Algorithmic Pricing and Liquidity in Securities Markets*

Jean-Edouard Colliard, Thierry Foucault, and Stefano Lovo

December 17, 2023

Abstract

We let "Algorithmic Market Makers" (AMs), using Q-learning algorithms, determine prices for a risky asset in a standard market making game with adverse selection and compare these prices to the Nash equilibrium of the game. We observe that AMs effectively adapt to adverse selection, adjusting prices post-trade as anticipated. However, AMs charge a markup over the competitive price and this markup increases when adverse selection costs decrease, in contrast to the predictions of the Nash equilibrium. We attribute this unexpected pattern to the diminished learning capacity of AMs when faced with increased profit variance.

Keywords: Algorithmic pricing, Market Making, Adverse Selection, Market Power, Reinforcement learning.

JEL classification: D43, G10, G14.

*Correspondence: colliard@hec.fr, foucault@hec.fr, lovo@hec.fr. All authors are at HEC Paris, Department of Finance, 1 rue de la Libération, 78351 Jouy-en-Josas, France. We are grateful to Winston Dou, Terrence Hendershott, Anton Lines, Mao Ye, Bart Yueshen, participants in "The Microstructure Exchange", the Microstructure Asia Pacific Online Seminar, the 2022 Oxford Artificial Intelligence and Financial Markets Workshop, the 2023 NYU Stern Microstructure Conference, the 2023 Western Finance Association Meetings, the 2023 European Finance Association Meetings, the 2023 Financial Markets Liquidity Conference, the 2023 Luiss Finance Workshop, the 2023 CFM-Imperial conference and seminar participants at Aalto University, Bank of England, Bank of France, Bundesbank, Cornell University, CRESE, Frankfurt School of Management, HEC Paris, Keio University, Paris School of Economics, Peking University, Tokyo University, University College London, University of Copenhagen, and University Paris 1, for helpful comments and suggestions. We thank Olena Bogdan, Amine Chiboub, Pietro Fadda, Chhavi Rastogi, and Andrea Ricciardi for excellent research assistance. This work was supported by the French National Research Agency (F-STAR ANR-17-CE26-0007-01, ANR EFAR AAP Tremplin-ERC (7) 2019), the Investissements d'Avenir Labex (ANR-11-IDEX-0003/Labex Ecodec/ANR-11-LABX-0047), the Chair ACPR/Risk Foundation "Regulation and Systemic Risk", the Natixis Chair "Business Analytics for Future Banking".

Introduction

Prices are increasingly set by algorithms in financial markets. For instance, Brogaard *et al.* (2014) and Chaboud *et al.* (2019) find, respectively, that about 42% and 60% of trades in stocks and currencies in their sample take place at prices set by algorithms. In U.S treasury markets, principal trading firms (PTFs), which also rely on algorithms, account for 21% of total trading volume (Brain *et al.* (2018)). Even in residential real-estate, pricing algorithms are now used (by intermediaries such as OpenDoor or Offerpad) to make cash offers to homeowners (Buchak *et al.* (2019)). Given this evolution, to understand price formation in financial markets, researchers must study how algorithms behave.[1] This requires examining how they respond to adverse selection and discover asset values, two important features specific to financial markets.[2]

Our goal in this paper is to study this question. One possibility is that algorithmic pricers simply behave as portrayed in existing models of market making (pioneered by Glosten and Milgrom (1985) and Kyle (1985)) but with enhanced efficiency (e.g., by reacting faster to information). However, a central tenet of these models is that marker makers behave like Bayesian learners whereas this is not how recent powerful decision-making algorithms (e.g., those used for playing Go or determining on-line retailers' prices) behave. Instead, they start with no prior about their environment and learn iteratively how to make decisions by experimenting, receiving feedback, and adjusting their behavior accordingly. This approach, "reinforcement learning," has achieved notable success and is therefore a plausible alternative for modeling pricing algorithms. Consequently, our paper undertakes a comparative analysis of market maker behavior using reinforcement learning algorithms against the predictions of Bayesian learning models of market making.

More specifically, we consider a market making game similar to Glosten and Milgrom (1985) and we study, via extensive simulations (experiments), how market makers using Q-learning algorithms (a specific type of reinforcement learning algorithm) set their prices and how these compare to those

[1] For instance, Goldstein *et al.* (2021) notes that *"Just as insights into human behavior from the psychology literature spawned the field of behavioral finance, so can insights into algorithmic behavior (or the psychology of machines) spawn an analogous blossoming of research in algorithmic behavioral finance."*

[2] See *"Why the wisdom of the market crowd beats AI"*, Financial Times, May 22, 2023, for a skeptical view on whether algorithms have the ability to discover the fair value of an asset. The failure of one "iBuyer" (Zillow) also suggests that learning to price in adverse selection might not be so easy for algorithms. See Buchak *et al.* (2019) or *"Zillow sent its algorithm to take on the housing market. The housing market won"*, November 9, 2021, The Washington Post.

predicted by the Nash equilibrium of the market making game, where market makers are Bayesian.[3] In the market making game one client wishes to buy one share of a risky asset and requests quotes from $N \geq 2$ market makers. These simultaneously respond with an offer and the client buys if the best offer is less than her valuation for the asset, which is the sum of the payoff of the asset and a private valuation (the client's "liquidity shock"). Thus, holding market makers' prices constant, the client is more likely to buy the asset when its payoff is high than when its payoff is low. Market makers are therefore exposed to adverse selection.

In the Nash equilibrium of this game, market makers post prices equal to their expectation of the asset payoff conditional on the client accepting their offer, so that their expected profit is zero. Thus, their quoted spread (their markup relative to the unconditional expected value of the asset) is just equal to the adverse selection cost. This property has several implications. First, the quoted spread decreases with the dispersion of the client's liquidity shocks (σ) because this reduces the adverse selection cost. Second, an increase in the volatility of the asset payoff (Δ_v) has the opposite effect. Last, these two predictions do not depend on the number of market makers (for $N \geq 2$).

The core of our analysis consists in testing whether these (very standard) predictions are satisfied when quotes are set by Algorithmic Market Makers (henceforth "AMs") who play the market making game using Q-learning algorithms with no initial knowledge of the environment (the distribution of the asset payoff and the client's valuation, the number of market makers etc.). Consequently, AMs must play the game repeatedly to learn the average profit ("Q-value") they can obtain at each price and bid accordingly. Thus, for a given parametrization of the market making game (a choice of (σ, Δ_v, N)), we let AMs receive requests sequentially from a large number of different clients (1 million in our experiments). For each new request, AMs simultaneously and independently choose a price following an algorithm that either picks a price randomly in a fixed set ("explore") or pick the price with the largest Q-value at the moment the client arrives ("exploit"). This exploration/exploitation choice is random with a decaying probability of exploration.[4] After the client's decision is made, the Q-value of the price chosen by each AM is updated by taking a

[3] We use simulations because even though the Q-learning algorithm is very simple, we cannot solve for the long run prices chosen by the algorithms analytically. This is a standard issue (and approach) in the analysis of Q-learning algorithms.

