 \le t \le T \tag{3.1.2}$$

where $F = (F_t)_{t\in[0,T]}$ denotes some unaffected price process in $\mathcal{H}^2$ following the Vasicek process with some $\kappa, \eta > 0$ as

$$dF_t = \kappa(\eta - F_t)dt + \sigma dW_t, \tag{3.1.3}$$

and $Y = (Y_t^{\boldsymbol{\nu}})_{t\in[0,T]}$ denotes an aggregated linear and exponentially decaying price distortion from the unaffected price process F following the SDE

$$dY_t^{\boldsymbol{\nu}} = -\rho Y_t dt - \sum_{k=1}^K \lambda_k \bar{\nu}_t^k dt, \tag{3.1.4}$$

where ρ corresponding to the resiliency of the price distortion, $\lambda_k \ge 0$ corresponding to the scale of market impact of each sub-population $\mathcal{K}_k$, and $\bar{\nu}_t^k := \sum_{j \in \mathcal{K}_k} \nu_t^j$ corresponding to the average trading rate of all agents within sub-population $\mathcal{K}_k$.

Remark 3.1.1. Note that by assuming the unaffected price process F follows the Vasicek process, we implicitly assume that $F \in \mathcal{H}^2$, where $\mathcal{H}^2$ represents the class of all (special) semi-martingales P which ensures the existence of the canonical decomposition $P = \bar{M} + A$, for some local martingale $\bar{M}$ and a predictable finite-variation process $A = (A_t)_{t \in [0,T]}$ such that

$$\mathbb{E}\left[[\bar{M}]_T\right] + \mathbb{E}\left[\left(\int_0^T |dA_s|\right)^2\right] < \infty. \tag{3.1.5}$$

This specific feature would turn out to be critical to solve the examples in chapter 4.

Hence, the agents j in the sub-population $\mathcal{K}_k$ is able to keep track of their inventory process as $q^{v^j} = (q_t^{v^j})_{t \in [0,T]}$ such that

$$q_t^{\nu^j} = \mathfrak{Q}_0^{j,k} + \int_0^t \nu_s^j ds, \tag{3.1.6}$$

as well as their accumulated cash process $X^j = (X_t^j)_{t \in [0,T]}$, such that

$$X_t^j = \int_0^t -(S_s^{\boldsymbol{\nu}} + c_k \nu_s^j)\nu_s^j ds, \tag{3.1.7}$$

where $c^k > 0$ corresponds to the instantaneous transaction cost for each sub-population k. Further, each agent only observes the information generated by the paths of the asset price process $(S_t)_{t \in [0,T]}$, their own inventory process $(q_t^{j,v^j})_{t \in [0,T]}$ and accumulated cash process $(X_t^{\nu^j})_{t \in [0,T]}$.

To sum up, for an agent $j \in \mathcal{K}_k$, it is tracking its inventory process and accumulated cash process as

$$dq_t^{\nu^j} = \nu^j dt, \quad q_0^{\nu^j} = \mathfrak{Q}_0^{j,k} \tag{3.1.8}$$

$$dX_t^j = -(S^{\boldsymbol{\nu}_t} + c_k \nu_t^j)\nu_t^j dt, \quad X_0^{\nu^j} = 0, \tag{3.1.9}$$

and observing the price process

$$dS_t^{\boldsymbol{\nu}} = (\kappa(\eta - F_t) + \rho Y_t + \sum_{k \in \mathfrak{K}} \bar{\nu}_t^k)dt + dM_t, \quad S_0^{\bar{\nu}} = S_0. \tag{3.1.10}$$

Further, we denote agents' private state process as $\boldsymbol{x}^j = (\boldsymbol{x}_t^j)_{t \in [0,T]} := (t, S_t^{\boldsymbol{\nu}}, X_t^j, q_t^j)_{t \in [0,T]}$ and global state process $\boldsymbol{x} = (\boldsymbol{x}_t^j)_{t \in [0,T]} := (t, S_t^{\boldsymbol{\nu}}, \boldsymbol{X}_t, \boldsymbol{q}_t)_{t \in [0,T]}$, where $\boldsymbol{X}_t := (X_t^1, \cdots, X_t^N)$ and $\boldsymbol{q}_t := (q_t^1, \cdots, q_t^N)$ for notation convenience.

3.2 Objective Functions

With the above environment set up, the market has been transformed into a multi-agent game where each agent chooses their own trading policy to maximise an objective function, while simultaneously considering the impact of other agents' trading policies. Therefore, the problem has been shifted from finding the "optimal" trading strategy for each agent i to finding the Nash Equilibrium considering all of the market participants. Specifically, each agent j within a sub-population $\mathcal{K}_k$ chooses a trading policy ν^j to maximise an objective function (total expected reward at time 0)

$\mathcal{R}_0^j$ defined as

$$\mathcal{R}_0^j(\nu^j, \nu^{-j}) = \mathbb{E}\left[X_T^{\nu^j} + q_T^{j,\nu^j}\left(S_T^{\boldsymbol{\nu}} - \Phi_k q_T^{j,\nu^j}\right) - \phi_k \int_0^T (q_T^{j,\nu^j})^2 du\right], \tag{3.2.1}$$

And following the similar operation in [11],

$$\mathcal{R}_0^j(\nu^j, \nu^{-j}) = \mathbb{E}\left[\int_0^T -(S_t^{\boldsymbol{\nu}} + c_k \nu_t^j)\nu_t^j dt + q_T^{j,\nu^j}\left(S_T^{\boldsymbol{\nu}} - \Phi_k q_T^{j,\nu^j}\right) - \phi_k \int_0^T (q_T^{j,\nu^j})^2 du\right] \tag{3.2.2}$$

where $\Phi_k > 0$ and $\phi^k > 0$ are hyper-parameters describing the cost of liquidating all the leftover inventory at time T and the running risk aversion of holding inventory during the game as we did in the lectures, whereas $-j$ is used for convenience to refer to all other competing agents to agent j. For example, $\boldsymbol{\nu}^{-j} = (\nu^1, \cdots, \nu^{j-1}, \nu^{j+1}, \cdots, \nu^N)$ indicates the action policies of all other agents in the population.

In the next section, we will present the theoretical solutions and the sketched proofs for three different scenarios consist of the case of optimal liquidation for the single agent [19] when $k = 1$ and $N = 1$, the case of trading crowds [9] when $k = 1$ and $N > 1$, and the general case of mean-field game with the same belief [11] when $k > 1$ and $N > 1$, with some modifications for each example to fit in our model set-up.

Note that neither $\mathcal{R}_0^j$ nor r_t^j for all $t \leq T$ can be explicitly derived without full information $\boldsymbol{x}$ of the environment. Further, the derived theoretical solutions do not guarantee whether the self-interested agents would reach the Nash equilibrium as we indicated in section 2.3. These limitations motivate us to construct and examine our modified decentralized mean-field Game Q-learning algorithm in chapter 5 with the theoretical performance in the cases above.

Chapter 4

Algorithmic Trading and Mean-Field Game Solutions

4.1 Preliminary: Signal of the Vasicek Process

Before we dive into the details, it would be convenient to first decompose the the unaffected price process F as suggested in remark 3.1.1.

Since F follows the Vasicek process, we know that the closed form solution for the SDE described in equation (3.1.3) is [35].

$$F_s = F_t e^{-\kappa(s-t)} + \eta(1 - e^{-\kappa(s-t)}) + \sigma \int_s^t e^{-\kappa(u-s)} dW_u, \tag{4.1.1}$$

for any $0 < t < s < T$. Hence the Vasicek model F, as a semi-martingale in $\mathcal{H}^2$ [36], can be decomposed into $F = \bar{M} + A$, such that

$$A_s = F_t e^{-\kappa(s-t)} + \eta(1 - e^{-\kappa(s-t)}) \tag{4.1.2}$$

$$\bar{M}_t = \sigma \int_s^t e^{-\kappa(u-s)} dW_u, \tag{4.1.3}$$

for any $t \in [0, T]$. Therefore, we may obtain

$$dA_s = \kappa(\eta - F_t) e^{-\kappa(s-t)} ds, \tag{4.1.4}$$

given the filtration information $\mathcal{F}_t$ up to time stage t, for any $0 < t < s < T$. Intuitively, dA_s provides the signal indicating the trend of the unaffected price F given the current price F_t. In particular, if the current price is greater to the mean it is reverting to, i.e., $F_t > \eta$, then there is a downward pressure on the unaffected price amplified by the speed of reversion κ. And vice versa.

This equation (4.1.4) would turn out to help solve the optimal liquidation and the trading with crowds problems, as we will see in the following subsections.

4.2 Optimal Liquidation for Single Agent

In this subsection, we provide the optimal trading strategy obtained by the Gateaux derivative when there are only one sub-population and a single trading agent, i.e., the scenario when $K = 1$ and $N = 1$.

First, to simplify our analysis, we may rewrite the objective function in equation (3.2.1) as

$$\mathcal{R}_0(\nu) = \mathbb{E}\left[\int_0^T -(F_t + Y_t^\nu)\nu_t dt - \int_0^T c\nu^2 dt + q_T\,(S_T^\nu - \Phi q_T^\nu) - \phi \int_0^T (q_T^\nu)^2 du\right], \tag{4.2.1}$$

to fit the single agent case. This objective function, as designed, matches [19, eqn. 2.6] with slight difference in the notation.

Remark 4.2.1. Note that though the transaction cost $c > 0$ does not generate the temporary price impact as it is in the setting of Neuman and Voss, it provides the same paid price deduction from the perspective of the agent. By the nature of the temporary price, the transaction cost c in my paper serves the same property as the temporary price impact in Neuman and Voss.

Therefore, we obtain the exact same forward backward SDE system described in [19, Lemma 5.2].

Lemma 4.2.2 (A simplified version of Neuman and Voss's Lemma 5.2 [19])**.** *The control $\nu \in \mathcal{A}$ provides the optimal liquidation strategy if and only if the processes $(\nu, X^\nu, Y^\nu, Z^\nu)$ satisfy the following coupled linear FBSDE system*

$$\begin{cases} dq_t^\nu &= \nu_t dt, \quad q_0^\nu = q \\ dY_t^\nu &= -\rho Y_t^\nu dt - \lambda \nu_t dt, \quad Y_0^\nu = y \\ d\nu_t &= \frac{1}{2c}\left(dF_t - dY_t^{\nu^*} - dZ_t^{\nu^*} + d\tilde{M}_t - 2\phi q_t^{\nu^*} dt\right), \quad \nu_T = \frac{\Phi}{c} q_T^\nu - \frac{1}{2c} Y_T^\nu \\ dZ_t^\nu &= \nu_t dt - d\tilde{N}_t, \quad Z_T^\nu = 0 \end{cases} \tag{4.2.2}$$

for suitable squre integrable martingales $\tilde{M} = (\tilde{M}_t)_{t\in[0,T]}$ and $\tilde{N} = (\tilde{N}_t)_{t\in[0,T]}$. Furthermore, the FBSDE system in equation (4.2.2) has a unique solution.

Proof. See appendix A.1.1. □

This solution to the equation (4.2.2) in this paper is a specific case in [19]. Therefore, by [19, Theorem 3.2], we have the following theorem.

Theorem 4.2.3 (A simplified version of Neuman and Voss's Theorem 3.2 [19])**.** *Denote the auxiliary matrix*

$$L := \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & -\rho & 1 & 0 \\ -\frac{\phi}{c} & -\frac{\rho}{2c} & 0 & \frac{\rho}{2c} \\ 0 & 0 & 1 & \rho \end{bmatrix} \in \mathbb{R}^{4\times 4}, \tag{4.2.3}$$

and the functions $S(t) := [S_{ij}(t)]_{1\le i,j\le 4}$ and $G(t) := (G_i(t))_{1\le i\le 4}$ such that

$$S(t) := e^{Lt} \tag{4.2.4}$$

$$G(t) := \begin{bmatrix} \frac{\Phi}{c} & -\frac{1}{2c} & -1 & 0 \end{bmatrix} S(t), \tag{4.2.5}$$

for $t > 0$. Further, let

$$\begin{aligned} v_0(t) &:= \left(1 - \frac{G_4(t)}{G_3(t)}\frac{S_{4,3}(t)}{S_{4,4}(t)}\right)^{-1} & v_1(t) &:= \frac{G_4(t)}{G_3(t)}\frac{S_{4,1}(t)}{S_{4,4}(t)} - \frac{G_1(t)}{G_3(t)} \\ v_2(t) &:= \frac{G_4(t)}{G_3(t)}\frac{S_{4,2}(t)}{S_{4,4}(t)} - \frac{G_2(t)}{G_3(t)} & v_3(t) &:= \frac{G_4(t)}{G_3(t)} \end{aligned} \tag{4.2.6}$$

for $t > 0$. Then under the assumption that F follows the Vasicek process, there exists a unique optimal strategy $\nu^* \in \mathcal{A}$ such that

$$\begin{aligned} \nu_t^* =& v_0(T-t)\Big[v_1(T-t)q_t^{\nu^*} + v_2(T-t)Y_t^{\nu^*} \\ &+ \frac{\kappa(\eta - F_t)}{2c}\left(v_3(T-t)\int_t^T e^{-\kappa(s-t)}\frac{S_{4,3}(T-s)}{S_{4,4}(T-t)}ds - \int_t^T e^{-\kappa(s-t)}\frac{G_3(T-s)}{G_3(T-t)}ds\right)\Big] \end{aligned} \tag{4.2.7}$$

Proof. See appendix A.1.2. □

Remark 4.2.4. Although the optimal trading strategy ν^* depends on the current unaffected price F_t given the filtration $\mathcal{F}_t$, but given the parameters in table 5.1 and table 5.2 during the experiments, the impact is oblivious. More analysis in that can be found in section 5.2.1.

4.3 Trading with Crowds

In this subsection, we provides the Nash equilibrium obtained by the Gateaux derivative when there are only one sub-population but $N > 1$ trading agents, i.e., the scenario when $K = 1$ and $N > 1$.

First, to simplify our analysis, we may rewrite the objective function in equation (3.2.1) as

$$\mathcal{R}_0^j(\nu^j, \nu^{-j}) = \mathbb{E}\left[X_T^{\nu^j} + q_T^{j,\nu^j}\left(S_T^{\boldsymbol{\nu}} - \Phi q_T^{j,\nu^j}\right) - \phi\int_0^T (q_T^{j,\nu^j})^2 du\right], \tag{4.3.1}$$

to fit the trading with crowds case. Again, we can quick realize that the objective function described in equation (4.3.1) matches to what is defined in [9, eqn. 2.6], with slightly difference in the notation.

Similarly, we obtain the following forward backward SDE system as described in [9, Lemma 2.5].