[4] Q-learning algorithms are usually specified so that the likelihood of exploration decays over time because exploring is costly (it requires taking an action that, according to the AMs' assessment, seems suboptimal) and, intuitively, less useful in terms of learning as experience accumulates. The decaying rate for exploration is a parameter of the algorithm.

weighted average of its *realized profit* with the client and the Q-value of this price just before the client's decision.[5]

A key feature of this iterative process (the Q-learning algorithm) is that each AM gradually learns, via exploration, the average profit it can obtain at each price. For a given environment, the path of prices chosen by an AM is stochastic because (a) the client's decision is stochastic, (b) the asset payoff is stochastic and (c) the prices chosen by AMs are stochastic (since they experiment). Consequently, the long run Q-value of each price and therefore the prices eventually chosen by the AMs are also stochastic. Thus, for each parametrization of the market making game, we run $1,000$ different simulations (experiments) and we focus on the average long-run outcomes across these experiments.

We observe several interesting regularities, most at odds with the predictions of the Nash equilibrium. For all parametrizations, in the long run, AMs always charge prices such that their true expected profit is positive. Thus, they learn how to price adverse selection. However, their prices are frequently well above the Nash equilibrium price. As a result, AMs' average realized spreads (across experiments) are large compared to those predicted by the Nash equilibrium. However, AMs "leave money on the table:" Each AM could obtain a larger expected profit by (unilaterally) undercutting the price to which the others have eventually settled.[6] Hence AMs do not learn to undercut each other down to the competitive price. Moreover, in contrast to the prediction of the Nash equilibrium, we observe that AMs' average quoted spread increases with the dispersion of clients' liquidity shocks, despite the fact that their adverse selection cost declines. As a result, AMs' average realized spread (rents) increases with the dispersion of clients' liquidity shocks.

Surprisingly, this finding suggests that adverse selection makes it easier for AMs to "learn to undercut" (lowering the price to attract demand). To analyze this point in more details, we also run simulations without adverse selection, holding constant the likelihood of a trade at a given price. To this end, we assume that the distribution of clients' valuations is identical to that in our baseline experiments but that this valuation is uncorrelated with the asset payoff. In this case, other things equal, AMs' average quoted spreads are smaller than when there is adverse selection. This observation confirms that AMs learn not to be adversely selected (they widen their spreads when

[5] A new realization of the asset payoff is drawn after each client's arrival. Moreover, these realizations and clients' liquidity shocks are i.i.d. Thus, each request is exactly a repetition of the static market making game.

[6] In most cases, in a given experiment, AMs end up posting the same price.

there is adverse selection). However, other things equal, AMs' average realized spreads are *larger* in the experiments with no adverse selection, confirming our conjecture that adverse selection makes AMs *more* competitive.

In experiments with adverse selection, we also observe that, other things equal, the AMs' average quoted spread increases with the volatility of the asset payoff, as the Nash equilibrium predicts. However, we find the same pattern when there is *no* adverse selection. This is surprising because, in this case, the Nash equilibrium implies that AMs' average quoted spread should be nil and therefore insensitive to the volatility of the asset payoff. This finding suggests that an increase in the asset volatility makes it more difficult for AMs to learn to undercut. This is confirmed by the fact that AMs' average realized spreads increase with the asset volatility when there is no adverse selection. Interestingly, this effect is much dampened when there is adverse selection and AMs' average realized spreads in this case are smaller than without adverse selection, for each value of the asset payoff volatility. These observations again suggest that adverse selection makes it easier for AMs to learn to undercut.

Last, we observe that an increase in the number of AMs (N) leads to smaller average quoted and realized spreads.[7] This observation seems intuitive (when the number of competitors increases, prices become more competitive). However, the Nash equilibrium of the market making game predicts that prices should be identical for all $N \geq 2$. Moreover, we observe that even with 10 AMs, their average quoted and realized spreads remain above those predicted by the Nash equilibrium.

Overall, the behavior of AMs using Q-learning algorithms deviates significantly from that predicted by the Nash equilibrium of this game, even after a long learning phase. First, they post quotes well above competitive quotes, despite the fact that the market making (one-shot) game has no "collusive" equilibria.[8] Moreover, they behave more competitively when the adverse selection cost is higher. This finding is puzzling. It implies that, other things equal, average realized bid-ask spreads should be inversely related to adverse selection costs (e.g., larger in Treasuries than in stocks).

[7] Interestingly, this pattern is also found empirically by Brogaard and Garriott (2019) who study the effects of entry of high-frequency market makers on the liquidity of Canadian stocks.

[8] There are surprisingly few empirical papers on the effect of pricing algorithms on realized bid-ask spreads in securities markets. An exception is Hendershott *et al.* (2011), who find that algorithmic trading (AT) *increases* dealers' *realized* bid-ask spreads (profits). Commenting on this result, they write (on p.4): *"This is surprising because we initially expected that if AT improved liquidity, the mechanism would be competition between liquidity providers."*

We propose the following explanation for these findings. To reach competitive prices, AMs must learn to undercut each other. In early episodes, they all explore by playing random prices. Gradually, they learn how to best respond to their competitors by lowering their price. Each AM then learns how to best respond to this best response, etc. In order to "learn to undercut," an AM must estimate the expected payoff from undercutting and obtain a higher estimate than that for the current price. This estimation is difficult because the client's demand, the asset value, and other AMs' behavior are stochastic. Hence, a large number of explorations is required to learn that undercutting a certain price is profitable. As the AMs explore less as time passes, they learn to undercut only a limited number of times, which is typically too low to reach the competitive price.

Intuitively, in our setting, the acuteness of this issue is determined by the true variance of AMs' profits at a given price. Indeed, when this variance is larger, AMs' realized profit at a given price (their feedback from the environment) is a noisier estimate of their true expected profit at this price (holding their competitors' price fixed). Thus, for a given exploration rate, an increase in the volatility of AMs' profits at a given price implies that the long-run Q-value of this price is a less precise estimate of the true expected profit at this price. AMs can therefore end up underestimating the benefit of undercutting the long run price on which they settle and fail to learn to undercut down to the competitive level.

The effects of the dispersion of clients' liquidity shocks (σ) and the volatility of the asset payoff (Δ_v) on AMs' prices are consistent with this interpretation. In the no adverse selection case, an increase in σ or Δ_v raises the variance of AMs' profits because they increase respectively (i) the variance of the trading volume for each AM ("demand risk") and (ii) the variance of the value of their position ("inventory risk"). Thus, it is more difficult for AMs to precisely estimate the true expected profit of lowering their price. Thus, they learn less quickly to undercut each other and settle on a less competitive price. The same intuitions apply when there is adverse selection. However, other things equal, adverse selection reduces the variance of AMs' profits because it increases the mass of profits close to zero. Thus, surprisingly, it makes it easier for Q-learning algorithms to learn to undercut, which explains why their prices are closer to those predicted by the Nash equilibrium when adverse selection costs are larger.

In existing models (Kyle (1985) or Glosten and Milgrom (1985)), market makers discover asset payoffs dynamically by updating their quotes based on the order flow. To study whether AMs do

so, we modify the market making game to allow AMs to sequentially receive requests from two clients before the asset pays off. Then, as in our baseline experiments, we let AMs using Q-learning algorithms learn how to play the "two-period" market making game. In this case AMs can make their price for the second client contingent on the outcome with the first client. Hence, for each possible price, AMs must estimate their average profit in the second period contingent on each possible outcome for them ("state") after the first client's decision. We obtain two interesting results. First, as predicted by the Nash equilibrium of the two-period market making game, AMs raise their quotes if the first client buys the asset and decrease their quotes otherwise. However, in contrast to the Nash equilibrium, their update is too large in the former case and far too small in the latter, which leads therefore to larger average realized spreads in the second period than in the Nash equilibrium. Intuitively, AMs have fewer opportunities to estimate the average profit in each possible state in the second period than in the first period. As a result, their estimation of the expected profit of each price in each possible state in the second period is noisier and, as in the one period case, this effect makes them less competitive. This finding suggests that, empirically, algorithmic pricing should be associated with stronger price reversals following trades.