Lemma 4.3.1 (A simplified version of Neuman and Voss's lemma 2.5. in [9]). *A set of controls $(\nu^j)_{j\in\mathfrak{N}}$ yields the unique Nash equilibrium if and only if the processes $(\nu^j, X^{\nu^j}, Y^{\nu^j}, Z^{\nu^j})$, for $j \in \mathfrak{N}$, satisfy the following coupled linear forward backward SDE system*

$$\begin{cases} dq_t^{\nu^j} &= \nu_t^j dt, \quad q_0^{\nu^j} = q^j \\ dY_t^{\boldsymbol{\nu}} &= -\rho Y_t^{\boldsymbol{\nu}} dt - \frac{\lambda}{N}\sum_{\in\mathfrak{N}} \nu_t^i dt, \quad Y_0^{\boldsymbol{\nu}} = y \\ d\nu_t^j &= \frac{1}{2c}\left(dF_t - dY_t^{\boldsymbol{\nu}} - dZ_t^{\nu^j} + d\tilde{M}_t - 2\phi q_t^{\nu^j} dt\right), \quad \nu_T^j = \frac{\Phi}{c} q_T^{\nu^j} - \frac{1}{2c} Y_T^{\boldsymbol{\nu}} \\ dZ_t^{\nu^j} &= \rho Z_t^{\nu^j} dt + \frac{\lambda}{N}\nu_t^j dt - d\tilde{N}_t, \quad Z_0^{\nu^j} = 0 \end{cases} \tag{4.3.2}$$

for suitable squre integrable martingales $\tilde{M}^j = (\tilde{M}_t^j)_{t\in[0,T]}$ and $\tilde{N}^j = (\tilde{N}_t^j)_{t\in[0,T]}$, $j \in \mathfrak{N}$. Furthermore, the FBSDE system in equation (4.3.2) has a unique solution.

Proof. See appendix A.2.1. □

To solve the FBSDE system, we would need to obtain the averaged version of the Nash equilibrium $(\bar{\nu}, \bar{q}^{\bar{\nu}}, \bar{Y}^{\bar{\nu}}, \bar{Z}^{\bar{\nu}})$ first, which motivates us to obtain the decoupled version of equation (4.3.2) as follows.

Corollary 4.3.2. *The mean-field controls $\bar{\nu}$ yields the unique Nash equilibrium if and only if the mean-field process $(\bar{\nu}, \bar{q}^{\bar{\nu}}, \bar{Y}^{\bar{\nu}}, \bar{Z}^{\bar{\nu}}) := \frac{1}{N}\sum_{i\in\mathfrak{N}}(\nu^j, q^{\nu^j}, Y^{\boldsymbol{\nu}}, Z^{\nu^j})$ satisfy the following coupled linear* *forward backward SDE system*

$$\begin{cases} d\bar{q}_t^{\bar{\nu}} &= \bar{\nu}_t dt, \quad \bar{q}_0^{\bar{\nu}} = q^j \\ d\bar{Y}_t^{\bar{\nu}} &= -\rho \bar{Y}_t^{\bar{\nu}} dt - \lambda \bar{\nu}_t dt, \quad Y_0^{\bar{\nu}} = y \\ d\bar{\nu}_t &= \frac{1}{2c}\left(dF_t - dY_t^{\boldsymbol{\nu}} - d\bar{Z}_t^{\bar{\nu}} + d\bar{M}_t - 2\phi \bar{q}_t^{\bar{\nu}} dt\right), \quad \bar{\nu}_T = \frac{\Phi}{c}\bar{q}_T^{\bar{\nu}} - \frac{1}{2c} Y_T^{\boldsymbol{\nu}} \\ d\bar{Z}_t^{\bar{\nu}} &= \rho \bar{Z}_t^{\bar{\nu}} dt + \frac{\lambda}{N}\bar{\nu}_t dt - d\bar{N}_t, \quad Z_0^{\nu^j} = 0 \end{cases} \tag{4.3.3}$$

for suitable squre integrable martingales $\bar{M} = (\bar{M}_t)_{t\in[0,T]}$ and $\bar{N}^j = (\bar{N}_t)_{t\in[0,T]}$. Furthermore, the FBSDE system in equation (4.3.3) *has a unique solution.*

Proof. See appendix A.2.2. □

To solve the equation (4.3.2), we need to first solve the equation (4.3.3), which again is a specific case in [9]. Therefore, by [9, Proposition 2.9], we have

Corollary 4.3.3. *Denote the auxiliary matrix*

$$\bar{L} := \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & -\rho & \lambda & 0 \\ -\frac{\phi}{c} & \frac{\rho}{2c} & -\frac{\lambda(N-1)}{2cN} & \frac{\rho}{2c} \\ 0 & 0 & \frac{\lambda}{N} & \rho \end{bmatrix} \in \mathbb{R}^{4\times 4}, \tag{4.3.4}$$

and the functions $\bar{S}(t) := [\bar{S}_{ij}(t)]_{1\le i,j\le 4}$ and $\bar{G}(t) := (\bar{G}_i(t))_{1\le i\le 4}$ such that

$$\bar{S}(t) := e^{\bar{L}t} \tag{4.3.5}$$

$$\bar{G}(t) := \begin{bmatrix} \frac{\Phi}{c} & -\frac{1}{2c} & -1 & 0 \end{bmatrix} \bar{S}(t), \tag{4.3.6}$$

for $t > 0$. Further, let

$$\begin{aligned} \bar{v}_0(t) &:= \left(1 - \frac{\bar{G}_4(t)}{\bar{G}_3(t)}\frac{\bar{S}_{4,3}(t)}{\bar{S}_{4,4}(t)}\right)^{-1} & \bar{v}_1(t) &:= \frac{\bar{G}_4(t)}{\bar{G}_3(t)}\frac{\bar{S}_{4,1}(t)}{\bar{S}_{4,4}(t)} - \frac{\bar{G}_1(t)}{\bar{G}_3(t)} \\ \bar{v}_2(t) &:= \frac{\bar{G}_4(t)}{\bar{G}_3(t)}\frac{\bar{S}_{4,2}(t)}{\bar{S}_{4,4}(t)} - \frac{\bar{G}_2(t)}{\bar{G}_3(t)} & \bar{v}_3(t) &:= \frac{\bar{G}_4(t)}{\bar{G}_3(t)} \end{aligned} \tag{4.3.7}$$

for $t > 0$. Then under the assumption that F follows the Vasicek process, there exists a unique Nash equilibrium $\bar{\nu}^* \in \bar{\mathcal{A}}$, where $\bar{\mathcal{A}} := \left\{\sum_{j\in\mathfrak{N}} \nu^j : \nu^j \in \mathcal{A}^j\right\}$, such that

$$\begin{aligned} \bar{\nu}_t^* =& \bar{v}_0(T-t)\Big[\bar{v}_1(T-t)\bar{q}_t^{\bar{\nu}^*} + \bar{v}_2(T-t)\bar{Y}_t^{\bar{\nu}^*} \\ &+ \frac{\kappa(\eta - F_t)}{2c}\left(\bar{v}_3(T-t)\int_t^T e^{-\kappa(s-t)}\frac{\bar{S}_{4,3}(T-s)}{\bar{S}_{4,4}(T-t)}ds - \int_t^T e^{-\kappa(s-t)}\frac{\bar{G}_3(T-s)}{\bar{G}_3(T-t)}ds\right)\Big] \end{aligned} \tag{4.3.8}$$

Proof. See appendix A.2.3. □

Remark 4.3.4. Note the similarity between theorem 4.2.3 and corollary 4.3.3. The only difference comes from the third and forth rows of the base matrix L and $\bar{L}$. Specifically, $L = \bar{L}$ when $N = 1$.

Intuitively, one may regard the mean-field process $(\bar{\nu}, \bar{q}^{\bar{\nu}}, \bar{Y}^{\bar{\nu}}, \bar{Z}^{\bar{\nu}})$ as a *giant representative* of the sub-population executing the mean-field action $\bar{\nu}$ producing the price impact $\bar{Y}^{\bar{\nu}}$ while tracking its own mean-field process $\bar{q}^{\bar{\nu}}$. However, its action would be slightly influenced by its component agents. Hence, you may regard the representative as a single agent optimal liquidation problem as we solved in theorem 4.2.3, with slightly difference as a result of the interactions within its component agents.

Therefore, following the Neuman and Voss in [9, prop. 2.9], we obtain the following theorem.

Theorem 4.3.5. *Denote the auxiliary matrix*

$$L := \begin{bmatrix} 0 & -1 & 0 \\ -\frac{\phi}{c} & -\frac{\lambda}{2cN} & \frac{\rho}{2c} \\ 0 & \frac{\lambda}{N} & \rho \end{bmatrix} \in \mathbb{R}^{3\times 3}, \tag{4.3.9}$$

and the functions $S(t) := [S_{ij}(t)]_{1\le i,j\le 3}$ and $G(t) := (G_i(t))_{1\le i\le 3}$ such that

$$S(t) := e^{Lt} \tag{4.3.10}$$

$$G(t) := \begin{bmatrix} \frac{\Phi}{c} & -1 & 0 \end{bmatrix} S(t), \tag{4.3.11}$$

for $t > 0$. Further, let

$$\begin{aligned} v_0(t) &:= \left(1 - \frac{G_3(t)}{G_2(t)}\frac{S_{3,2}(t)}{S_{3,3}(t)}\right)^{-1} \qquad v_1(t) := \frac{G_3(t)}{G_2(t)}\frac{S_{3,1}(t)}{S_{3,3}(t)} - \frac{G_1(t)}{G_2(t)} \\ v_2(t) &:= \frac{G_3(t)}{G_2(t)} \end{aligned} \tag{4.3.12}$$

for $t > 0$. Then under the assumption that F follows the Vasicek process, there exists a unique optimal Nash equilibrium $\boldsymbol{\nu}^* = (\nu^{j,*})_{j\in\mathfrak{N}} \in \mathcal{A}$ such that

$$\begin{aligned} \nu_t^{j,*} =& v_0(T-t)\Bigg[v_1(T-t)q_t^{\nu^{j,*}} - \frac{1}{2cG_2(T-t)}\bar{Y}_T^{\bar{\nu}^*} \\ &+ \frac{\kappa(\eta - F_t)}{2c}\left(v_2(T-t)\int_t^T e^{-\kappa(s-t)}\frac{S_{3,2}(T-s)}{S_{3,3}(T-t)}ds - \int_t^T e^{-\kappa(s-t)}\frac{G_3(T-s)}{G_3(T-t)}ds\right) \\ &- \frac{1}{2c}\left(v_2(T-t)\int_t^T \frac{S_{3,2}(T-s)}{S_{3,3}(T-t)}d\bar{Y}_s - \int_t^T \frac{G_3(T-s)}{G_3(T-t)}d\bar{Y}_s\right)\Bigg] \end{aligned} \tag{4.3.13}$$

Proof. See Appendix appendix A.2.3. □

Note that since $\bar{Y}^{\bar{\nu}^*}$ is deterministic (see also in appendix A.2.4) with finite variation, we know that the integral part with respect to $d\bar{Y}_s$ in equation (4.3.13) can be simplified further to a form with only ds term. However, since $\bar{Y}^{\bar{\nu}^*}$ is needed to know beforehand to compute the $\nu^{j,*}$, the current form would be simple enough for numerically obtaining $\nu^{j,*}$ and actually produce less errors due to floating point. Hence, we would regard theorem 4.3.5 as it is.

4.4 Mean-Field Games with the Same True belief

In this subsection, we provides the analysis by the Gateaux derivative when there are $K > 1$ sub-populations and $N > 1$ trading agents in total, i.e., the general scenario of our model set-up when $K > 1$ and $N_k \ge 1$ for any k. Although this is a simplified version of Casgrain and Jaimungal in [11] to some degree, note that the extra feature on the exponentially decaying transient price impact process Y also extends the complexity of solving the problem.

Note that for each agent $j \in \mathfrak{N}$, the analysis for the concaveness of the objective function in equation (3.2.1) does not differ from the trading crowds scenario. Therefore, similarly to the argument in section 4.3, it admits a unique maximizer characterized by the critical point at which the Gateaux derivative vanishes for any direction in $\mathcal{A}^j$.

Lemma 4.4.1. *A set of controls $(\nu^j)_{j\in\mathfrak{N}}$ yields the unique Nash equilibrium if and only if the processes $(\nu^j, X^{\nu^j}, Y^{\nu^j}, Z^{\nu^j})$, for $j \in \mathcal{K}_k$, satisfy the following coupled linear forward backward*

SDE system

$$\begin{cases} dq_t^{\nu^j} &= \nu_t^j dt, \quad X_0^{\nu^j} = x^j \\ dY_t^{\boldsymbol{\nu}} &= -\rho Y_t^{\boldsymbol{\nu}} dt - \sum_{k\in\mathfrak{K}} \frac{\lambda_k}{N_k} \sum_{i\in\mathcal{K}_k} \nu_t^i dt, \quad Y_0^{\boldsymbol{\nu}} = y \\ d\nu_t^j &= \frac{1}{2c}\left(dF_t - dY_t^{\boldsymbol{\nu}} - dZ_t^{\nu^j,k} + d\tilde{M}_t^{j,k} - 2\phi_k q_t^{\nu^j} dt\right), \quad \nu_T^j = \frac{\Phi_k}{c_k} q_T^{\nu^j} - \frac{1}{2c_k} Y_T^{\nu} \\ dZ_t^{\nu^j,k} &= \rho Z_t^{\nu^j,k} dt + \frac{\lambda_k}{N_k} \nu_t^j dt - d\tilde{N}_t^{j,k}, \quad Z_0^{\nu^j} = 0 \end{cases} \tag{4.4.1}$$

for suitable squre integrable martingales $\tilde{M}^{j,k} = (\tilde{M}_t^{j,k})_{t\in[0,T]}$ and $\tilde{N}^{j,k} = (\tilde{N}_t^{j,k})_{t\in[0,T]}$, $j \in \mathcal{K}_k$. Furthermore, the FBSDE system in equation (4.3.2) has a unique solution.

Proof. See appendix A.3.1. □

Similarly, we obtain the following forward backward SDE system as follows.

Corollary 4.4.2. *A set of sub-mean-field controls $(\bar{\nu}^k)_{k\in\mathfrak{K}}$ yields the unique Nash equilibrium if and only if the mean-field process $(\bar{\nu}^k, \bar{q}^{\bar{\nu}^k}, \bar{Y}^{\bar{\nu}^k}, \bar{Z}^{\bar{\nu}^k}) := \frac{1}{N_k}\sum_{i\in\mathcal{K}_k}(\nu^j, q^{\nu^j}, Y^{\boldsymbol{\nu}}, Z^{\nu^j})$ satisfy the following coupled linear forward backward SDE system*

$$\begin{cases} d\bar{q}_t^{\bar{\nu}^k} &= \bar{\nu}_t^k dt, \quad \bar{q}_0^{\bar{\nu}^k} = \bar{q}^k \\ d\bar{Y}_t^{\bar{\nu}^k} &= -\rho \bar{Y}_t^{\bar{\nu}^k} dt - \sum_{k'\in\mathfrak{K}} \lambda_{k'} \bar{\nu}_t^{k'} dt, \quad Y_0^{\bar{\nu}^k} = y \\ d\bar{\nu}_t^k &= \frac{1}{2c}\left(dF_t - dY_t^{\boldsymbol{\nu}^k} - d\bar{Z}_t^{\bar{\nu}^k} + d\bar{M}_t^k - 2\phi \bar{q}_t^{\bar{\nu}^k} dt\right), \quad \bar{\nu}_T^k = \frac{\Phi}{c_k}\bar{q}_T^{\bar{\nu}^k} - \frac{1}{2c_k} Y_T^{\boldsymbol{\nu}^k} \\ d\bar{Z}_t^{\bar{\nu}^k} &= \rho \bar{Z}_t^{\bar{\nu}^k} dt + \frac{\lambda_k}{N_k}\bar{\nu}_t^k dt - d\bar{N}_t^k, \quad Z_0^{\nu^j} = 0 \end{cases} \tag{4.4.2}$$

for suitable squre integrable martingales $\bar{M} = (\bar{M}_t)_{t\in[0,T]}$ and $\bar{N}^j = (\bar{N}_t)_{t\in[0,T]}$. Furthermore, the FBSDE system in equation (4.3.3) has a unique solution.