Last, it is worth stressing that our goal is not to study how market making algorithms should be designed. We just use Q-learning as a behavioral model of pricing algorithms in financial markets because it captures the essence of more complex decision making algorithms. Thus, it is a good starting point to predict and explain the effects of pricing algorithms in financial markets. For instance, as explained previously, our analysis implies that with algorithmic pricing, one should observe empirically a *negative* effect of shocks reducing adverse selection costs (e.g., greater firms' disclosure) on dealers' realized spreads and more pronounced price reversals after trades. Future research can test these predictions and assess their robustness when other models of behavior for algorithms are used.

In the next section, we position our contribution in the literature. Section 2 presents the market making game and its Nash equilibrium. Section 3 describes the Q-learning algorithms used by AMs in our experiments and reports our experimental results. In Section 4, we interpret these results and in Section 5, we study the two-periods market making game. Section 6 concludes. Some formal derivations are in the appendix and an online appendix provides additional results.

1 Contribution to the Literature

Our paper is related to the emerging literature on algorithmic pricing and the possibility for algorithms to sustain non competitive outcomes.[9] Calvano *et al.* (2020) show that Q-learning algorithms can learn dynamic collusive strategies in a repeated differentiated Bertrand game. Asker *et al.* (2023) and Abada *et al.* (2022) show that supra competitive prices can be reached in this type of environment even if collusive strategies (via dynamic punishment strategies) are ruled out theoretically, through what Abada *et al.* (2022) call "collusion by mistake".[10] Banchio and Skrzypacz (2022) find that Q-learning algorithms post less competitive bids in first price auctions than in second price auctions. Banchio and Mantegazza (2022) show how reinforcement learning can be approximated with a continuous time system of differential equations. In contrast to our setting, in these models, player's payoff are deterministic and the only source of noise an algorithm faces in estimating its action's payoffs comes from the stochastic play of the other algorithms. For example, in Banchio and Skrzypacz (2022), bidders and sellers have a fixed valuation for the auctioned good and bidders are not exposed to adverse selection in their setting (they consider private value auctions).

In line with other papers, we find that pricing algorithms relying on Q-learning can lead to non competitive outcomes even when dynamic strategies are ruled out and when price setters compete in prices. However, new to the literature, we find that adverse selection mitigates this issue.[11] To our knowledge, we are the first to study how market makers using Q-learning algorithms behave in presence of adverse selection.[12] Dou *et al.* (2023) study how informed traders using Q-learning algorithms behave in a Kyle (1985)'s environment. Their analysis and ours are complementary: We focus on market makers' pricing behavior while Dou *et al.* (2023) focus on informed investors' order submission strategies. Interestingly, they find that, in noisier environments, informed investors behave less competitively (submit orders of smaller sizes and get larger average profits). This

[9] Regulators have expressed concerns about this possibility in online retailers' markets (see MacKay and Weinstein (2022), Competition Market Authority (2018), OECD (2017)). We are not aware of similar concerns expressed for securities markets so far.

[10] This idea is in line with an earlier literature in machine learning showing that games between Q-learning algorithms do not necessarily converge to a Nash equilibrium (Wunder *et al.*, 2010). See also Waltman and Kaymak (2008) for an application to Cournot competition.

[11] Another uncommon feature of our setting is that the demand faced by pricing algorithms is stochastic. See also Hansen *et al.* (2021), Cartea *et al.* (2022b), or Wilk (2022) for other settings in which selling algorithms face a stochastic demand elasticity, but without adverse selection.

[12] Cont and Xiong (2023) and Guéant and Manziuk (2019) study how market makers using reinforcement algorithms set prices in the face of inventory holding costs. However, there is no adverse selection in their framework.

observation echoes our finding that an increase in the variance of AMs' profits (e.g., due to an increase in the dispersion of their clients' liquidity demands) leads AMs to settle on less competitive prices. As in their set-up, this finding is related to the collusion by mistake (or, using the terminology of Dou *et al.* (2023), "artificial stupidity") phenomenon. Cartea *et al.* (2022a) and Cartea *et al.* (2022b) study different families of reinforcement learning algorithms and develop new methods to study which ones may converge to non Nash behavior in a market making environment.[13]

Our paper also contributes to the literature on algorithmic trading in securities markets. The theoretical literature on this issue (e.g., Biais *et al.* (2015), Budish *et al.* (2015), Menkveld and Zoican (2017), Baldauf and Mollner (2020), etc.) has mainly focused on how the increase in the speed with which algorithms can respond to information increases or reduces liquidity suppliers' exposure to adverse selection, using traditional workhorses models (Glosten and Milgrom (1985) or Kyle (1985)). Yet, O'Hara (2015) calls for the development of new methodologies to study the effects of algorithms in financial markets, writing that as a result of algorithmic trading: *"the data that emerge from the trading process are consequently altered [...] For microstructure researchers, I believe these changes call for a new research agenda, one that recognizes how the learning models used in the past are lacking [...]."*

Our paper responds to this call. Instead of modeling algorithmic traders as Bayesian learners, with an omniscient knowledge of the environment in which they operate, we model them as Q-learning algorithms. These algorithms learn by trial and error with almost no prior knowledge of the environment, which represents the polar opposite of standard Bayesian learning. Moreover, Q-learning is relatively simple and transparent, which makes it a good candidate for a workhorse model of algorithmic interaction. As explained in the introduction, this approach generates strikingly different predictions for those of canonical Bayesian-learning models, some consistent with empirical findings about algorithmic trading (see Footnotes 7 and 8).

[13] In particular, Cartea *et al.* (2022b) show that using a finer pricing grid (a lower "tick size") reduces the scope for collusion. Pouget (2007) compares two trading mechanisms: A call market and a Walrasian tatonnement in an environment in which both mechanisms have the same Nash equilibrium outcomes. Using computer simulations, he finds that when traders learn via a reinforcement learning model, convergence to equilibrium is achieved in the Walrasian tatonnement but not in the call market.

2 The Economic Environment

In this section, we describe the market making game played by Q-learning algorithms in the experiments and the Nash equilibrium of this game, which serves as a benchmark.

2.1 The Market Making Game

One investor ("client") wants to buy one share of a risky asset.[14] The asset payoff, $\tilde{v}$, has a binary distribution, $\tilde{v} \in \{v_L, v_H\}$, with $\Delta_v := v_H - v_L \geq 0$ and $\mu := \Pr(\tilde{v} = v_H)$. This payoff is realized before trading starts but it is publicly disclosed only after the investor makes her trading decision.

The client privately knows her own valuation for the asset, that is equal to $\tilde{v}^C = \tilde{w}^C + \tilde{L}$. We consider two cases: the *adverse-selection case*, where $Pr(\tilde{v} = \tilde{w}^C) = 1$, and the *no-adverse-selection case*, where $\tilde{v}$ and $\tilde{w}^C$ are i.i.d. In both cases, $\tilde{L}$ is normally distributed with mean zero and variance σ^2, and is independentl from $\tilde{v}$ and $\tilde{w}^C$. We refer to $\tilde{L}$ as *the client's liquidity shock* and denote its c.d.f by $G(.)$. The distribution of $\tilde{v}^C$ is a mixture of two normal distributions with means v_L or v_H, respectively (see Figure 1).