However, notice that we may still unable to solve the FBSDE system in corollary 4.4.2 as it is still coupled indicated by the $\bar{Y}_t^{\bar{\nu}^k}$ term. To this end, denote the distribution (or weight) of each sub-population $\mathcal{K}_k$ as

$$p_k := \frac{N_k}{N} \in (0,1), \quad \sum_{k\in\mathfrak{K}} p_k = 1, \tag{4.4.3}$$

we finally obtain the following decoupled FBSDE system.

Corollary 4.4.3. *The mean-field controls $\bar{\nu}$ yields the unique Nash equilibrium if and only if the mean-field process $(\bar{\nu}, \bar{q}^{\bar{\nu}}, \bar{Y}^{\bar{\nu}}, \bar{Z}^{\bar{\nu}}) := \sum_{k'\in\mathfrak{K}} p_{k'}(\lambda_{k'}\bar{\nu}^{k'}, q^{\bar{\nu}^{k'}}, Y^{\bar{\nu}^{k'}}, Z^{\bar{\nu}^{k'}})$ satisfy the following coupled linear forward backward SDE system*

$$\begin{cases} d\bar{q}_t^{\bar{\nu}} &= \bar{\nu}_t dt, \quad \bar{q}_0^{\bar{\nu}} = \bar{q} \\ d\bar{Y}_t^{\bar{\nu}} &= -\rho \bar{Y}_t^{\bar{\nu}} dt - \bar{\nu}_t dt, \quad Y_0^{\bar{\nu}} = y \\ d\bar{\nu}_t &= \frac{1}{2c}\left(dF_t - dY_t^{\boldsymbol{\nu}} - d\bar{Z}_t^{\bar{\nu}} + d\bar{M}_t - 2\phi \bar{q}_t^{\bar{\nu}} dt\right), \quad \bar{\nu}_T = \frac{\Phi}{c_k}\bar{q}_T^{\bar{\nu}} - \frac{1}{2c_k} Y_T^{\boldsymbol{\nu}} \\ d\bar{Z}_t^{\bar{\nu}} &= \rho \bar{Z}_t^{\bar{\nu}} dt + \frac{1}{N}\bar{\nu}_t dt - d\bar{N}_t, \quad Z_0^{\nu^j} = 0 \end{cases} . \tag{4.4.4}$$

Remark 4.4.4. Notice that the distinct pairs (N_k, λ_k) over all $k \in \mathfrak{K}$ becomes an obstacle when we decouple the FBSDE in equation (4.4.2). To this end, we have to define the mean-field strategy $\bar{\nu}$ as the weighted average over the K sub-populations weighted by their sub-population size N_k and market price impact λ_k, for each $k \in \mathfrak{K}$. Similar to our intuition in remark 4.3.4, this modified structure can be viewed as a giant representative executing the mean-field action $\bar{\nu}'$ with some market price impact $\bar{\lambda}$, or the mean-field action $\bar{\nu}$ as we defined above, but re-scaled to an standardized space. However, bear in mind that $\bar{\lambda}$ is not constant with respect to $(\bar{\nu}^k)_{k\in\mathfrak{K}}$. Therefore, we decide to recommend the second interpretation and notation.

Notice the similarity of the decoupled FBSDE in corollary 4.4.3 and corollary 4.3.2. Hence, we can immediately obtain the following result with $\lambda = 1$.

Corollary 4.4.5. *Denote the auxiliary matrix*

$$\bar{L} := \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & -\rho & 1 & 0 \\ -\frac{\phi}{c} & \frac{\rho}{2c} & -\frac{(N-1)}{2cN} & \frac{\rho}{2c} \\ 0 & 0 & \frac{1}{N} & \rho \end{bmatrix} \in \mathbb{R}^{4\times 4}, \tag{4.4.5}$$

and the functions $\bar{S}(t) := [\bar{S}_{ij}(t)]_{1\leq i,j\leq 4}$ *and* $\bar{G}(t) := (\bar{G}_i(t))_{1\leq i\leq 4}$ *such that*

$$\bar{S}(t) := e^{\bar{L}t} \tag{4.4.6}$$

$$\bar{G}(t) := \begin{bmatrix} \frac{\Phi}{c} & -\frac{1}{2c} & -1 & 0 \end{bmatrix} \bar{S}(t), \tag{4.4.7}$$

for $t > 0$. *Further, let*

$$\begin{aligned} \bar{v}_0(t) &:= \left(1 - \frac{\bar{G}_4(t)}{\bar{G}_3(t)} \frac{\bar{S}_{4,3}(t)}{\bar{S}_{4,4}(t)}\right)^{-1} & \bar{v}_1(t) &:= \frac{\bar{G}_4(t)}{\bar{G}_3(t)} \frac{\bar{S}_{4,1}(t)}{\bar{S}_{4,4}(t)} - \frac{\bar{G}_1(t)}{\bar{G}_3(t)} \\ \bar{v}_2(t) &:= \frac{\bar{G}_4(t)}{\bar{G}_3(t)} \frac{\bar{S}_{4,2}(t)}{\bar{S}_{4,4}(t)} - \frac{\bar{G}_2(t)}{\bar{G}_3(t)} & \bar{v}_3(t) &:= \frac{\bar{G}_4(t)}{\bar{G}_3(t)} \end{aligned} \tag{4.4.8}$$

for $t > 0$. *Then under the assumption that* F *follows the Vasicek process, there exists a unique Nash equilibrium* $\bar{\nu}^* \in \bar{\mathcal{A}}$, *where* $\bar{\mathcal{A}} := \left\{\sum_{j\in\mathfrak{N}} \nu^j : \nu^j \in \mathcal{A}^j\right\}$, *such that*

$$\begin{aligned} \bar{\nu}_t^* =& \bar{v}_0(T-t) \Big[\bar{v}_1(T-t)\bar{q}_t^{\bar{\nu}^*} + \bar{v}_2(T-t)\bar{Y}_t^{\bar{\nu}^*} \\ &+ \frac{\kappa(\eta - F_t)}{2c} \left(\bar{v}_3(T-t) \int_t^T e^{-\kappa(s-t)} \frac{\bar{S}_{4,3}(T-s)}{\bar{S}_{4,4}(T-t)} ds - \int_t^T e^{-\kappa(s-t)} \frac{\bar{G}_3(T-s)}{\bar{G}_3(T-t)} ds\right)\Big] \end{aligned} \tag{4.4.9}$$

Remark 4.4.6. Note that the result in corollary 4.4.5 gives limited insight about the *unscaled* mean-field trading strategy $\bar{\nu}'$. Thus, we did not implement and plot it as the benchmark for the major-minor example in section 5.2.

Notice that the resulting Nash equilibrium $\bar{\nu}^*$ in corollary 4.4.5 is deterministic, suggesting that the FBSDE systems proposed in corollary 4.4.2 and lemma 4.4.1 can be solved progressively and the results would be deterministic as well.

As proposed in corollary 4.4.2 and indicated in [11], such a coupled FBSDE system is too complex to solve explicitly. One widely used approach in the mean-field games, as proposed by Casgrain and Jaimungal, is to restructure the finite-agent game into a infinite-agent game, such that $N \to \infty$ with $p_k \in (0,1)$ unchanged. Instead of solving the individual trading strategies, this approach would solve the *finite sub-population representative* mean-field game with scaled price impact. In particular, each negligible individual in the infinite-agent game would behave closely to its sub-population's mean-field actions within some boundary ϵ in the equilibrium, and such a Nash equilibrium is called ϵ-Nash equilibrium. After that, the attained Nash equilibrium can be shown to be an well-defined approximation for the Nash equilibrium of the finite-agent game, within some error-tolerance $\epsilon > 0$. However, as it is out of the scope of the paper, we would not dive into the details of solving this ϵ-Nash equilibrium but instead suggest interested readers to consult [9, 11] for more details.

Chapter 5

Decentralized Mean-Field Game Solutions

Although the mean-field game method provides a straightforward solution for most of the regularly defined optimal execution problems, the mathematical complexity becomes a barrier for practitioners to apply the technique in the industry. Further, even though the existence and uniqueness of the Nash equilibrium can be shown, the FBSDE systems are sometimes too complex to solve explicitly, while infinite-agent games are somewhat not precisely what we want. Also, it does not guarantee whether the game process would achieve the Nash equilibrium during the game. Therefore, motivated by [12, 18], we considered and modified the decentralized learning algorithm to simulate the learning process and reached the quasi-Nash equilibrium with acceptable computational cost.

5.1 Framework and Algorithms

To characterize the mean-field game problem as a multi-agent stochastic game $\langle S^j, \mathcal{A}^j, r^j, p, \gamma \rangle_{j \in \mathbb{N}}$, we must first discretize the objective functional in equation (3.2.1) and equation (3.2.2) into

$$\begin{aligned} \mathcal{R}_0^j(\nu^j, \nu^{-j}) &= \mathbb{E}\left[X_T^{\nu^j} + q_T^{j,\nu^j} \left(S_T^{\boldsymbol{\nu}} - \Phi_k q_T^{j,\nu^j} \right) - \phi_k \sum_{t=0}^{T} (q_T^{j,\nu^j})^2 dt \right] \\ &= \mathbb{E}\left[\sum_{t=0}^{T} -(S_t^{\boldsymbol{\nu}} + c_k \nu_t^j) \nu_t^j dt + q_T^{j,\nu^j} \left(S_T^{\boldsymbol{\nu}} - \Phi_k q_T^{j,\nu^j} \right) - \phi_k \sum_{t=0}^{T} (q_T^{j,\nu^j})^2 dt \right] \end{aligned} \tag{5.1.1}$$

where $dt > 0$ is a fixed real number, and then define the function $\boldsymbol{r} = (r_1, \cdots, r_N)$, in which

$$r^j : S \times \mathcal{A} \to \mathbb{R} \tag{5.1.2}$$

provides the reward for each agent j at each state x_t given every market participator's action $\boldsymbol{\nu} = (\nu^j, \boldsymbol{\nu}^{-j})$, such that

$$\mathcal{R}_0^j(\nu^j, \boldsymbol{\nu}^{-j}) = \mathbb{E}\left[\sum_{t=0}^{T} r^j(x_t, \nu_t^j, \boldsymbol{\nu}_t^{-j}) \right]. \tag{5.1.3}$$

Here the state process x follows $\{x_0 \sim p_0, x_{t+1} \sim p(\cdot | x_t, \boldsymbol{\nu}_t), t \in [0, T]\}$ with $p_0 \in \Delta(\mathcal{X})$ as the initial state distribution, as we introduced in chapter 2.

Remark 5.1.1. Notice that while the range of selections for $\boldsymbol{r}$ provides an extra degree of freedom in solving the multi-agent games, the choice we made may also impact the learning efficiency and sometimes even the convergence (e.g., see reward shaping in [37, 38]). A concrete and related example can be found in the course project by Somner-Bogard and Wang [39].

Following the majority of the literature (e.g., see [12]), here we define the *reward function* r^j as

$$r^j(x_t, \nu_t^j, \boldsymbol{\nu}_t^{-j}) = \begin{cases} -(S_t^{\boldsymbol{\nu}} + c_k \nu_t^j)\nu_t^j dt, & \text{if } t < T \\ -(S_T^{\boldsymbol{\nu}} + c_k \nu_T^j)\nu_T^j dt + q_T^{\nu^j}(S_T^{\boldsymbol{\nu}} - \Phi q_T^{\nu^j}), & \text{otherwise} \end{cases}, \quad \forall\, 0 \le t \le T \tag{5.1.4}$$

for any $j \in \mathfrak{N}$, to characterize the immediate reward when $t < T$ as well as the terminal reward when $t = T$. Theoretically, the agents j would be able to capture the jump of the terminal reward governed by the extended Bellman equation for Nash equilibria $(\nu^{j,*}, \boldsymbol{\nu}^{-j,*})$ in [12, eqn. 3.1]

$$\mathcal{R}_t^j(\nu^{j,*}, \boldsymbol{\nu}^{-j,*}; x) = \max_{\nu_t^j \in \mathcal{A}_t^j} \left\{ r^j(\nu_t^j, \boldsymbol{\nu}_t^{-j,*}; x_t) + \mathbb{E}_{x_{t+1} \sim p(\cdot|x_t;\boldsymbol{\nu}_t)} \left[\mathcal{R}_{t+1}^j(\nu^{j,*}, \boldsymbol{\nu}^{-j,*}; x') \right] \right\}, \tag{5.1.5}$$

where by abuse of notation, we denote $\mathcal{R}_t^j(\nu^j, \boldsymbol{\nu}^{-j}; x_t) := \mathbb{E}\left[\sum_{s=t}^T r^j(x_s, \nu_s^j, \boldsymbol{\nu}_s^{-j})\right]$ as the expected total reward since after time stage t given $\boldsymbol{\nu}$ and state process x, and $\boldsymbol{\nu}^{-j,*}$ left as fixed at the perspective of the agents j. Following the literature in reinforcement learning, here we define the *value function* as

$$V^j(x_t) := \mathcal{R}_t^j(\nu^{j,*}, \boldsymbol{\nu}^{-j,*}; x_t) \tag{5.1.6}$$

which represents the value of each state x_t at their Nash equilibria, as well as the *state-action value function*, also often called the *Q-Function*, as

$$Q^j(x_t; \boldsymbol{\nu_t}) := r^j(x_t; \boldsymbol{\nu}_t) + \mathbb{E}_{x_{t+1} \sim p(\cdot|x_t;\boldsymbol{\nu}_t)}\left[V^j(x_{t+1})\right] \tag{5.1.7}$$

which represents the value of each state-action pair $(x_t, \boldsymbol{\nu}_t)$ at their Nash equilibria. Hence in [12], the target is to approximate such global functions $\boldsymbol{V} := (V^1, \cdots, V^N)$ and $\boldsymbol{Q} := (Q^1, \cdots, Q^N)$, so that every agent at current state x_t is controlled by these global functions to select their actions $\boldsymbol{\nu}_t^*$ all together that maximizes the $\boldsymbol{Q}$ function. But as we discussed in section 2.3, agents j would not be able to make the decision based on the Q-function without the knowledge of $\boldsymbol{\nu}^{-j}$.

To address this issue, we decompose the Q-function Q^j into the *mean-field Q-function* $\hat{Q}^j$ as

$$Q^j(x_t, \nu_t^j, \boldsymbol{\nu}_t^{-j}) = \frac{1}{N-1} \sum_{k \neq j} \tilde{Q}^j(x_t, \nu_t^j, \mu_t) \approx \hat{Q}^j(x_t, \nu_t^j, \mu_t), \tag{5.1.8}$$

where $\tilde{Q}^j$ represents the the pairwise local interactions between agents j and agents k. A concrete proof of the approximation in equation (5.1.8) can be found in [34].

However, this approximation does not fully address the problem raised by the minimal information requirement because the agent j is still unable to make its decision on ν_t^j based on $\hat{Q}^j$ without knowing the mean-field action $\bar{\nu}_t$ ahead of time. Further, note that observing the mean-field action $\bar{\nu}_t$ at time stage t is not feasible in practice as the agents j itself has not selected its action ν_t^j at time stage t.