[INSERT FIGURE 1 ABOUT HERE]

After observing her valuation, the client requests quotes from N dealers, who simultaneously respond by posting a price (a_n for dealer n) at which they are willing to sell up to one share of the asset. We denote $\bar{a} = \{a_n\}_{1 \leq n \leq N}$ the vector of prices, $a^{\min} := \min_n\{a_n\}$ the best offer (i.e., the lowest price), and $N_{\min}$ the number of dealers posting this offer. The client buys if and only if the best offer is less than her valuation ($a^{\min} \leq \tilde{v}^C$).

Let $V(a^{\min}, \tilde{v}^C)$ be the client's realized demand (volume of trade). It is 1 if the client buys the asset and 0 otherwise. Dealer n's realized trading volume is:

$$I(a_n, \bar{a}, \tilde{v}^C) := V(a^{\min}, \tilde{v}^C) Z(a_n, \bar{a}), \tag{1}$$

where $Z(a_n, \bar{a}) = \frac{1}{N_{\min}}$ if $a_n = a^{\min}$ (the client's demand is split equally among the dealers posting

[14] We only consider the case in which the client is a buyer. This simplifies the analysis without changing the economics of the problem.

the best offer) and $Z(a_n, \bar{a}) = 0$ otherwise. Hence, dealer n's realized profit is:

$$\Pi(a_n, \bar{a}, \tilde{v}^C, \tilde{v}) := I(a_n, \bar{a}, \tilde{v}^C_\tau)(a^{\min} - \tilde{v}). \tag{2}$$

Dealers' Expected Profit. Consider the adverse-selection case ($\tilde{w}^C = \tilde{v}$) first. In this case, when the asset payoff is v, the client trades with probability:

$$D(a^{\min}, v) := \Pr(a^{\min} \leq \tilde{v}^C \mid \tilde{v} = v) = \Pr(a^{\min} \leq v + \tilde{L}) = 1 - G(a^{\min} - v). \tag{3}$$

Thus, holding the best price constant, a client is more likely to buy the asset when its payoff is high than when it is low since $D(a^{\min}, v_H) > D(a^{\min}, v_L)$ (see Figure 1). Dealers are therefore exposed to adverse selection. Let $\Delta_D(a^{\min})$ be the difference between the likelihoods of a buy when $v = v_H$ and $v = v_L$ (the red area in Figure 1):

$$\Delta_D(a^{\min}) := D(a^{\min}, v_H) - D(a^{\min}, v_L) > 0. \tag{4}$$

This difference decreases in σ, the dispersion of the client's liquidity shock $\tilde{L}$ (for $v_L \leq a^{\min} \leq v_H$), and increases in Δ_v, the volatility of the asset.[15] Dealers are therefore less exposed to adverse selection when σ increases or Δ_v decreases.

As $\tilde{w}^C = \tilde{v}$, we deduce from (2) that dealer n's expected profit, $\bar{\Pi}(a_n, \bar{a}; \mu) := \mathbb{E}_\mu(\Pi(a_n, \bar{a}, \tilde{v}^C_\tau, \tilde{v}))$, is:

$$\bar{\Pi}(a_n, \bar{a}; \mu) = Z(a_n, \bar{a})[\mu D(a^{\min}, v_H)(a^{\min} - v_H) + (1-\mu) D(a^{\min}, v_L)(a^{\min} - v_L)], \tag{5}$$

which can be written as:

$$\bar{\Pi}(a_n, \bar{a}; \mu) = \underbrace{Z(a_n, \bar{a}) \mathbb{E}_\mu(V(a^{\min}, \tilde{v}^C))}_{\text{Dealer's expected trading volume}} \left[\overbrace{(a^{\min} - \mathbb{E}_\mu(\tilde{v}))}^{\text{Quoted spread}} - \underbrace{\Delta_v \frac{(1-\mu)\mu \Delta_D(a^{\min})}{\mathbb{E}_\mu(V(a^{\min}, \tilde{v}^C)))}}_{\text{Adverse selection cost}} \right], \tag{6}$$

where $\mathbb{E}_\mu(\tilde{v}) := \mu v_H + (1-\mu) v_L$ and $\mathbb{E}_\mu(V(a^{\min}, \tilde{v}^C)) := \mu D(a^{\min}, v_H) + (1-\mu) D(a^{\min}, v_L)$ is the

[15] Indeed, remember that $D(a^{\min}, v) = 1 - G(a^{\min} - v)$, where $G(.)$ is the c.d.f of a Gaussian variable with mean zero and variance σ^2. Thus, $\frac{\partial D(a^{\min}, v)}{\partial \sigma} = \frac{exp(-\frac{(a^{\min} - v)^2}{2\sigma^2})}{\sqrt{2\pi\sigma^2}}(a^{\min} - v)$. It immediately follows that $\frac{\partial \Delta_D(a^{\min})}{\partial \sigma} < 0$ for $a^{\min} \in [v_L, v_H]$. A similar reasoning implies that $\frac{\partial \Delta_D(a^{\min})}{\partial \Delta_v} > 0$.

expected trading volume (the likelihood of a buy). The term in bracket in (6) is dealer n's expected profit per share conditional on a trade. It is equal to the dealer's "quoted spread" $(a^{\min} - \mathbb{E}_\mu(\tilde{v}))$, which is what the dealer would earn on average in the absence of adverse selection costs, *minus* the adverse selection cost.

Now consider the no-adverse-selection case, $\tilde{w}^C$ and $\tilde{v}$ are i.i.d. In this case, the likelihood that a client buys the asset when the best price is $a^{\min}$ is $\mathbb{E}_\mu(V(a^{\min}, \tilde{v}^C))$ whether the asset payoff is high or low. Consequently, the adverse selection cost is nil and therefore

$$\bar{\Pi}(a_n, \bar{a}; \mu) = Z(a_n, \bar{a})\mathbb{E}_\mu(V(a^{\min}, \tilde{v}^C))[a^{\min} - \mathbb{E}_\mu(\tilde{v})]. \tag{7}$$

Illiquidity Measures. We measure a client's average "trading cost" using two standard measures of illiquidity, namely the expected (half) quoted spread, $\overline{QS} := \mathbb{E}(a^{\min} - \tilde{v})$, and the expected (half) realized spread, $\overline{RS} := \mathbb{E}(a^{\min} - \tilde{v} \mid \tilde{v}^C > a^{\min})$. The expected realized spread differs from the quoted spread because it is computed using realizations of $(a^{\min} - \tilde{v})$ (the realized profits of dealers' posting the best price) *only* when trades happen $(\tilde{v}^C > a^{\min})$. The realized spread is often used by empiricists to measure dealers' average profits per share traded while the difference between the realized and the quoted spread is a measure of adverse selection costs.[16] Dealer n's expected profit (6) can be written as:

$$\bar{\Pi}(a_n, \bar{a}; \mu) = \underbrace{Z(a_n, \bar{a})\mathbb{E}_\mu(V(a^{\min}, \tilde{v}^C))}_{\text{Dealer's expected trading volume}} \overbrace{[a^{\min} - \mathbb{E}_\mu(\tilde{v} \mid \tilde{v}^C > a^{\min})]}^{\text{Expected realized spread}}, \tag{8}$$

2.2 Glosten-Milgrom Benchmark

Let a^* be the lowest price such that if $a^{\min} = a^*$ then dealers obtain zero expected profits. In our setting, this price is the Bertrand-Nash equilibrium of the market making game. This outcome is often the focus of the literature on market making (e.g., Glosten and Milgrom (1985) or Kyle (1985)). We use this "Glosten-Milgrom price" to benchmark our experiments.