Therefore, inspired by [18], we proceed with a further approximation in the mean-field Q-function $\hat{Q}^j$ such that

$$\hat{Q}^k(x_t, \nu_t^j, \hat{\mu}_t^j) := \mathbb{E}_{\boldsymbol{\nu}_t^{-j} \sim \hat{\mu}_t^j}\left[r^j(x_t, \nu_t^j, \boldsymbol{\nu}_t^{-j}) + \hat{V}^k(x_{t+1})\right], \tag{5.1.9}$$

where

$$\hat{\mu}_t^j := f^k(x_t^j, \mu_{t-1}) \tag{5.1.10}$$

is the predicted mean-field action for time stage t estimated with the private state x_t and observed mean-field action μ_{t-1}, for some functions $\{f^k\}_{k\in\mathfrak{K}}$. Different from [18], here we denote $\hat{Q}^k$ as the mean-field Q-function for the agents j in the sub-population $\mathcal{K}_k$.

Intuitively, this modification should not significantly affect the DMFG performance. In the context of mean-field optimal liquidation problem presented in this paper, the individual agents i and j in the same sub-population $\mathcal{K}_k$ are not necessarily distinguishable, in the sense that each of them have precisely the same structure of the objective function (i.e., same risk aversion hyper-parameters ϕ_k, Φ_k in equation (3.2.1)), and the same market price impact λ_k. Therefore, assuming that the agents failing in the same sub-population should behave under the same *logic* is appropriate.

Remarkably, the agents in the same sub-population would not behave exactly the same given their distinct private information x^j and individual prediction towards the current mean-field action $\hat{\mu}^j_t$. In particular, their inventory process q^j would introduce the variation between the decisions cross the agents within the same sub-population. This reasoning should also apply the same when predicting the current mean-field action $\hat{\mu}^j_{t+1}$. In particular, we have

$$\hat{\mu}^j_{t+1} = f^k(x^j, \mu_{t-1}) \neq f^k(x^i, \mu_{t-1}) = \hat{\mu}^i_{t+1},$$

for $i \neq j$. Note that this distinction is almost surely guaranteed by the randomized initial inventories.

Remark 5.1.2. Notice that the equation (5.1.9) is still ill-defined as there are infinitely many possible linear combinations of $\{\nu^i_t\}_{i\neq j}$ to form a given predicted mean-field action $\hat{\nu}^j_t$. However, recalling the intuition from the decentralized learning introduced in section 2.3.2, this relationship is sufficient to provide an appropriate updating rule as we proposed as follows. Need to note that the updating rule for contiguous action space would be slightly different from the discrete case.

To formulate our modified DMFG algorithm based on [18] in the contiguous action-state space setting, we first construct pairs of neural networks $(\hat{Q}^k_\theta, \hat{V}^k_\theta, f^k_\theta)$ parameterized by $\theta \in \Theta$ for estimating (Q^k, V^k, f^k) defined previously. In order to approximately satisfy the equation (5.1.9), our objective is turned to minimize

$$\mathbb{E}\left[\left\|\hat{Q}^k_\theta(x^j_t, \nu^j_t, \hat{\mu}^j_t) - r^j(x^j_t, \boldsymbol{\nu}_t) - \hat{V}^k_\theta(x^j_{t+1}, \nu^j_{t+1}, \hat{\mu}^j_{t+1})\right\|^2\right], \tag{5.1.11}$$

over all $\nu^j \in \mathcal{A}^j$, for any $j \in \mathfrak{N}$. Intuitively, this equation measures the gap between the estimated mean-field Q-value of the current state and action pair $(x_t, \nu^j_t; \mu_t)$ and the estimated state value of the next time stage x_{t+1} if ν^j_t is executed, and should diminish when $\hat{Q}^k \to Q^k$ and $\hat{V}^k \to V^k$.

Except for the fact that the equation (5.1.9) is ill-defined, the equation (5.1.11) is also not tractable as the transition probability $p(x_{t+1}|x_t, \boldsymbol{\nu}_t)$ is unknown to the agents for all t. To this end, we rely on the simulation-based stochastic gradient decent method established on the insight from [30] (and also see [12, 18]) to search for $\theta \in \Theta$ with the loss function

$$\mathcal{L}(\theta) = \frac{1}{B}\sum_{b=1}^{B}\left\|\hat{Q}^k_\theta(x^j_t, \nu^j_b, \hat{\mu}^j_b) - r^j(x^j_b, \boldsymbol{\nu}_b) - \hat{V}^k_\theta(x^j_{b+1}, \nu^j_{b+1}, \hat{\mu}^j_{b+1})\right\|^2, \tag{5.1.12}$$

where $b = 1, 2, \cdots, B$ corresponds to the sampled batch from the observed state-action triples $\{(x_b, \nu^j_b, \mu_b, r^j_b, x_{b+1})\}$ for each update with some batch size $B \in \mathbb{N}$.

This updating rule, in turns, provides

$$\nu^{j,*}_t(x_t, \hat{\nu}^j_t) = \underset{\nu^j_t \in \mathcal{A}_t}{\operatorname{argmax}}\left\{\hat{Q}^k_\theta(x_t, \nu^j_t, \nu^j_t)\right\} \tag{5.1.13}$$

given current state x_t and predicted current mean-field action $\hat{\nu}^j_t$ for execution during the training process.

One common obstacle in reinforcement learning, as in our algorithm, is to find the ν^j_t that maximizes the $\hat{Q}^j$ function at each time stage t in equation (5.1.5), given our contiguous action space $\mathcal{A}^j$. In general, there are three methods to work around that.

- Discretize the Q-function. That is, randomly draw M admissible actions $\{\nu^j_{t,m} \in \mathcal{A}^j_t\}_{m\leq M}$ and compute $\{Q^k(x_t, \nu^j_{t,m}, \hat{\mu}^j_{t,m})\}_{m\leq M}$. Then, we select the largest Q-values and regard
$$\hat{\nu}^{j,*}_t = \operatorname*{argmax}_{m\leq M}\{Q^k(x_t, \nu^j_{t,m}, \hat{\mu}^j_{t,m})\}_{m\leq M}.$$
- Train another Q-function maximizer. That is, at each iteration of the Q-function, we spontaneously train another Q-function maximizer $g_{Q^k}(x_t, \hat{\mu}^j_t) := \operatorname{argmax}_{\nu^j_t} Q^k(x_t, \nu^j_t, \hat{\mu}^j_t)$, so that
$$\hat{\nu}^{j,*}_t = g_{Q^k}(x_t, \hat{\mu}^j_t).$$
- Assume the Q-function to be strictly convex with an explicit global maximizer. Then we automatically obtain $\hat{\mu}^{j,*}_t$ at each time stage t during the training process.

While the first and second tricks provide better estimation, they sacrifice the convergent efficiency significantly as both involve tremendous computation at each iteration step. Hence, similar to [12], we decompose the Q-function $\hat{Q}^k_\theta$ into a summation of *advanced function* $\hat{A}^k_\theta$ and value function such that

$$\hat{Q}^k_\theta(x^j_t, \nu^j_t, \hat{\mu}^j_t) = \hat{V}^k_\theta(x^j_t) + \hat{A}^k_\theta(x^j_t, \nu^j_t, \hat{\mu}^j_t) \tag{5.1.14}$$

where $\hat{A}^k_\theta$ measures the optimal gap between $\hat{Q}^k_\theta$ and $\hat{V}^k_\theta$. In practice, the $\hat{A}^k$ is usually assumed to preserve the linear quadratic form as in

$$\hat{A}^k_\theta(x^j_t, \nu^j_t, \hat{\mu}^j_t) := -\left\|\nu^k_\theta(x^j_t, \hat{\mu}^j_t) - \nu^j_t\right\|^2_{\Sigma_\theta(x^j_t, \hat{\mu}^j_t)}, \tag{5.1.15}$$

for some positive definite matrix function $\Sigma_\theta(\cdot, \cdot)$. It follows that

$$\hat{\nu}^{j,*}_t = \nu^k_\theta(x^j_t, \hat{\mu}^j_t). \tag{5.1.16}$$

Remark 5.1.3. The third trick is usually designed for vector-form actions in the reinforcement learning literature (for example see [12, section 4]). But notice that the action $\nu^j_t \in \mathbb{R}$ in this paper. Therefore, the equation (5.1.15) boils down to

$$\hat{A}^k_\theta(x^j_t, \nu^j_t, \hat{\mu}^j_t) = -\Sigma(x^j_t, \hat{\mu}^j_t) \cdot (x^j_t, \hat{\mu}^j_t) - \nu^j_t)^2,$$

for some parameter $\Sigma_\theta(\cdot, \cdot) \in \mathbb{R} > 0$. Although it may seem over-simplified, the performance, as seen in the section 5.2, is satisfactory. Besides, we also tested several other convex functions (e.g., even-ordered polynomials up to 10) and the performance turns out to be no significant improvement.

The algorithm 1 below provides a pseudo-code of the actor-critic procedure for the decentralized mean-field game Q-learning algorithm. We apply the decentralized mean-field game Q-learning updates in [18] to the model-free mean-field framework with some modification in the update equations, where the neural networks for approximating the value functions V^j and mean-field Q-function Q^j are set to be identical among each sub-population $\mathfrak{K}_k$. Further, the agents j only observe their local state x^j_t and have only access to the previous global mean-field action $\bar{\nu}_t$.

Algorithm 1 Actor-Critic for Decentralized mean-field Game

```
Input: # Episodes $E > 0$, Minibatch Size $B > 0$, Terminal Time $T$, Time Step $dt$.
Input: Exploration Nose $\{\sigma_b : \sigma_b > 0\}_{b=1}^{B}$
Initialize Replay Buffer $\mathcal{D}$, Parameters $\{(\theta_A^k, \theta_V^k, \theta_f^k)\}_{k<K}$ for the critic $V_{\theta^k}$, the actor $Q_{\theta^k}$ and mean-field estimate $f_{\theta^k}$ parameterized by $(\theta_A^k, \theta_V^k, \theta_f^k)$.
Initialize the first observed mean-field action $\mu_0^j$ for each agents $j$ to a uniform distribution.
while Episode $< E$ do
    Obtain the current state $x_0^j$ for each agents $j$.
    while $t < T$ do                    ▷ $t$ can be extracted from $x_t$, and start from $t_1 > t_0 = 0$
        For each sub-population $k$, obtain the estimated mean-field action $\hat{\boldsymbol{\mu}}_t^k = (\hat{\mu}_t)_{j\in\mathcal{K}_k}$ based on observed mean mield action $\mu_{t-1}$ for all agents $j$ in the sub-population.
        For each sub-population $k$, obtain action $(\nu_t^j)_{j\in\mathcal{K}_k}$ from the $\text{argmax}_{\nu_t^j}\{Q_{\theta^k}(x_t^j, \nu_t^j, \hat{\mu}_t^j)\}$ at the current state $x_t^j$ for all agents $j$ in the sub-population.
        Execute the joint action $\boldsymbol{\nu} = (\nu_t^1, \cdots, \nu_t^N)$. Observe the rewards $\boldsymbol{r}_t = (r_t^1, \cdots, r_t^N)$ and the next state $\boldsymbol{x}_t = (x_t^1, \cdots, x_t^N)$ and the current observed mean-field action $\mu_t$.
        Store $y_t = (x_t, \boldsymbol{\nu}_t, \boldsymbol{r}_t, x_{t+1})$ in $\mathcal{D}$.
        Sample $\mathcal{D}' = \{y_i\}_{i=1}^{B}$ from $\mathcal{D}$.
        For each sub-population $k$, update the parameter $\theta_f^k$ of the mean-field network using a mean square error between the observed and estimated mean actions $\mu_t$ and $\mu_t^j$ for all agents $j$ in the sub-population.
        For each sub-population $k$, update the parameter $\theta_V^k$ of the critic network using stochastic gradient descent over the loss $\frac{1}{B+1}\sum_{y\in\mathcal{D}'\cup\{y_t\}}\mathcal{L}_V(y, \theta_V^k)$ for all agents $j$ in the sub-population.
        For each sub-population $k$, update the parameter $\theta_Q^k$ of the actor network using stochastic gradient descent over the loss $\frac{1}{B+1}\sum_{y\in\mathcal{D}'\cup\{y_t\}}\mathcal{L}_Q(y, \theta_V^k, \theta_Q^k)$ for all agents $j$ in the sub-population.
        Set the next state as the current state $x_t^j \leftarrow x_{t+1}^j$ and the observed mean-field action $\mu_{t-1} \leftarrow \mu_t$
    end while
end while
Return $\theta^k = (\theta_A^k, \theta_V^k, \theta_f^k)$ for $k < K$.
```

5.2 Experiments

In this section, we study the performance of our modified DMFG algorithm in the three optimal liquidation problems described in chapter 4. The parameters we used for the environment can be found in table 5.1. Further, all the agents are trained given the same algorithm as described in algorithm 1.

Table 5.1: Asset price process, and environment parameters.

dt	T	ρ	S_0	κ	η	σ
0.1	10	1	10	0.5	10	1

Each experiment in this subsection was run consecutively in training and evaluation phases. The training process follows the algorithm 1, but with a decaying probability $\epsilon_i \in [0.1, 0.99]$ at the i-th iteration to explore some random action $\boldsymbol{\nu}_t^k$ for each sub-population $\mathcal{K}_k$ at each time stage $t \in [0, T]$. After the training process, we would evaluate and record the performance of these agents for 200 games. In the following subsections, we present and analysis the performance of our agents for each of our experiment, compared with the theoretical optimal strategy or Nash equilibrium derived in chapter 4.

The training processes were run on a 1 GPU physical machine with 8 GB GPU memory and took roughly 0.2 days, 2 days and 3 days for each experiment, respectively. To remain a relatively acceptable training speed, the network structure was designed as a fully connected neural network with 5 hidden layers each containing roughly 20-50 nodes. This network structure, however, would potentially introduce some issues, which we would analyse in section 5.2.2 and section 5.2.3 in details.

For the following examples, we would abuse the notation of the objective function to denote the accumulated rewards received by time stage t as $\mathcal{R}_t$, and thereby the reward process as $\mathcal{R} = (\mathcal{R}_t)_{t \in [0,T]}$.

5.2.1 Optimal Liquidation for Single Agent

In this experiment, we studied and compared the performance of the DMFG algorithm with the theoretical optimal trading strategy we derived in section 4.2.

For this example, we assume a single major agent ($N_1 = 1$) with relatively large price impact $\lambda_1 = 0.15$ and relatively small transaction cost $c_1 = 0.15$. Further, although we assume the agent to be patient in the sense that ϕ is set to be 0.05, it indeed has a demand to liquidate its inventory within the time interval $[0, T]$ as we described in chapter 1, otherwise it would be forced to liquidate its left inventory q_T by time stage t at the cost $\Phi = 2$. The summary of agent-related parameters can be found in table 5.2.

Table 5.2: Agent-related hyper-parameters for optimal liquidation problem with major (single) agent.

k	λ	c	Φ	ϕ	N_k	$\mathbb{E}[q_0^j]$	$\mathbb{V}[q_0^j]$
Major Agent	0.15	0.15	2	0.05	1	20	3

After training for 1000 games, the DMFG agent was evaluated in the execution phase for 200 games, and achieved 94.25% average final reward compared with the optimal trading strategy derived in section 4.2. Here we presents the performance in figure 5.1.

The DMFG agent not only learned to liquidate its inventory by the terminal stage T but also learned to liquidate it faster during the beginning, even though there is still an obvious gap between the optimal and the DMFG trading strategies shown in figure 5.1a. However, given our relatively simple, fully-connected network structure, we did not expect our DMFG agent to be as sensitive as the theoretical result. Given that the Q-learning algorithm is model-free, i.e., the agent does not know the underlying dynamics, such a performance is satisfactory.