[16] See Foucault *et al.* (2013), Ch. 2, for a description of various measures of bid-ask spreads in securities markets and their interpretation.

With adverse selection, the Glosten-Milgrom price is (use (6) and (8)):

$$a^* = \mathbb{E}_\mu(\tilde{v} \mid \tilde{v}^C > a^*) = \mathbb{E}_\mu(\tilde{v}) + \underbrace{\Delta_v \frac{(1-\mu)\mu\Delta_D(a^*)}{\mathbb{E}_\mu(V(a^*, \tilde{v}^C))}}_{\text{Adverse selection cost}}, \tag{9}$$

Thus, in the Glosten-Milgrom benchmark, the quoted spread $a^* - \mathbb{E}_\mu(\tilde{v})$ is strictly positive (just equal to the adverse selection cost) while the expected realized spread is nil.[17] In the no-adverse-selection case, $a^* = \mathbb{E}_\mu(\tilde{v})$ (use (7)). Thus, dealers' quoted and realized spreads are nil.

In sum, the Glosten-Milgrom benchmark yields four testable hypotheses (see Appendix A.3) about dealers' prices:

1. **H.1.** In the adverse-selection case, the dealers' quoted spread $\overline{QS}$ is strictly positive, decreases with the dispersion of clients' liquidity shocks σ, and increases with the volatility of the asset payoff Δ_v.

2. **H.2.** In the no-adverse-selection case, the dealers' quoted spread $\overline{QS}$ is nil no matter the dispersion of clients' liquidity shocks σ nor the volatility of the asset payoff Δ_v.

3. **H.3.** In both cases, the dealers' expected realized spreads $\bar{RS}$ are zero (dealers make zero expected profits).

4. **H.4.** In both cases, the dealers' quoted spreads and realized spreads do not depend on the number of dealers for $N \geq 2$.

Our experiments test whether these hypotheses predict well the prices set by Algorithmic Market Makers (see the next section). We focus on these hypotheses because they are very standard properties of models of market making in finance. In particular, the fact that quoted spreads should decline with the dispersion of traders' private valuations (σ) or increase with the volatility of the asset payoff when there is adverse selection (see Table 1 for a numerical example) is robust to modeling details (we are not aware of a model predicting that dealers would charge large spreads when adverse selection costs decline). Moreover, the last hypothesis just states the well-known result that it takes only 2 price competitors to reach the Bertrand-Nash equilibrium.

[17] The Glosten-Milgrom price is the solution of a fixed point problem (equation (9)) for which there is no closed-form solution given our specification of $G(.)$. This problem always has at least one solution (when there are more than one, the Glosten-Milgrom price is the smallest root of (9)). See Appendix A.3.

[INSERT TABLE 1 ABOUT HERE]

Remark. One may wonder why we need to consider the case in which $\tilde{w}^C$ and $\tilde{v}$ are i.i.d. After all, when $\tilde{w}^C = \tilde{v}$, one can vary dealers' exposure to adverse selection by varying σ or Δ_v and when $\Delta_v = 0$, there is no adverse selection. This is true but these parameters also affect the likelihood of a trade ($\mathbb{E}_\mu(V(a^{\min}, \tilde{v}^C))$). Thus, in our experiments with adverse selection, when we vary Δ_v or σ, observed effects can be due to (i) variation in the likelihood of a trade, (ii) variation in adverse selection costs or (iii) both.[18] By considering the cases with and without adverse selection, we can therefore better isolate the effect of adverse selection, everything else equal (that is, holding σ and Δ_v constant).

3 Algorithmic Market Makers

To reach the Glosten-Milgrom equilibrium, dealers are implicitly assumed to know a lot about the primitives of the market making game. In particular, they know the expected profit they can obtain by posting a particular price, given their competitors' price, i.e., $\bar{\Pi}(a_n, \bar{a}; \mu)$. This is key. For instance, if other dealers set a price strictly above the competitive price, a dealer knows that she can obtain a strictly larger expected profit by undercutting slightly her competitors rather than matching their quotes. The standard theory does not explain how dealers learn $\bar{\Pi}(a_n, \bar{a}; \mu)$ and how they would behave without this knowledge.

In this section (and the rest of the paper), we assume that quotes are set by market makers using reinforcement algorithms (Q-learning) to learn the expected payoff of posting a given price and to select the price they post in each period. We refer to such market makers as Algorithmic Market Makers (AMs). In Section 3.1, we describe Q-learning algorithms and explain, in Section 3.2, how we parameterize them for our experiments.[19] We then present the prices and spreads chosen by AMs with and without adverse selection (Section 3.3) and compare them to those in the Glosten and Milgrom benchmark.

[18] For instance, when σ increases, the adverse selection cost drops but the likelihood of a trade increases. In the theory, this first effect fully explains the evolution of prices (the quoted spread should drop). However, the algorithms used in our experiments may behave differently when the likelihood of a trade changes (and in fact they do; see Section 3.3).

[19] See also Sutton and Barto (2018) for an introductory textbook on Q-learning algorithms.

3.1 Q-Learning Algorithms

Q-learning is an iterative procedure that defines (in our context) how each AM (i) selects the price it posts in each trading round and (ii) updates its estimate of the average profit at this price given the last observed profit. Hence, it defines a particular way to play the market making game for each AM. We restrict AMs to choose their quotes in a discrete and finite action set $\mathcal{A} = \{a_1, a_2...a_M\}$, where each a_m is a possible ask price.[20] We choose this price grid so that the expected payoff of the asset, the Glosten-Milgrom price, and the monopoly price (the one maximizing $\bar{\Pi}(a, a; \mu)$ when $N = 1$) are all in the range $[a_1, a_M]$ (see below).

The Q-learning algorithm used by each AM works as follows. It consists of T finite episodes. Each episode $t \in \{1, 2, \ldots, T\}$ consists of only one trading round and realizations of the asset payoffs are independent across episodes (one can think of episodes as "trading days"). To each AM n and episode t, we associate a so-called *Q-Matrix* $\mathbf{Q}_{n,t} \in \mathbb{R}^{M\times 1}$, which is simply a column vector of size M.[21] The m^{th} entry of the matrix, denoted $q_{m,n,t}$, represents the estimate by AM n, in episode t, of the profit from playing price a_m. For each AM, we initialize $\mathbf{Q}_{n,0}$ with random values. Specifically, for each AM n and each price index m, $q_{m,n,0}$ has a uniform distribution over $[\underline{q}, \overline{q}]$ and is i.i.d across prices and AMs.

The Q-learning algorithm specifies how each AM's Q-matrix evolves over time given the prices chosen by each AM and a client's decision in a given episode. This specification relies on two parameters (common to all AMs), $\alpha \in (0, 1)$ and $\beta > 0$ and a probability $\epsilon_t := e^{-\beta t}$. Given this parametrization, we iterate the following three steps for each episode t between 1 and T:

1. Action: We first determine the behavior of each AM in episode t. For each AM n, we define $m^*_{n,t} := arg\max_m q_{m,n,t-1}$ the index associated with the highest value in matrix $\mathbf{Q}_{n,t-1}$, and denote by $a^*_{n,t} := a_{m^*_{n,t}}$ the *greedy price* of this AM, that is, the price which according to the AM's estimate is profit-maximizing. With probability $1 - \epsilon_t$, AM n takes an "exploitation" action: it plays the greedy price. With probability ϵ_t, it takes an "exploration" action: the AM draws a random integer

[20] This constraint is necessary because the algorithm must evaluate the average profit associated with each possible price.