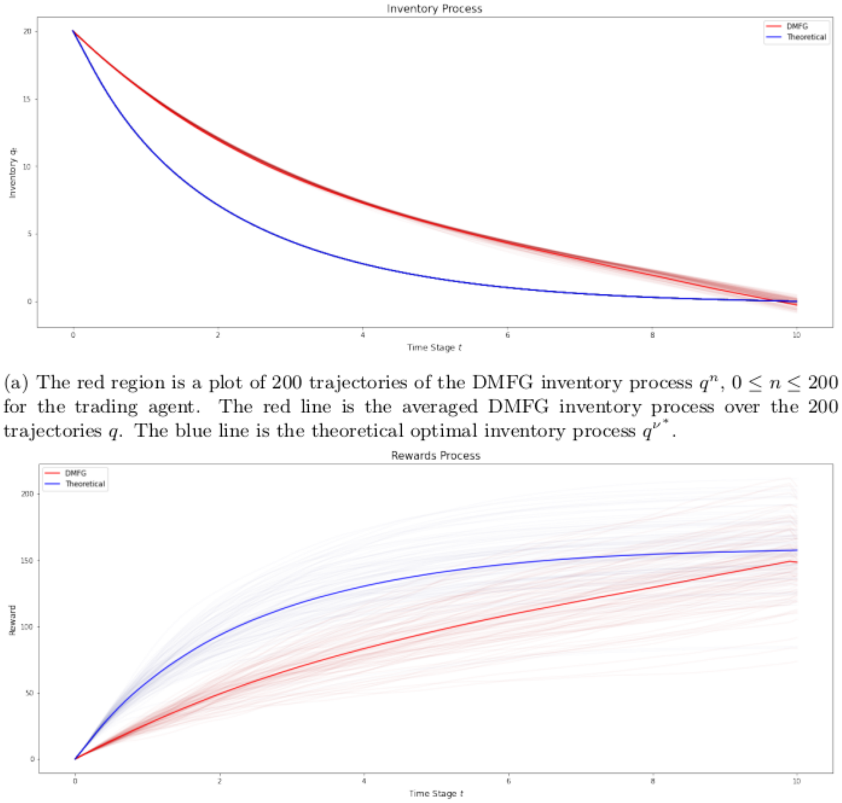
Based on this experiment, we conclude that the DMFG algorithm with our modifications described in section 5.1 (e.g., the assumption about the advantage function $\hat{A}$) is applicable to the optimal liquidation problem. This result encourages us to bound forward to more general mean-field examples in the following subsections.

5.2.2 Trading with Crowds

In this experiment, we studied and compared the performance of the DMFG algorithm with the theoretical Nash equilibrium derived in section 4.3.

For this example, we assumed the trading crowds with $N = 5$ trading agents with relatively *small* price impact $\lambda = 0.15$ and relatively *large* transaction cost $c = 0.3$. Even though λ_2 has the same value as λ_1 in table 5.2, note that it only defines the price impact of the sub-population as a whole, while each individual trading agent would only have $\lambda_2 / N_2 = 0.03$ as its price impact. Further, we also assumed these minor agents to be patient in the sense that ϕ is set to be 0.05. However, we assumed that they did not necessarily have the demand to liquidate all of their inventories by the terminal time stage T. Instead, we set $\Phi = 0.5$ which is slightly greater the transaction cost c. Intuitively, the terminal penalty can be interpreted as the expected value of

Figure 5.1: Performance comparison between DMFG and theoretical result in single agent setting.



(a) The red region is a plot of 200 trajectories of the DMFG inventory process q^n, $0 \leq n \leq 200$ for the trading agent. The red line is the averaged DMFG inventory process over the 200 trajectories q. The blue line is the theoretical optimal inventory process q^{ν^*}.

(b) The red region is a plot of 200 trajectories of the DMFG reward process $\mathcal{R}^n$, $0 \leq n \leq 200$ for the trading agent. The red line is the averaged DMFG reward process over the 200 trajectories $\mathcal{R}$. The blue line is the theoretical reward process $\mathcal{R}^{\nu^*}$ in Nash equilibrium.

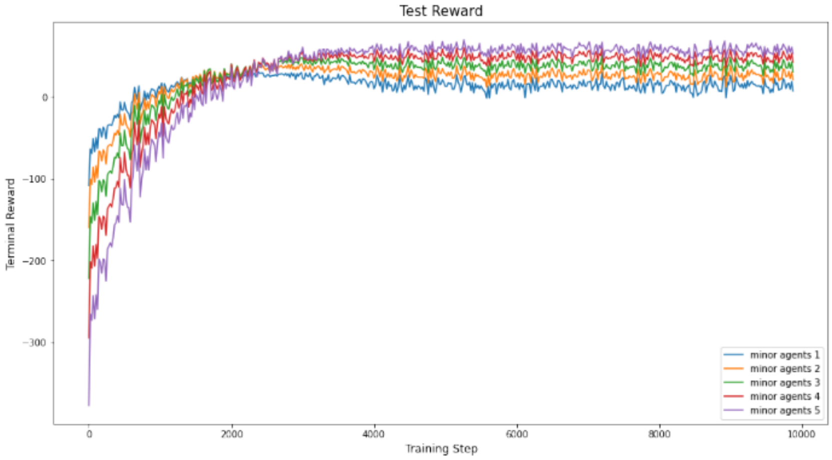
the terminal inventory if the agent j need to solve it immediately, with the value of holding the stock to the future into consideration. The summary of agent-related parameters can be found in table 5.3.

Table 5.3: Agent-related hyper-parameters for optimal liquidation problem with trading crowds (minor agents).

k	λ	c	Φ	ϕ	N_k	$\mathbb{E}[q_0^j]$	$\mathbb{V}[q_0^j]$
Minor Agent	0.15	0.3	0.5	0.05	5	5	3

This time, we trained the DMFG agents for 10000 games and then tested their performance in the evaluation phase for 200 games. The tested performance at each training step is plotted in figure 5.2. Here we fixed the initial inventory $\boldsymbol{q}_0 = (3, 4, 5, 6, 7)$ for better visualization despite we randomized it during the training process.

Figure 5.2: The averaged terminal reward on evaluation phase at each training step in trading crowds setting.



Different from the performance in section 5.2.1, however, the trading agents in this scenario did not achieve the theoretical result as we have observed previously in figure 5.3. In particular, the final reward ratio is presented in table 5.4.

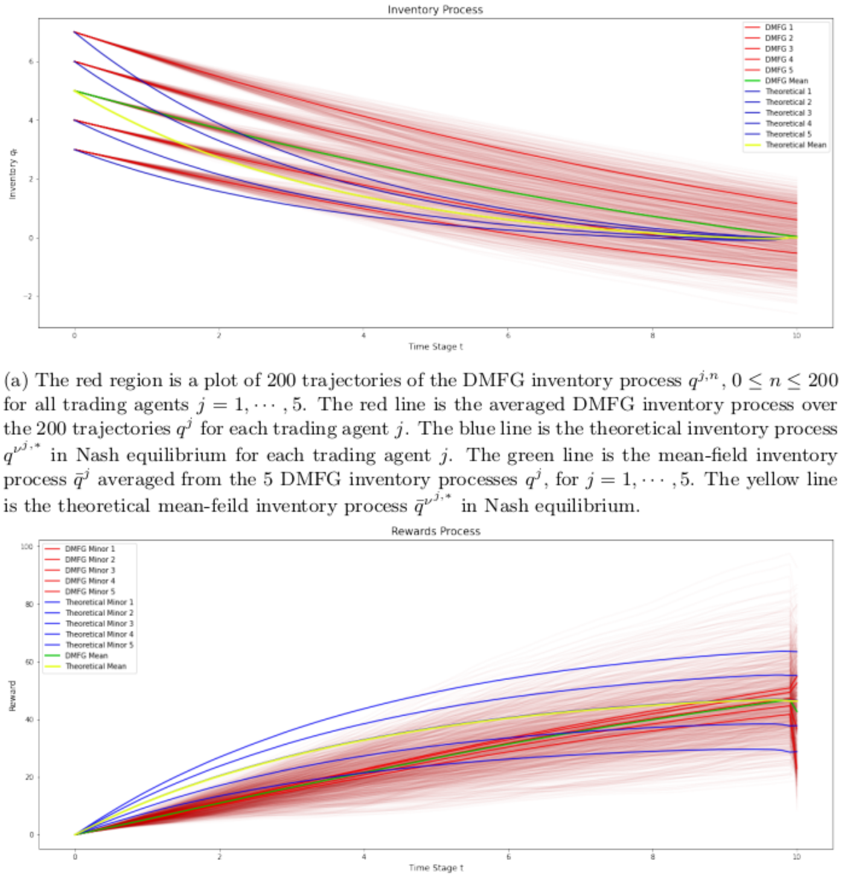
Table 5.4: Minor (Crowds) Agents Performance Ratio

Initial Inventory	3	4	5	6	7
Minor Agent	79.63%	97.223%	99.92%	95.33%	86.50%

This performance is not surprising if we investigate the DMFG inventory process presented in figure 5.3a. Contrasting from our analysis in section 5.1, the minor agents in the crowds did not select significantly different trading strategies over the 200 evaluation trajectories given distinct private information $x_t^j = (t, S_t, q_t^{\nu^j}, \hat{\mu}_t^j)$ for each agent j. This results in the quasi-parallel [1] inventory processes as the red lines indicated in figure 5.3a. This quasi-parallel feature would not disappear when we enlarge the variance of the initial inventories. Specifically, the terminal

[1] Note that the agent's action ν_t^j given its own private information does vary over agent $j = 1, \cdots, 5$. In particular, it tends to sell more when it currently has larger inventory q_t^j. However, this difference in the infinitesimal time period does not produce a fundamental difference on the inventory process q^j.

Figure 5.3: Performance comparison between DMFG and theoretical result in trading crowds setting.



(a) The red region is a plot of 200 trajectories of the DMFG inventory process $q^{j,n}$, $0 \leq n \leq 200$ for all trading agents $j = 1, \cdots, 5$. The red line is the averaged DMFG inventory process over the 200 trajectories q^j for each trading agent j. The blue line is the theoretical inventory process $q^{\nu^{j,*}}$ in Nash equilibrium for each trading agent j. The green line is the mean-field inventory process $\bar{q}^j$ averaged from the 5 DMFG inventory processes q^j, for $j = 1, \cdots, 5$. The yellow line is the theoretical mean-feild inventory process $\bar{q}^{\nu^{j,*}}$ in Nash equilibrium.

(b) The red region is a plot of 200 trajectories of the DMFG reward process $\mathcal{R}^{j,n}$, $0 \leq n \leq 200$ for all trading agents $j = 1, \cdots, 5$. The red line is the averaged DMFG reward process over the 200 trajectories $\mathcal{R}^j$ for each trading agent j. The blue line is the theoretical reward process $\mathcal{R}^{\nu^{j,*}}$ in Nash equilibrium for each trading agent j. The green line is the mean-field reward process $\bar{\mathcal{R}}^j$ averaged from the 5 DMFG reward processes $\mathcal{R}^j$, for $j = 1, \cdots, 5$. The yellow line is the theoretical mean-feild reward process $\bar{\mathcal{R}}^{\nu^{j,*}}$ in Nash equilibrium.

inventories would have a smaller variance compared with that of the initial inventories, but it would definitely not reach zero for each agent j as we expected in the theoretical Nash equilibrium result indicated from the blue lines in figure 5.3a.

While the individual trading agents did not produce the ideal result, the *mean-field representative* we specified in remark 4.3.4 still performed nicely. In particular, the DMFG mean-field process $\bar{q}$ started with $\bar{q}_0 = 5$ would overlap with that of the individual trading agent with initial inventory $q_0^3 = 5$, as shown by the green line and the mid blue line in figure 5.3a. Further, notice that both of them would liquidate their inventories by the terminal time stage T. Also, such a mean-field inventory process achieved the highest performance ratio at 99.92% among other individual inventory processes. In fact, one way to interpret the plot is to regard the other individual inventory processes as quasi-parallel to the mean-field one.

Remark 5.2.1. Note that the green DMFG mean-field inventory process is still significantly different from the yellow theoretical mean-field inventory process indicated in figure 5.3a. As we discussed before, we did not expect our DMFG agents to be as sensitive as the theoretical result given our relatively simple fully connected network structure. Intuitively, the agents were encouraged to maximize the accumulated rewards $\mathcal{R}_T^j$ it received by the terminal stage T. Therefore, once it achieves a relatively maximized terminal rewards $\mathcal{R}_T^j$ on average within a certain threshold, the learning process would slow down.

Nevertheless, recall that we have shown in section 4.3 (also rigorously in [9]) that the Nash equilibrium in the trading with crowds scenario is *unique*. Further, it has also been jointly proved by [30, 18] that the DMFG algorithm should converge, though the algorithm has been slightly modified in this paper. That is to say, the DMFG learning algorithm should provide the theoretical performance as we have seen in section 5.2.3 as long as it converges to an equilibrium. Therefore, it is natural to ask why the individual trading agents cannot produce a similar inventory process to the theoretical result as we derived in section 4.3.

There are several hypotheses to explain this empirical result. The first straightforward explanation is that the 10000-game training process does not appear to be large enough for convergence. But note that 10000 games mean 10000 × 100 data points as each game has 100-time stages, as well as that 1000 games are large enough for the single agent case to converge. This does not provide a convincing explanation for us.

Besides, one may argue that the modified DMFG learning algorithm cannot inherit the properties from [30, 18] and hence does not guarantee a provably convergent result. It does provide a plausible explanation considering the empirical example we suggested in section 2.3. That is, this empirical result supports that the existence and uniqueness of the Nash equilibrium do not necessarily guarantee convergence, as suggested by Nagel's empirical experiments in 1995 [13]. However, as the proof of the convergence of this modified DMFG algorithm is out of the scope of this paper, we would not provide further proof for our DMFG algorithm, given the limited time available for the thesis project. Instead, we would leave this as an open problem for future studies.

Apart from the explanations provided above, one another hypothesis appears more promising to us. Recall the theoretical Nash equilibrium derived in theorem 4.3.5. Notice that the required input is $(t, F_t, q^{\nu^j}, (\bar{Y}_t^{\bar{\nu}^*})_{t\in[0,T]})$, whereas our input is $(t, S_t, q^{\nu^j}, \hat{\mu}_t)$. In particular, even if our model-free agents find a way to decompose S_t into F_t and M_t from its experiences of playing the games for 10000 games and find a perfect pattern to predict μ_t given μ_{t-1} so that they can produce $\bar{Y}_t$ up to filtration $\mathcal{F}_t$, notice that they have a built-in inability [2] to compute $\bar{Y}_T$ given the information

[2] Note that in theorem 4.3.5, however, $\bar{Y}_T$ is able to compute beforehand due to the introspective thinking process (or $\mathbb{E}_t[\bar{Y}_T]$ in [9] given a general signal process A). In the terminology of RL literature, we say that each agent j is controlled by a centralized system. Since the centralized system has access to complete information, $\bar{Y}_T$ (or $\mathbb{E}_t[\bar{Y}_T]$) can be computed beforehand and provided for each agent j at time stage t. This also explains why the Deep Q-learning method for the trading crowds scenario by Casgrain, Ning, and Jaimungal in [12] works.

by time stage t. Therefore, in general, they will never be able to converge to the theoretical Nash equilibrium as we derived in theorem 4.3.5.

On the other hand, this hypothesis also explains why the mean-field representative (or, the individual trading agent with initial inventory $q_0^3 = 5$) produces a 99.92% terminal reward divided by the mean-field theoretical terminal reward. Recall the theoretical mean-field Nash equilibrium derived in corollary 4.3.3. The required input is $(t, F_t, \bar{q}_t, \bar{Y}_t)$, which can be represented by our input $(t, S_t, q_t^{\nu^3}, \hat{\mu}_t)$. That is, instead of learning the individual Nash equilibrium, the neural network for the sub-population learned the mean-field "optimal" strategy given the private information, with a slight approximation to the individual "optimal" strategies up to the filtration $\mathcal{F}_t$.

Note that the logic behind the section 5.1 is still holding. To our knowledge, there are two ways to fix this hypothetical problem.

Firstly, one may consider constructing two neural networks for each sub-population instead of only one. While both of the neural networks would be designed and trained as what we have designed in algorithm 1, the second neural network would take the estimated mean-field process $\hat{Y} = (\hat{Y}_t)_{t \in [0,T]}$ generated by the first one as its input at the beginning stage $t = 0$ or during each time stage $t \in [0, T]$ for each game. Intuitively, the first neural network serves the function of *group cognition* as defined in [40]. In contrast, the second one provides the desired optimal trading strategies given their group cognition.

The new network structure retains linearity to the sub-population amount regarding the computation cost, except that it may arguably violate our assumption that each individual has minimal information, as the information within the sub-population level has been revealed to the second neural network in some sense. Specifically, the neural network for group cognition aggregates the private information of each agent j during the training. It then reveals it by providing the second neural network the estimated sub-population price impact process $\hat{Y}$ up to filtration $\mathcal{F}_T$.

As we mentioned at the beginning of section section 5.2, the second method is to investigate the fully connected feedforward network structure. Since we have noticed that the main issue arises from the incapability to compute the exact $\bar{Y} = (\bar{Y}_t)_{t \in [0,T]}$ given $\mathcal{F}_t$, a recurrent neural network, such as the LSTM [41] and/or the multi-head attention [42] should be considered, in the sense that to help the trading agents either realize that Y_T is deterministic (which is only valid given our Vasicek process), or estimate the future mean actions $(\mu_t, \mu_{t+1}, \cdots, \mu_T)$ given the history $(\mu_1, \mu_2, \cdots, \mu_{t-1})$.

Remark 5.2.2. Such a modification, as we will argue in section 5.2.3, may turn out to be insufficient in the general case when there are $K > 1$ different sub-populations.

Unfortunately, both methods are out of the scope of this paper, given the limited time and the available physical machine. Therefore, we did not carry out the implementation to test our hypothesis and left it as one of the open problems for our future studies.

Based on this experiment, we conclude that the modified DMFG algorithm may not be able to perform perfectly at the individual level for each sub-population but is satisfactory at the mean-field level. Further, we offer some analysis and several possible explanations for this outcome. Although the result is not ideal, these attempted explanations still provide some support for our following experiment and the subsequent research.

5.2.3 Major-Minor

In this experiment, we studied the performance of the DMFG algorithm for the general case in our model set-up. In particular, we set the $K = 2$, with a single $(N_1 = 1)$ trading (major) agent in the first sub-population $\mathcal{K}_1$, and a trading crowds of $(N_2 = 5)$ minor agents in the second sub-population $\mathcal{K}_2$. This experiment is motivated by the model presented in [10]. In this scenario, we cannot provide a theoretical Nash equilibrium as shown in section 4.4. However, we may still utilize some insights from previous theoretical results to evaluate our modified DMFG agents' performance.

For this example, we assume the single major agent with relatively large price impact $\lambda_1 = 0.15$ and relatively small transaction cost $c_1 = 0.15$, whereas the trading crowds with $N = 5$ minor agents with relatively small price impact $\lambda_2 = 0.15$ and relatively large transaction cost $c_2 = 0.3$. Again, note that the major agent has larger market price impact because λ only defines the price impact of the sub-population as a whole. The market price impact of each *individual* trader in the trading crowds is only $\lambda_2/N_2 = 0.03$.

Further, both sub-populations were assumed to be patient in the sense that $\phi_1 = \phi_2 = 0.05$. However, the major agent has a demand to liquidate its inventory within the time interval $[0, T]$ as described in chapter 1. Otherwise, it would be forced to liquidate its left inventory q_T by time stage t at the cost $\Phi_1 = 2$, while the minor agents do not, given their $\Phi_2 = 0.5$ which is only slightly greater than their transaction cost c_2. In other words, only the major agent was required to liquidate its inventory by terminal stage T, which is more realistic as different trading agents may have their fundamental analysis to decide whether to liquidate the shareholding for a specific underlying stock. Agents in different sub-populations do not necessarily need to make the same decision to liquidate their inventory within the next hour (e.g., T = 1 hour).

The summary of agent-related parameters can be found in table 5.5.

Table 5.5: Agent-related hyper-parameters for optimal liquidation problem with major-minor agents

k	λ	c	Φ	ϕ	N_k	$\mathbb{E}[q_0^j]$	$\mathbb{V}[q_0^j]$
Major Agent	0.15	0.15	2	0.05	1	20	3
Minor Agent	0.15	0.3	0.5	0.05	5	5	3

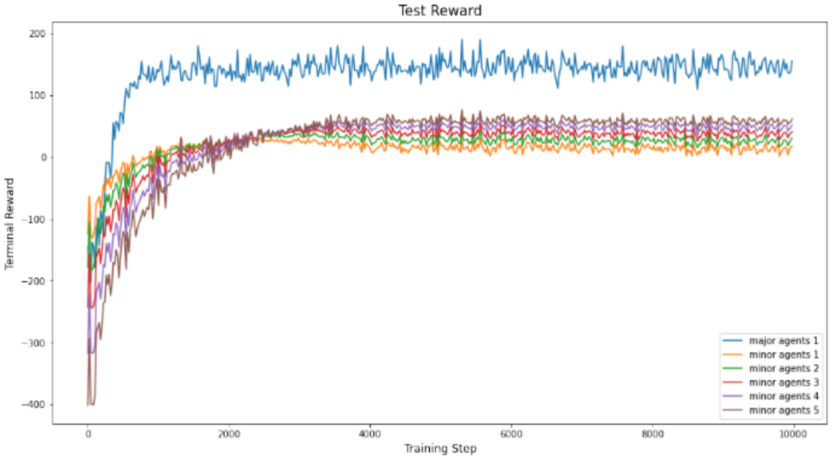
We trained the DMFG agents in this example for 10000 games and then tested their performance in the evaluation phase for 200 games. Again, we fixed the initial inventory for $\boldsymbol{q}_0^{k_1} = (20)$ and $\boldsymbol{q}_0^{k_2} = (3, 4, 5, 6, 7)$ for better visualization despite we randomized it during the training process. But in this scenario, we cannot provide a theoretical result for comparison as we did in the previous two examples. Similar to section 5.2.2, we regard the DMFG learning dynamics converges by around 4000 given the evaluation plot in figure 5.4.

Remark 5.2.3. Note that the agent $j = 5$ in figure 5.4 starting with a lower terminal reward but ending with a higher one among the trading crowds during the training process, mainly because it has the largest initial inventory $q_0^5 = 7$ and learns to liquidate its inventory by the terminal stage T.

Firstly, most characteristics of the major and minor agents' performances as in section 5.2.1 and section 5.2.2, especially the quasi-parallel feature, remain as we expected. Further, notice that the major agent converges faster than the minor agents, which contradicts our intuition as the minor agents have five times more training data set than the major agent. That is, each of the minor agents' private information and learning experience contributes to the minor-agent sub-population neural network. In contrast, the major-agent neural network only receives one at a time. This outcome, in turn, supports our hypothesis in the previous section if we admit that our modified DMFG algorithm can only learn the mean-field representative's value function and Q-function. Recall the remark 4.3.4, the giant representative not only needs to learn how to react according to its opponent's actions, but also how to coordinate with its component. This extra feature adds difficulty for the representative to converge.

While, the hypothesis in section 5.2.2 suggests that the outcome of the DMFG algorithm may

Figure 5.4: The averaged terminal reward on evaluation phase at each training step in major-minor setting.



not converge to the Nash equilibrium, the DMFG learning dynamics still provide some interesting results.

First, both major agent and minor agents are competing to each others. Due to the existence of the minor agents who are also liquidating and create a downward pressure toward the market, i.e., a larger Y_t, hinted by theorem 4.2.3 and theorem 4.3.5, we know that both the major and minor agents have a trend to sell more if Y_t increase. Intuitively, if there exists a large Y_t given $\mathcal{F}_t$, i.e., the trading agents know there will be a downward pressure over the price, their best responses would be sell more if they originally decide to sell.

Secondly, note that the existence of *the opponent sub-populations* results in the intention for both the major and minor agents to liquidate more aggressively compared with their performances in section 5.2.1 and section 5.2.2. In particular, we observed that the major and minor agents tend to liquidate earlier than the terminal stage T as they did separately in the previous example. Besides, as indicated in figure 5.5 [3] , both of them significantly tend to sell slightly more than their initial inventories. Specifically, both the major and the mean-field representative of the minor agents tend to terminate their investment account by T with roughly -1 share held on average.

Further, notice that there exists competition during the training process as well. Based on theorem 4.2.3 and theorem 4.3.5 (also hinted in [10]), we may imagine that the Nash equilibrium should be that both major and minor agents liquidate their inventories all together by terminal time stage T. However, in figure 5.5, we observe that the major agent converge first by liquidating all its inventory by time stage T (figure 5.5a), and then the liquidating actions of the minor agents push the major agents to liquidate and exit the market early (figure 5.5b).

This outcome is not surprising. The extra liquidation from the minor agents amplifies the market price impact Y_t for each time stage t. Suppose the minor agents' strategies are observable to the major agent, i.e., the major agent knows there would be an extra downward pressure except for what it itself causes. It is reasonable for the major agent to liquidate faster than the minor agents.

However, the fact that the major agent converges early revealed the major agent's strategy and market impact to the minor agents. Therefore, we observed from figure 5.5c that the minor

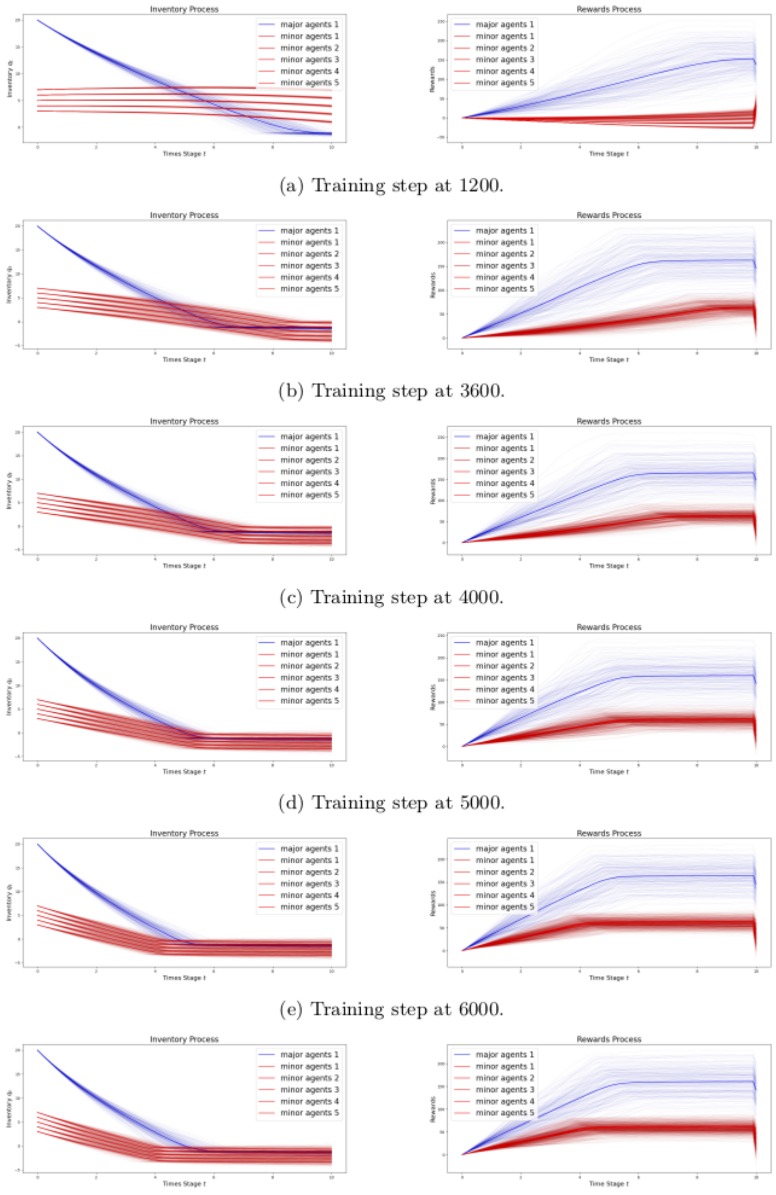
[3] For example, one may notice that the inventory processes are significantly less than 0 cross all training steps, and there is always a relatively large drop for each agent at the terminal stage T in the reward process, indicating that the terminal penalty is not negligible.

agents start to liquidate faster as well, given the same logic presented above, which in turn pushes the major agent to increase its liquidating speed and exit the market even earlier, as we can see comparing between figure 5.5c, figure 5.5d and figure 5.5e. This competition lasts until the major agent cannot offer a faster liquidation, given its large amount of initial inventory and its opponents' limited market impact [4]. On the other hand, recall that the major agent's strategy converges faster. The minor agents can then form their optimal strategies based on the major agent's trading-speed limit, with the trade-off between liquidating speed and transient price impact also considered. Finally, as indicated in figure 5.5e and figure 5.5f, This competition terminates, and the DMFG algorithm reaches an equilibrium. Presumably speaking, it is not the Nash equilibrium. And unfortunately, there is no winner in that equilibrium.

The outcome, from our perspective, is hard to understand. Note that by design, as claimed in [22, 30, 18], the agents learned with minimal information, and are oblivious the existence of its opponents. In other words, it shows that the local-Q functions revealed the existence of opponents. On the other hand, the result also provides some insight for solving the section 4.4, combined with the theorem 4.2.3 and theorem 4.3.5.

[4] After all, the market price impact caused by the minor agents is limited given their relatively small initial inventory. But bear in mind that the total inventory of the trading crowds is $25 > 20$.

Figure 5.5: Performance comparison between DMFG and theoretical result in major-minor setting.



(a) Training step at 1200.

(b) Training step at 3600.

(c) Training step at 4000.

(d) Training step at 5000.

(e) Training step at 6000.

(f) Training step at 10000.

Conclusion and Subsequent Problems

In this thesis project, we introduced optimal liquidation in high-frequency trading and multi-agent reinforcement learning in stochastic games. Specifically, we have implemented and presented a modified decentralized mean-field games algorithm for acceptable computational costs. Further, we compared the theoretical results from the high-frequency trading literature with our modified DMFG algorithm and offered some intuitive explanations and hypotheses. We believe our thesis project has opened up fruitful directions for future research on developing the individual-population paired DMFG framework and the recurrent neural network based DMFG framework. Besides, we believe this interdisciplinary project between high-frequency trading and multi-agent reinforcement learning provides a concrete methodology to examine our learning techniques in finance and improve their trackability for practice purposes.