[21] In general, the Q-matrix of an agent has S columns, each corresponding to a state realized at the beginning of each episode that can affect the average payoff obtained by the agent with a given action. If there is no such state, $S = 1$, which is the case considered here. In particular, we do not allow AMs to condition the choice of their price on their past trading history to be as close as possible to the market making game considered in Section 2.1.

$\tilde{m}_{n,t}$ between 1 and M (all values being equiprobable) and quotes $a_{n,t} = a_{\tilde{m}_{n,t}}$. Thus, $a_{\tilde{m}_{n,t}}$ is chosen randomly in $\mathcal{A}$. Whether to explore and which price to try are drawn independently across dealers. We denote $\bar{a}_t = (a_{1,t}, a_{2,t}...a_{n,t})$ the vector of prices quoted by all AMs in episode t (i.e., for the t^{th} client).

2. Feedback: We then determine the feedback each AM receives from the market. Nature draws the asset payoff $\tilde{v}_t$ and the client's liquidity shock $\tilde{L}_t$. In the adverse-selection case, we set $\tilde{w}_t^C = v_t$. In the no-adverse-selection case, $\tilde{w}_t^C$ is drawn independently from the same distribution as $\tilde{v}_t$. We compute $a_t^{\min} = \min_n\{a_{n,t}\}$ the best offer in episode t, and if $v_t^C \geq a_t^{\min}$ a trade occurs. In this case, each AM n receives a profit equal to $\pi_{n,t} = \Pi(a_{n,t}, \bar{a}_t, \tilde{v}_t^C, \tilde{v}_t)$, as given by (2). In particular, the AMs quoting $a_t^{\min}$ share the profit (or loss) from selling the asset (while others get zero).

3. Update: Finally, we determine how each AM takes the feedback into account. Following a standard version of Q-learning, each AM updates its Q-matrix as follows:

$$q_{m,n,t} = \begin{cases} \alpha\pi_{n,t} + (1-\alpha)q_{m,n,t-1} & \text{if } a_{n,t} = a_m \\ q_{m,n,t} & \text{if } a_{n,t} \neq a_m \end{cases} \tag{10}$$

In words, after playing action m the AM updates the associated value in the Q-matrix and inputs a weighted average of the observed payoff and the previous value. The values associated with other actions do not change.

To understand the intuition behind this class of algorithms, it is important to remember that they are designed to use no ex-ante knowledge of the expected profit associated with each action. The goal of the algorithm is to estimate this expected profit and eventually take the action that seems to maximize the average profit given the AM's estimate. The only way to learn is to experiment different prices, in particular by "exploring" in Step 1, receive feedback from the environment (Step 2), update the estimate (Step 3), and accumulate observations by repeating these three steps. As observations accumulate and the estimates become more precise, the algorithm can more often "exploit" and play the action that is associated with the highest estimate of the expected profit, called the "greedy" action.

This general logic is common to the entire family of reinforcement learning algorithms. Q-

learning is the simplest class of such algorithms. In particular, it uses only two parameters to control the trade-off between "exploring" and "exploiting" and how to update AMs' estimates of their expected profit at each price:

– The parameter β controls the speed at which ϵ_t decays over time, so that AMs explore a lot in early episodes and end up exploiting with a probability close to 1 in later episodes. The logic is as follows. Experimentation is potentially costly since it means that the AM posts a price which, according to its current estimate, does not yield the highest expected profit. In early episodes, it makes sense to pay this cost because early estimates are unreliable anyway, and so experimenting with a new price may uncover a more profitable action. As the number of past episodes grows, information accumulates and the learning gain in experimenting becomes smaller. Intuitively, the algorithm should therefore gradually shift from exploring to exploiting over time. This is governed by β: a larger β means that the shift to exploiting will occur faster.

– The parameter α controls the sensitivity of the AMs' estimates to new observations. The higher is α, the higher is the impact of a new realization of the profit obtained by an AM at a given price on its estimate of the expected profit at this price. Importantly, the AM's profit at a given price is random because both the asset payoff and the client's decision to trade are random (see Section 4.1). Thus, even if all AMs keep playing the same prices, a too large α leads to unstable estimates (consider the extreme case $\alpha \to 1$). If α is small, the entries of the Q-matrix are more stable but learning is slower (in the extreme case in which $\alpha \to 0$ there is no learning).

Importantly, there is no basis on which one can, a priori, choose α and β since the algorithm's designer is supposed to know nothing about the environment (this is the reason why the Q-learning algorithm is used). Thus, α and β must be seen as fixed parameters. Similarly, there are many variants of the Q-learning algorithm, with different specifications for the experimentation probability ϵ_t and the updating rule (10), and more sophisticated classes of reinforcement learning algorithms. We choose a simple Q-learning algorithm for comparability with recent literature in finance and economics, and because it features in a simple and transparent way the main properties of reinforcement learning algorithms more generally.

3.2 Experimental Design

Our experiments aim at testing whether the Glosten-Milgrom benchmark (in particular, Hypotheses H.1 to H.4) predicts well the prices set by AMs using Q-learning algorithms. As these algorithms are designed to learn by exploring for many periods, this test should be performed after many iterations ("training period"), so that the algorithms have had enough time to learn. However, there are two difficulties here.

First, two different runs of T episodes may lead to different outcomes, depending on the realizations of the different random elements in the simulation. Even with a deterministic initial Q-matrix and given prices, the payoff an AM actually receives in a given episode t is stochastic and depends on the realization of $\tilde{v}_t$ and $\tilde{L}_t$. Hence, it is possible that in a given history an AM was "lucky" with a certain price and ends up choosing this price very often, whereas in a different history the same AM was unlucky with this same price and hence plays differently. To address this issue, we run a large number K of experiments consisting of T episodes each, holding the parameters of the market making game constant, and we focus on the distribution of outcomes (e.g., the average and the standard deviation of quoted spreads) across these experiments.

Second, the Q-learning algorithms that we use do not converge to a constant action as the number of episodes T grows large (see Appendix A.5 for a formal analysis).[22] The intuition is as follows. Suppose that this is not true. That is, after some period, AMs play the same greedy-price a_m forever and, to simplify, that AMs do not experiment anymore. At this price, the likelihood that the client does not trade for the next T' episodes is always strictly positive, because $\Pr(\tilde{v}^C < a_m) > 0$ in our setting. Consequently, over the next T' episodes, the AMs' estimate of their profit at price a_m will decline. As this estimate can become arbitrarily close to zero with a positive probability for T' large enough, there is always another price that can become the greedy-price with a positive

[22] Watkins and Dayan (1992), Jaakkola *et al.* (1994), or Tsitsiklis (1994) study conditions under which Q-learning converges to the optimal action. These conditions are not met in our setup, for three reasons: (i) convergence to the optimal action requires the algorithms to experiment an infinite number of times, whereas our specification of ϵ_t leads to a finite expected number of experimentations; (ii) the updating rule needs to be such that the weight given to each additional observation goes to zero as T goes to infinity, whereas (10) always gives a constant weight α to the latest observation; (iii) the environment needs to be stationary, which is not the case in a multi-agent problem in which each agent changes its strategy over time. It is possible to change the algorithm to avoid problems (i) and (ii), at the cost of losing comparability with the recent literature using Q-learning algorithms in economics and finance. We do this in Online Appendix OA.2. We still observe a distance with the predictions of the Glosten-Milgrom benchmark, due to problem (iii).

probability - a contradiction.[23] In sum, in our setting, there is no price that can be a greedy price forever, no matter how long is the training period. To address this issue, we choose a large value of T and focus on the average value of different variables in episode T, across K experiments. We check that T is large enough that this average value no longer depends on T.[24] That is, we focus on the long run average behavior of the AMs.

After experimenting with different parameterizations to address the two issues above, we settled on the following baseline parameters. The parameters of the economic environment are the same as in Table 1: $\Delta_v = 4$, $\sigma = 5$, $v_H = 4$, $v_L = 0$, $\mu = 0.5$, and $N = 2$ (two AMs). In addition, AMs can choose all prices between 1.01 and 14.9 included on a grid with a tick size of 0.1 (139 prices in total). This specification makes sure that the zero expected profit prices are in the range of possible prices for all specifications considered in our experiments. We initialize the Q-matrices with random values following a uniform distribution between $\underline{q} = 3$ and $\overline{q} = 6$, so that all values of the initial Q-matrix are above the maximal payoff a dealer can get in a given period.[25] We run $K = 1,000$ experiments, each experiment consisting of $T = 1,000,000$ episodes. In all experiments we set $\beta = 8.10^{-5}$ and $\alpha = 0.01$. This means that the algorithm chooses to experiment 12,500 times in expectation, and hence "explores" each price around 90 times on average.[26]

For each set of parameters, in episode t of experiment k we compute the minimum ask price $a_t^{min,k}$ and the realized asset value v_t^k. We the compute the following variables:

1. **The quoted spread** QS_t^k, which is the best offer minus the expected payoff of the asset:

$$QS_t^k = a_t^{min,k} - \mathbb{E}[\tilde{v}]. \tag{11}$$

[23] In Appendix A.5, we also show formally that the values of each AM's Q-matrix cannot converge to a single point.

[24] Other papers in the literature take a different approach and wait for the algorithms to keep the same action for a large number of episodes before ending each experiment. That is, each experiment has potentially a different T. We do not follow this approach as it can in principle be misleading in a stochastic setup, see the Online Appendix OA.4. However, we observe that in most experiments the algorithms have indeed taken the same action for a large number of periods, so that this difference in approaches is likely inconsequential in practice.

[25] This specification is common in the literature on Q-learning to guarantee that all actions are chosen sufficiently often to overcome the initial values of the Q-matrix. See in particular Asker *et al.* (2023). Indeed, as long as $q_{m,n,t}$ is larger than the maximal payoff the agent can obtain, action m will necessarily be picked again because all the cells associated with actions that are played eventually fall below the maximal payoff.

[26] Each price will be played many more times due to the initialization of the Q-matrix, and in addition a price will be played with some probability when it becomes the greedy price.

2. **The realized spread** (only when there is a trade) RS_t^k, which is:

$$RS_t^k = a_t^{min,k} - v_t^k. \tag{12}$$

We then compute the average over the K experiments of these variables in the last episode ($t = T$).[27] The average quoted spreads and average realized spreads are empirical estimates of the expected quoted spread, $\overline{QS}$, and the expected realized spread, $\overline{RS}$ (defined in Section 2) and we use these estimates to test whether hypotheses H.1. to H.4. (see Section 2.2) predict AMs' long run prices.

As AMs must post their quotes on a grid, we cannot expect hypotheses H.2, H.3, and H.4 to strictly hold since the Glosten-Milgrom price might not be on the grid (and therefore AMs' realized spread cannot be exactly zero). Moreover, if the tick size is large enough and the number of dealers small enough, the market making game can have two Nash equilibria in pure strategies and one equilibrium in mixed strategy (see Appendix A.6 for more details). Thus, when we report the results from our simulations (Section 3.3), we always report the quoted and realized spreads in the least competitive pure-strategy Nash equilibrium. In any case, as the tick size in our experiments is small, the difference between the Glosten-Milgrom price and the price in the least competitive Nash equilibrium is small.

3.3 Experimental Results

We first report, in Figure 2 (Panel A), the distribution of the greedy price in the last episode in the baseline case, with $\Delta_v = 4$ and $\sigma = 5$ (in all 1,000 experiments both AMs have the same greedy price in the last episode). In this case, the Glosten-Milgrom price is $a^* = 2.68$ and is therefore not exactly on the grid of possible prices. In the least competitive Nash equilibrium, dealers post a price of 2.8 (about 1 tick above the Glosten and Milgrom price). As the figure shows, AMs' quotes vary across experiments (standard deviation of 0.73) and, in all experiments, the greedy price is above the least competitive Nash equilibrium (and therefore far above the Glosten-Milgrom price). The modal greedy price in the last episode is 4.60 and the mean is 4.97. At any price above 2.8, each AM is strictly better off undercutting its competitor since 2.8 is the least competitive Nash equilibrium. For instance, consider the case in which both AMs settle on a price of 5. At this price,

[27] The average realized spread is: $\frac{\sum_{k=1}^{K} V_T^k RS_T^k}{\sum_{k=1}^{K} V_T^k}$. That is, it is computed only when a trade occurs.

in the baseline case, each AM obtains a true expected profit of 0.30. However, each AM could obtain a greater expected profit, of 0.59, by undercutting its competitor by one tick (posting a price of 4.90). The AMs do not learn this.[28]

Panel B of Figure 2 shows the evolution over episodes 1 to T of the average greedy price (averaged over K experiments). In the first part of the learning process (roughly the $20,000$ first episodes), the average greedy price decreases and then stabilizes at 4.97. Thus, initially at least, AMs seem to learn to lower their price to attract more clients. However, as their experimentation rate decays, they have fewer opportunities to learn and their assessment of their profit at each price changes less from one client to the next. As a result, in a given experiment, the greedy price tends to stabilize. Yet, it varies across experiments (the figure shows the evolution of the greedy prices one standard deviation away from the average) because the trading history is not the same.

[INSERT FIGURE 2 ABOUT HERE]

In Panel A of Figure 3, we study the effect of the dispersion in clients' liquidity shocks σ on AMs' average quoted spread. To this end, we run $K = 1,000$ experiments for different values of σ ranging from 1 to 9 (other parameters are as in the baseline case), both in the adverse-selection case and the no-adverse-selection case ($18,000 = 2 \times 1,000 \times 9$ experiments overall). For each value of σ, we then compute and plot the average quoted spread $\overline{QS}$ in each case. We also plot the quoted spread in the Glosten-Milgrom benchmark, with and without adverse selection.

Consider the adverse-selection case first. For all values of σ, the average quoted spread in this case is largely above the quoted spread in the least competitive Nash equilibrium. Strikingly, this is also the case when there is no adverse selection. These observations confirm for a broader set of parameters that AMs settle on non-competitive prices, failing to learn that they could increase their expected profit by undercutting their competitor at these prices. Moreover, the figure shows that, both in the adverse-selection case and the no-adverse-selection case, the average quoted spread increases with σ, the dispersion of clients' liquidity shocks. This is not consistent with our hypotheses H.1 and H.2, which imply that the quoted spread should be decreasing with this dispersion in the former case and insensitive to this dispersion in the latter. Moreover, these findings cannot be

[28] Of course, by playing a price of 4.90, the AM may well eventually induce its competitor to post another price, say 4.90, at which they will both be worse off. However, nothing in the AM's design allows for this type of forward-looking reasoning (in particular, as AMs cannot condition their prices on the past trading history, they cannot learn that undercutting might generate a loss in future profits by triggering a drop in their competitor's price).

explained by the fact that AMs' quotes are constrained to be on a specific price grid, since H.1 and H.2 also hold in the least competitive Nash equilibrium (see the dotted-dashed lines on the figure).