Appendix A

Technical Proofs

A.1 Single Agent

Lemma A.1.1 (Lemma 5.1 [19]). *For $\nu \in \mathcal{A}$, we have*

$$\langle \mathcal{R}_0(\nu), \alpha \rangle = \mathbb{E}\left[\int_0^T \alpha_t \left(F_t - Y_t^\nu - \int_t^T e^{-\rho(s-t)} \lambda \nu_s ds - 2c\nu_t + 2\phi \int_y^T q_s^\nu ds + 2\Phi q_T^\nu - F_T\right) dt\right] \tag{A.1.1}$$

Proof. Here, we provide a sketch of proof from [19, Lemma 5.1] as a practice. This proof will be useful for the other two cases in chapter 4.

Let $\epsilon > 0$ and $\nu, \alpha \in \mathcal{A}$. Note that $X_t^{\nu+\epsilon\alpha} = X_t^\nu - \epsilon \int_0^t \alpha_s ds$ and $Y_t^{\nu+\epsilon\alpha} = Y_t^\nu + \epsilon \int_0^t \alpha_s ds$. Next, compute that

$$\begin{aligned}
&\mathcal{R}_0(\nu + \epsilon\alpha) - \mathcal{R}_0(\nu) \\
&= \epsilon\mathbb{E}\left[\int_0^T \alpha_t \left(F_t - Y_t^\nu - \int_t^T e^{-\rho(s-t)} \lambda \nu_s ds - 2c\nu_t + 2\phi \int_y^T q_s^\nu ds + 2\Phi q_T^\nu - F_T\right) dt\right] \\
&\quad - \epsilon^2\mathbb{E}\left[\int_0^T \left(\int_0^t e^{-\rho(s-t)} \alpha_s ds\right) \alpha_t dt + c\int_0^T \alpha_t^2 dt + \phi \int_0^T \left(\int_0^t \alpha_s ds\right)^2 dt - \Phi\left(\int_0^T \alpha_t dt\right)^2\right].
\end{aligned} \tag{A.1.2}$$

Then, recall the definition of Gateaux derivative, we divide the above equation by ϵ and take the limit to 0, hence we would obtain the desired result after applying Fubini's theorem twice. □

A.1.1 Proof of Lemma 4.2.2

Proof. Here, we provide a sketched proof of lemma 4.3.1 following the proof in [19, Lemma 5.2.] as a practice learning to solve the single agent problem using Gateaux derivative.

By lemma A.1.1, we first set $\langle \mathcal{R}_0(\nu), \alpha \rangle = 0$, i.e.,

$$\mathbb{E}\left[\int_0^T \alpha_t \left(F_t - Y_t^\nu - \int_t^T e^{-\rho(s-t)} \lambda \nu_s ds - 2c\nu_t + 2\phi \int_y^T q_s^\nu ds + 2\Phi q_T^\nu - F_T\right) dt\right] = 0, \tag{A.1.3}$$

for all $\alpha \in \mathcal{A}$.

Necessity: Assume that ν^* maximizes $\mathcal{R}_0$, i.e., equation (A.1.3) is satisfied. Then, by tower property,

$$\begin{aligned}\mathbb{E}\Bigg[\int_0^T \alpha_t \Bigg(F_t - Y_t^{\nu^*} - \mathbb{E}_t\left[\int_t^T e^{-\rho(s-t)}\lambda\nu_s^* ds\right] - 2c\nu_t \\ +\mathbb{E}_t\left[2\phi\int_t^T q_s^{\nu^*}ds + 2\Phi q_T^{\nu^*} - F_T\right]\Bigg)dt\Bigg] = 0,\end{aligned} \tag{A.1.4}$$

for all $\alpha \in \mathcal{A}$. Hence, it implies that

$$F_t - Y_t^{\nu^*} - \mathbb{E}_t\left[\int_t^T e^{-\rho(s-t)}\lambda\nu_s^* ds\right] - 2c\nu_t^* + \mathbb{E}_t\left[2\phi\int_t^T q_s^{\nu^*}ds + 2\Phi q_T^{\nu^*} - F_T\right] = 0 \tag{A.1.5}$$

with $d\mathbb{P}\bigotimes ds$-a.e. on$\Sigma \times [0,T]$.

Next, define two auxiliary square integrable martingales

$$\tilde{M}_t := \mathbb{E}_t\left[2\phi\int_t^T q_s^{\nu^*}ds + 2\Phi q_T^{\nu^*} - F_T\right], \quad \tilde{N}_t := \mathbb{E}_t\left[\int_0^T e^{-\rho(s-t)}\lambda\nu_s^* ds\right] \tag{A.1.6}$$

and square integrable process

$$Z_t^{\nu^*} := \int_0^t e^{-\rho(s-t)}\lambda\nu_s^* ds - e^{\rho t}\tilde{N}_t \tag{A.1.7}$$

for all $t \in [0,T]$. Then, we obtain

$$F_t - Y_t^{\nu^*} + Z_t^{\nu^*} - 2c\nu_t + \tilde{M}_t - 2\phi\int_0^t q_s^{\nu^*}ds = 0, \tag{A.1.8}$$

which implies

$$\begin{aligned} d\nu_t^* &= \frac{1}{2c}\left(dF_t - dY_t^{\nu^*} - dZ_t^{\nu^*} + d\tilde{M}_t - 2\phi q_t^{\nu^*}dt\right) \\ &= \frac{1}{2c}\left(dF_t - dY_t^{\nu^*} - \nu_t^* dt - d\tilde{N}_t + d\tilde{M}_t - 2\phi q_t^{\nu^*}dt\right) \\ \nu_T^* &= \frac{1}{2c}\left(F_T - \rho Y_T^{\nu^*} - Z_T^{\nu^*} + \tilde{M}_T - 2\phi\int_0^T q_t^{\nu^*}dt\right) = \frac{\Phi}{c}q_T^{\nu^*} - \frac{1}{2c}Y_T^{\nu^*} \end{aligned} \tag{A.1.9}$$

Sufficiency: It would be sufficient to show that the Gateaux derivative is zero suppose that $(\nu, X^\nu, Y^\nu, Z^\nu)$ is a solution to the FBSDE in equation (4.2.2) and $\nu \in \mathcal{A}$. Here, we would skip the proof and suggest the interested to read the Neuman and Voss's original paper for detail. □

A.1.2 Proof of Theorem 4.2.3

Proof. By the theorem 3.2 in [19], there exists a unique optimal strategy $\nu^* \in \mathcal{A}$, such that for any signal process A, we have

$$\begin{aligned}\nu_t^* =& v_0(T-t)\Big[v_1(T-t)q_t^{\nu^*} + v_2(T-t)Y_t^{\nu^*} \\ &+ \frac{1}{2c}\left(v_3(T-t)\mathbb{E}_t\left[\int_t^T \frac{S_{4,3}(T-s)}{S_{4,4}(T-t)}dA_s\right] - \mathbb{E}_t\left[\int_t^T \frac{G_3(T-s)}{G_3(T-t)}dA_s\right]\right)\Big].\end{aligned} \tag{A.1.10}$$

Further, by remark 3.1.1 and equation (4.1.4), we know that the signal process $(A_s)_{t\leq s\leq T}$ for the Vasicek process is deterministic given the filtration $\mathcal{F}_t$ up to time t. Hence, we can eliminate the expectation operator $\mathbb{E}_t[\cdot]$, and plug the equation (4.1.4) into equation (A.1.10), so that

$$\begin{aligned}\nu_t^* =& v_0(T-t)\Big[v_1(T-t)q_t^{\nu^*} + v_2(T-t)Y_t^{\nu^*} \\ &+ \frac{1}{2c}\left(v_3(T-t)\int_t^T \frac{S_{4,3}(T-s)}{S_{4,4}(T-t)}dA_s - \int_t^T \frac{G_3(T-s)}{G_3(T-t)}dA_s\right)\Big] \\ =& v_0(T-t)\Big[v_1(T-t)q_t^{\nu^*} + v_2(T-t)Y_t^{\nu^*} \\ &+ \frac{\kappa(\eta-F_t)}{2c}\left(v_3(T-t)\int_t^T e^{-\kappa(s-t)}\frac{S_{4,3}(T-s)}{S_{4,4}(T-t)}ds - \int_t^T e^{-\kappa(s-t)}\frac{G_3(T-s)}{G_3(T-t)}ds\right)\Big]\end{aligned} \tag{A.1.11}$$

□

A.2 Trading with Crowds

To solve the mean-field problem, we first need to decompose the jointly aggregated transient price impact $Y^{\boldsymbol{\nu}}$ into the summation of Y^{ν^j} and $Y^{\boldsymbol{\nu}^{-j}}$ as

$$\begin{aligned} Y_t^{\nu^j} &:= \frac{1}{N}e^{-\rho t} + \frac{\lambda}{N}\int_0^t e^{-\rho(s-t)}\nu_s^j ds \\ Y_t^{\boldsymbol{\nu}^{-j}} &:= \frac{N-1}{N}e^{-\rho t} + \frac{\lambda}{N}\sum_{i\neq j}\int_0^t e^{-\rho(s-t)}\nu_s^i ds \end{aligned} \tag{A.2.1}$$

so that $S_t^{\boldsymbol{\nu}} = F_t - Y_t^{\nu^j} - Y_t^{\boldsymbol{\nu}^{-j}}, \forall t \in [0,T]$.

Lemma A.2.1. *For $\nu^j \in \mathcal{A}^j$, we have*

$$\begin{aligned}\langle \mathcal{R}_0^j(\nu^j, \boldsymbol{\nu}^{-j}), \alpha^j\rangle = \mathbb{E}\Big[\int_0^T \alpha_t^j \Big(F_t - Y_t^{\nu^j} - Y_t^{\boldsymbol{\nu}^{-j}} - \int_t^T e^{-\rho(s-t)}\frac{\lambda}{N}\nu^j ds - 2c\nu_t^j \\ +2\phi\int_y^T q_s^{\nu^j} ds + 2\Phi q_T^{\nu^j} - F_T\Big) dt\Big]\end{aligned} \tag{A.2.2}$$

Proof. Here, we provide a sketch of proof following the hint in [19, Section 9] as a practice. Note that the proof is skipped as it is regarded as trivial by Neuman and Voss, but it is not as trivial as it is to me.

Notice that each agent j characterize its best response by maximizing its objective functional regard the opponents' action $\boldsymbol{\nu}^{-j}$ as given. The Nash equilibrium is achieved when all the agents provide their best responses. Therefore, for each agent j, we are only interested in the Gateaux derivative with respect to ν^j only. Following the similar computation in lemma A.1.1, we obtain

$$\begin{aligned}\langle \mathcal{R}_0^j(\nu^j, \boldsymbol{\nu}^{-j}), \alpha^j\rangle = \mathbb{E}\Big[\int_0^T \alpha_t^j \Big(F_t - Y_t^{\nu^j} - Y_t^{\boldsymbol{\nu}^{-j}} - \int_t^T e^{-\rho(s-t)}\frac{\lambda}{N}\nu^j ds - 2c\nu_t^j \\ +2\phi\int_y^T q_s^{\nu^j} ds + 2\Phi q_T^{\nu^j} - F_T\Big) dt\Big]\end{aligned} \tag{A.2.3}$$

□

A.2.1 Proof of Lemma 4.3.1

Proof. Here, we provide a sketched proof of lemma lemma 4.3.1 following the guidance in [9, Lemma 2.5.] as a practice learning to solve the mean-field problem using Gateaux derivative. Note that the proof is skipped as it is regarded as trivial by Neuman and Voss, but it is a great practice for the proof of lemma 4.4.1.

By lemma A.2.1, we first set $\langle \mathcal{R}_0^j(\nu^j, \boldsymbol{\nu}^{-j}), \alpha^j \rangle = 0$, i.e.,

$$\begin{aligned}\mathbb{E}\Bigg[\int_0^T \alpha_t^j \Bigg(F_t - Y_t^{\boldsymbol{\nu}} - \int_t^T e^{-\rho(s-t)} \frac{\lambda}{N} \nu^j ds - 2c\nu_t^j \\ +2\phi \int_y^T q_s^{\nu^j} ds + 2\Phi q_T^{\nu^j} - F_T \Bigg) dt \Bigg] = 0,\end{aligned} \tag{A.2.4}$$

for all $\alpha^j \in \mathcal{A}^j$. One should quickly notice the similarity between equation (A.1.3) and equation (A.2.4).

Necessity: Assume that $\nu^{j,*}$ maximizes $\mathcal{R}_0^j$ given $\boldsymbol{\nu^{-j}}$, i.e., equation (A.2.4) is satisfied. Then by tower property,

$$F_t - Y_t^{\boldsymbol{\nu}} - \mathbb{E}_t\left[\int_t^T e^{-\rho(s-t)} \frac{\lambda}{N} \nu_s^{j,*} ds\right] - 2c\nu_t^{j,*} + \mathbb{E}_t\left[2\phi \int_t^T q_s^{\nu^{j,*}} ds + 2\Phi q_T^{\nu^{j,*}} - F_T\right] = 0 \tag{A.2.5}$$

with $d\mathbb{P} \bigotimes ds$-a.e. on$\Sigma \times [0, T]$.

Next, define two auxiliary square integrable martingales

$$\tilde{M}_t^j := \mathbb{E}_t\left[2\phi \int_t^T q_s^{\nu^{j,*}} ds + 2\Phi q_T^{\nu^{j,*}} - F_T\right], \quad \tilde{N}_t^j := \mathbb{E}_t\left[\int_0^T e^{-\rho(s-t)} \lambda \nu_s^{j,*} ds\right] \tag{A.2.6}$$

and square integrable process

$$Z_t^{\nu^{j,*}} := \int_0^t e^{-\rho(s-t)} \lambda \nu_s^{j,*} ds - e^{\rho t} \tilde{N}_t^j \tag{A.2.7}$$

for all $t \in [0, T]$. Then, we obtain

$$F_t - Y_t^{\boldsymbol{\nu}} + Z_t^{\nu^{j,*}} - 2c\nu_t^{j,*} + \tilde{M}_t^j - 2\phi \int_0^t q_s^{\nu^{j,*}} ds = 0, \tag{A.2.8}$$

which implies

$$\begin{aligned} d\nu_t^{j,*} &= \frac{1}{2c}\left(dF_t - dY_t^{\boldsymbol{\nu}} - dZ_t^{\nu^{j,*}} + d\tilde{M}_t^j - 2\phi q_t^{\nu^{j,*}} dt\right) \\ \nu_T^{j,*} &= \frac{1}{2c}\left(F_T - \rho Y_T^{\boldsymbol{\nu}} - Z_T^{\nu^{j,*}} + \tilde{M}_T^j - 2\phi \int_0^T q_t^{\nu^{j,*}} dt\right) = \frac{\Phi}{c} q_T^{\nu^{j,*}} - \frac{1}{2c} Y_T^{\nu^*}. \end{aligned} \tag{A.2.9}$$

Sufficiency: The proof of the sufficiency would be similar to that of appendix A.1.1 with slightly change from single person to $\frac{1}{N}$, as we did above. Therefore, here we would skip the proof and leave it as an exercise for whoever reads to here. □

A.2.2 Proof of Corollary 4.3.2

Proof. By definition. For example,

$$\begin{aligned} d\bar{\nu}_t &= \sum_{i \in \mathfrak{N}} d\bar{\nu}_t^i \\ &= \frac{1}{2c} \sum_{i \in \mathfrak{N}} \left(dF_t - dY_t^{\boldsymbol{\nu}} - dZ_t^{\nu^i} + d\tilde{M}_t^i - 2\phi q_t^{\nu^i} dt \right) \\ &= \frac{1}{2c} \left(dF_t - dY_t^{\boldsymbol{\nu}} - d\bar{Z}_t^{\bar{\nu}} + d\bar{M}_t - 2\phi \bar{q}_t^{\bar{\nu}} dt \right) . \end{aligned} \tag{A.2.10}$$

And similarly for the other processes. □

A.2.3 Proof of Corollary 4.3.3

Proof. First, notice that the matrix $\bar{S}$ and the vector $\bar{G}$ defined in corollary 4.3.3 are deterministic. Hence, recall the equation (4.1.4) and by [9, Lemma 2.5], we would obtain

$$\begin{aligned} \bar{\nu}_t^* =& \bar{v}_0(T-t) \left[\bar{v}_1(T-t) \bar{q}_t^{\bar{\nu}^*} + \bar{v}_2(T-t) \bar{Y}_t^{\bar{\nu}^*} \right. \\ &+ \left. \frac{\eta\kappa}{2c} e^{-\kappa t} \left(\bar{v}_3(T-t) \int_t^T \frac{\bar{S}_{4,3}(T-s)}{\bar{S}_{4,4}(T-t)} dt - \int_t^T \frac{\bar{G}_3(T-s)}{\bar{G}_3(T-t)} dt \right) \right] . \end{aligned} \tag{A.2.11}$$

Note that such a $\bar{\nu}_t^*$ is deterministic over timestamp $t \in (0,T)$. Thus by corollary 4.3.3, both $\bar{q}^{\bar{\nu}}$ and $\bar{Y}^{\bar{\nu}}$ are deterministic. Those implies that

$$\begin{aligned} \nu_t^{j,*} =& v_0(T-t) \left[v_1(T-t) q_t^{\nu^*} - \frac{1}{2cG_2(T-t)} \bar{Y}_T^{\bar{\nu}} \right. \\ &+ \left. \frac{1}{2c} \left(v_2(T-t) \int_t^T \frac{S_{3,2}(T-s)}{S_{3,3}(T-t)} \left(dA_s - d\bar{Y}_s^{\bar{\nu}} \right) - \int_t^T \frac{G_3(T-s)}{G_3(T-t)} \left(dA_s - d\bar{Y}_s^{\bar{\nu}} \right) \right) \right] \\ =& v_0(T-t) \left[v_1(T-t) q_t^{\nu^*} - \frac{1}{2cG_2(T-t)} \bar{Y}_T^{\bar{\nu}} \right. \\ &+ \left. \frac{1}{2c} \left(v_2(T-t) \int_t^T \frac{S_{3,2}(T-s)}{S_{3,3}(T-t)} ds - d\bar{Y}_s^{\bar{\nu}} - \int_t^T \frac{G_3(T-s)}{G_3(T-t)} ds - d\bar{Y}_s^{\bar{\nu}} \right) \right] \\ =& v_0(T-t) \left[v_1(T-t) q_t^{\nu^{j,*}} - \frac{1}{2cG_2(T-t)} \bar{Y}_T^{\bar{\nu}^*} \right. \\ &+ \frac{\kappa(\eta - F_t)}{2c} \left(v_2(T-t) \int_t^T \frac{S_{3,2}(T-s)}{S_{3,3}(T-t)} ds - \int_t^T \frac{G_3(T-s)}{G_3(T-t)} ds \right) \\ &- \left. \frac{1}{2c} \left(v_2(T-t) \int_t^T \frac{S_{3,2}(T-s)}{S_{3,3}(T-t)} d\bar{Y}_s - \int_t^T \frac{G_3(T-s)}{G_3(T-t)} d\bar{Y}_s \right) \right] \end{aligned} \tag{A.2.12}$$

□

A.2.4 Proof of Theorem 4.3.5

Proof. Similar to appendix A.2.3. □

A.3 Mean-Field Games with the Same True Belief

Suppose the agent j belongs to the k-th sub-population. To solve the mean-field problem, we first need to decompose the jointly aggregated transient price impact $Y^{\boldsymbol{\nu}}$ into the summation of Y^{ν^j},

$Y^{\boldsymbol{\nu}^{-j}}$ and $Y^{\nu^{-k}}$

$$Y_t^{\nu^j} := e^{\rho t} + \frac{\lambda_k}{N_k} \int_0^t e^{-\rho(s-t)} \nu_s^j ds \tag{A.3.1}$$

$$Y_t^{\nu^j} := e^{\rho t} + \frac{\lambda_k}{N_k} \sum_{i \neq j, i \in \mathcal{K}_k} \int_0^t e^{-\rho(s-t)} \nu_s^i ds + \sum_{i \in \mathcal{K}_{k'}} \frac{\lambda_{k'}}{N_{k'}} \int_0^t e^{-\rho(s-t)} \nu_s^i ds \tag{A.3.2}$$

Lemma A.3.1. *For $\nu^j \in \mathcal{A}^j$, we have*

$$\begin{aligned} \langle \mathcal{R}_0^j(\nu^j, \boldsymbol{\nu}^{-j}), \alpha^j \rangle = \mathbb{E}\Bigg[\int_0^T \alpha_t^j \Bigg(& F_t - Y_t^{\boldsymbol{\nu}} - \int_t^T e^{-\rho(s-t)} \frac{\lambda_k}{N_k} \nu^j ds - 2c_k \nu_t^j \\ & +2\phi_k \int_y^T q_s^{\nu^j} ds + 2\Phi_k q_T^{\nu^j} - F_T \Bigg) dt \Bigg] \end{aligned} \tag{A.3.3}$$

Proof. Similar to appendix A.2.1. □

A.3.1 Proof of Lemma 4.4.1

Proof. By lemma A.3.1, we first set $\langle \mathcal{R}_0^j(\nu^j, \boldsymbol{\nu}^{-j}), \alpha^j \rangle = 0$, i.e.,

$$\begin{aligned} \mathbb{E}\Bigg[\int_0^T \alpha_t^j \Bigg(& F_t - Y_t^{\boldsymbol{\nu}} - \int_t^T e^{-\rho(s-t)} \frac{\lambda_k}{N_k} \nu^j ds - 2c_k \nu_t^j \\ & +2\phi_k \int_y^T q_s^{\nu^j} ds + 2\Phi_k q_T^{\nu^j} - F_T \Bigg) dt \Bigg] = 0, \end{aligned} \tag{A.3.4}$$

for all $\alpha^j \in \mathcal{A}^j$.

Necessity: Assume that $\nu^{j,*}$ maximizes $\mathcal{R}_0^j$ given $\boldsymbol{\nu^{-j}}$, i.e., equation (A.3.4) is satisfied. Then by tower property,

$$F_t - Y_t^{\boldsymbol{\nu}} - \mathbb{E}_t\left[\int_t^T e^{-\rho(s-t)} \frac{\lambda_k}{N_k} \nu_s^{j,*} ds\right] - 2c_k \nu_t^{j,*} + \mathbb{E}_t\left[2\phi_k \int_t^T q_s^{\nu^{j,*}} ds + 2\Phi_k q_T^{\nu^{j,*}} - F_T\right] = 0 \tag{A.3.5}$$

with $d\mathbb{P} \bigotimes ds$-a.e. on$\Sigma \times [0, T]$.

Next, define two auxiliary square integrable martingales

$$\tilde{M}_t^{j,k} := \mathbb{E}_t\left[2\phi_k \int_t^T q_s^{\nu^{j,*}} ds + 2\Phi_k q_T^{\nu^{j,*}} - F_T\right], \quad \tilde{N}_t^{j,k} := \mathbb{E}_t\left[\int_0^T e^{-\rho(s-t)} \lambda_k \nu_s^{j,*} ds\right] \tag{A.3.6}$$

and square integrable process

$$Z_t^{\nu^{j,*},k} := \int_0^t e^{-\rho(s-t)} \lambda \nu_s^{j,*} ds - e^{\rho t} \tilde{N}_t^{j,k} \tag{A.3.7}$$

for all $t \in [0, T]$. Then, we obtain

$$F_t - Y_t^{\boldsymbol{\nu}} + Z_t^{\nu^{j,*},k} - 2c_k \nu_t^{j,*} + \tilde{M}_t^{j,k} - 2\phi_k \int_0^t q_s^{\nu^{j,*}} ds = 0, \tag{A.3.8}$$

which implies

$$\begin{aligned} d\nu_t^{j,*} &= \frac{1}{2c}\left(dF_t - dY_t^{\boldsymbol{\nu}} - dZ_t^{\nu^{j,*},k} + d\tilde{M}_t^{j,k} - 2\phi_k q_t^{\nu^{j,*}} dt\right) \\ \nu_T^{j,*} &= \frac{1}{2c_k}\left(F_T - \rho Y_T^{\boldsymbol{\nu}} - Z_T^{\nu^{j,*},k} + \tilde{M}_T^{j,k} - 2\phi \int_0^T q_t^{\nu^{j,*}} dt\right) = \frac{\Phi_k}{c_k} q_T^{\nu^{j,*}} - \frac{1}{2c_k} Y_T^{\nu^*}. \end{aligned} \tag{A.3.9}$$

□

Bibliography

[1] Robert C Merton. Optimum consumption and portfolio rules in a continuous-time model. In *Stochastic optimization models in finance*, pages 621–661. Elsevier, 1975.

[2] Robert C Merton and Paul Anthony Samuelson. *Continuous-time finance*. Blackwell Boston, 1992.

[3] Álvaro Cartea, Sebastian Jaimungal, and José Penalva. *Algorithmic and high-frequency trading*. Cambridge University Press, 2015.

[4] Alessandro Micheli and Eyal Neuman. Evidence of crowding on russell 3000 reconstitution events. *arXiv preprint arXiv:2006.07456*, 2020.

[5] Markus K Brunnermeier and Lasse Heje Pedersen. Predatory trading. *The Journal of Finance*, 60(4):1825–1863, 2005.

[6] RENÉ A Carmona and Joseph Yang. Predatory trading: a game on volatility and liquidity. *Preprint. URL: http://www. princeton. edu/rcarmona/download/fe/PredatoryTradingGameQF. pdf*, 2011.

[7] Ciamac C Moallemi, Beomsoo Park, and Benjamin Van Roy. Strategic execution in the presence of an uninformed arbitrageur. *Journal of Financial Markets*, 15(4):361–391, 2012.

[8] Alain Bensoussan, Jens Frehse, Phillip Yam, et al. *Mean field games and mean field type control theory*, volume 101. Springer, 2013.

[9] Eyal Neuman and Moritz Voß. Trading with the crowd, 2021.

[10] Xuancheng Huang, Sebastian Jaimungal, and Mojtaba Nourian. Mean-field game strategies for optimal execution. *Applied Mathematical Finance*, 26(2):153–185, 2019.

[11] Philippe Casgrain and Sebastian Jaimungal. Mean-field games with differing beliefs for algorithmic trading, 2018.

[12] Philippe Casgrain, Brian Ning, and Sebastian Jaimungal. Deep q-learning for nash equilibria: Nash-dqn. *CoRR*, abs/1904.10554, 2019.

[13] Rosemarie Nagel. Unraveling in guessing games: An experimental study. *The American economic review*, 85(5):1313–1326, 1995.

[14] Drew Fudenberg, Fudenberg Drew, David K Levine, and David K Levine. *The theory of learning in games*, volume 2. MIT press, 1998.

[15] Drew Fudenberg and David K Levine. Consistency and cautious fictitious play. *Journal of Economic Dynamics and Control*, 19(5-7):1065–1089, 1995.

[16] Josef Hofbauer and William H Sandholm. On the global convergence of stochastic fictitious play. *Econometrica*, 70(6):2265–2294, 2002.

[17] Dileep Kalathil, Naumaan Nayyar, and Rahul Jain. Decentralized learning for multiplayer multiarmed bandits. *IEEE Transactions on Information Theory*, 60(4):2331–2345, 2014.

[18] Sriram Ganapathi Subramanian, Matthew E. Taylor, Mark Crowley, and Pascal Poupart. Decentralized mean field games. *CoRR*, abs/2112.09099, 2021.

[19] Eyal Neuman and Moritz Voß. Optimal signal-adaptive trading with temporary and transient price impact. *SIAM Journal on Financial Mathematics*, 13(2):551–575, 2022.

[20] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, Jan 2016.

[21] Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, Nov 2019.

[22] Asuman Ozdaglar, Muhammed O. Sayin, and Kaiqing Zhang. Independent learning in stochastic games, 2021.

[23] Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008.

[24] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control*, pages 321–384, 2021.

[25] Lloyd S Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10):1095–1100, 1953.

[26] Junling Hu and Michael P Wellman. Nash q-learning for general-sum stochastic games. *Journal of machine learning research*, 4(Nov):1039–1069, 2003.

[27] Tjalling Koopmans. *Activity analysis of production and allocation*. Wiley, 1951.

[28] Dov Monderer and Lloyd S Shapley. Fictitious play property for games with identical interests. *Journal of economic theory*, 68(1):258–265, 1996.

[29] Muhammed O Sayin, Francesca Parise, and Asuman Ozdaglar. Fictitious play in zero-sum stochastic games. *SIAM Journal on Control and Optimization*, 60(4):2095–2114, 2022.

[30] Muhammed O. Sayin, Kaiqing Zhang, David S. Leslie, Tamer Basar, and Asuman E. Ozdaglar. Decentralized q-learning in zero-sum markov games. *CoRR*, abs/2106.02748, 2021.

[31] David S Leslie and Edmund J Collins. Individual q-learning in normal form games. *SIAM Journal on Control and Optimization*, 44(2):495–514, 2005.

[32] Jeffrey C Ely and Okan Yilankaya. Nash equilibrium and the evolution of preferences. *journal of Economic Theory*, 97(2):255–272, 2001.

[33] William H Sandholm. Preference evolution, two-speed dynamics, and rapid social change. *Review of Economic Dynamics*, 4(3):637–679, 2001.

[34] Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. Mean field multi-agent reinforcement learning. *CoRR*, abs/1802.05438, 2018.

[35] Wikipedia contributors. Vasicek model — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Vasicek_model&oldid=1106586278`, 2022. [Online; accessed 26-August-2022].

[36] Oldrich Vasicek. An equilibrium characterization of the term structure. *Journal of financial economics*, 5(2):177–188, 1977.

[37] Yujing Hu, Weixun Wang, Hangtian Jia, Yixiang Wang, Yingfeng Chen, Jianye Hao, Feng Wu, and Changjie Fan. Learning to utilize shaping rewards: A new approach of reward shaping. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 15931–15941. Curran Associates, Inc., 2020.

[38] Adam D. Laud. Theory and application of reward shaping in reinforcement learning. *ProQuest Dissertations and Theses*, page 97, 2004.

[39] Ben Somner-Bogard and Tianhao Wang. Stochastic control in finance project: Merton portfolio optimisation with consumption. `https://github.com/PINGXIANG/CoursePaper/blob/main/Merton%20Portfolio%20Optimisation%20with%20Consumption.pdf`, 2022. unpublished.

[40] Wikipedia contributors. Group cognition — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Group_cognition&oldid=1101012557`, 2022. [Online; accessed 29-August-2022].

[41] Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45, 2012.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.