In Panel B of Figure 3, we report the average realized spreads $\overline{RS}$ for different values of σ. The figure shows in another way that AMs do not post competitive quotes: Their average profits per trade (average realized spread) are far above zero and those in the least competitive Nash equilibrium. Interestingly, AMs learn to cope with adverse selection since their average realized spread is positive for all values of σ. Thus, their average quoted spread exceeds adverse selection costs on average, which explains why AMs' average quoted spreads are larger when there is adverse selection than when there is not. However, AMs' average realized spreads are *smaller* with adverse-selection case than without, all else equal. Thus, adverse selection induces AMs to behave *more* competitively (charge smaller markups relative to costs).

This finding is surprising. According to hypothesis H.3., realized spreads should be zero on average, whether or not there is adverse selection. Moreover, we are not aware of theories predicting that sellers (here dealers) should become more competitive when they are exposed to adverse selection. Furthermore, price discreteness cannot (at least in an obvious way) explain why the average realized spread is always *smaller* with adverse selection than without.

Last, AMs' average realized spreads increase with the dispersion of clients' liquidity shocks. Thus, AMs get larger rents, whether there is adverse selection or not, when the dispersion of clients' liquidity shocks increases. This finding is again at odds with the Glosten-Milgrom benchmark (it rejects hypothesis H.3) and is not predicted even after accounting for price discreteness. It suggests again that it becomes more difficult for AMs to learn to undercut when the dispersion of clients' liquidity shocks gets larger (and therefore adverse selection costs smaller).

[INSERT FIGURE 3 ABOUT HERE]

In Figure 4, we consider the effect of the volatility of the asset payoff, Δ_v. It show (Panel A) that the average quoted spread increases with the asset volatility in the adverse-selection case, as Hypothesis H.1 predicts. However, the average quoted spread also increases with volatility in the no-adverse-selection case. This is unexpected: H.2 predicts that the quoted spread should not depend on this volatility when there is no adverse selection.

This observation suggests that an increase in asset volatility makes AMs less competitive. This

conjecture is confirmed by Panel B of Figure 4, which shows that AMs' average profit per trade (their average realized spread) increases with the asset volatility. However, it does so less in the adverse selection case and, like we observed before, we find that AMs' average realized spread is smaller in the adverse selection case for all values of Δ_v, even though their average quoted spread is larger. Overall, Figure 4 conveys a message similar to that of Figure 3: AMs learn to cope with adverse selection and adverse selection makes them more competitive.

[INSERT FIGURE 4 ABOUT HERE]

Figure 5 shows the evolution of AMs' average quoted and realized bid-ask spreads when the number of AMs, N, increases from 2 to 10. As N increases, the average quoted and realized spreads decline. However, even with 10 AMs, they remain significantly larger than in the least competitive Nash equilibrium. It may seem intuitive that more numerous AMs makes prices more competitive. However, in the market making game considered in this paper, it takes only 2 dealers to obtain the Bertrand-Nash equilibrium. Thus, in this equilibrium, average quoted spread should remain stable when N increases from 2 to 10 (Hypothesis H.4), in contrast to what we observe experimentally.[29] Last, as when we vary σ and Δ_v, we observe that AMs' average quoted spreads are larger when there is adverse selection than when there is not, for all values of N. However, AMs' average realized spreads are smaller in the former case.

[INSERT FIGURE 5 ABOUT HERE]

In sum, AMs learn to not be adversely selected: In all environments considered in our experiments, their average realized spreads are positive and AMs charge larger quoted spreads when adverse selections costs are strictly positive than when there are nil. However, AMs' long run prices deviate in many ways from those predicted by the Nash equilibrium of the market making game:

1. AMs settle on prices well above the least competitive Nash equilibrium of the mark making game. This means that each AM could obtain a larger expected profit by undercutting its competitor. However, it fails to learn this (because of estimation errors; more on this below).

[29] Interestingly, Brogaard and Garriott (2019) find empirically that average bid-ask spreads gradually decline with entry of new high frequency market makers, for a sample of Canadian stocks. Their finding is more consistent with the pattern shown in Figure 5 than that predicted by the Glosten-Milgrom benchmark.

2. AMs' prices are more competitive (closer to costs) when there is adverse selection than when there is not.

3. AMs' prices are less competitive–whether there is adverse selection or not–when the dispersion of clients' liquidity shocks or the volatility of the asset payoff increase.

These outcomes cannot be explained by the standard analysis of the market making game. In fact, as shown in this section, most of the predictions of this analysis are rejected in our experiments. In the next section, we propose an explanation for this finding.

4 Interpretation

4.1 Noisy Learning, Competition, and Adverse Selection

It is perhaps not surprising that Q-learning algorithms do not converge to the Glosten-Milgrom benchmark: these algorithms face a stochastic environment and experiment only a finite number of times. Their estimates of the payoffs associated with different strategies are therefore noisy, which can lead them to take actions that appear suboptimal (mistakes) for an observer knowing the true expected profit at each price. While finite experimentation is an important feature of the algorithms we use, our algorithms still experiment many times, at least sufficiently to find the true best response when the price of their competitor is fixed (see Section 4.2). Moreover, in itself, lack of experimentation cannot explain (i) why the period T price is systematically above the Glosten-Milgrom price, instead of being randomly distributed around it, and (ii) why AMs post more competitive prices when their adverse selection cost increases.

Our proposed interpretation of these experimental findings is two-pronged. It relies on two features of the environment. First, in the early phase of their learning, each AM faces a non stationary environment because its competitor is frequently experimenting. It is therefore difficult for AMs to assess the average profit that can be achieved at each price. This issue becomes less acute over time because AMs experiment less and less. However, precisely because of this, they fail to learn to undercut until the competitive price is reached. Second, even holding constant the price of their competitor, each AM's profit is a noisy estimate of its expected profit at a give n price. Any parameters that makes the variance of the profit at a given price larger makes this noise larger

and AMs more prone to making mistakes. We explain these two points in more details now.

To understand the first issue, consider again the baseline case considered in Figure 2. In the first episodes, both AMs are experimenting with a high probability. AM 1 for instance is gradually learning how to best respond to AM 2. However, most of the time, AM 2 chooses a random price between 1.1 and 14.9 , since the likelihood of experimentation is high in early episodes. The best response to this random play by AM 2 is actually for AM 1 to play $a = 5.6$. As AM 1 plays prices closer to 5.6 more often (since the likelihood of experimentation declines over time), AM 2 should in principle learn that its best response is then to play prices below 5.6, in an undercutting process typical of Bertrand competition. However, because both AMs experiment less and less often over time, this undercutting process will typically not last long enough to reach the Glosten-Milgrom price. For instance, both AMs may have reached a price of only 5.0 when the probability of experimenting ever again becomes very small. If for both AMs playing prices below 5.0 did not prove profitable in the past (when the other AM was playing differently), then the AMs appear "stuck" with supra-competitive prices. In sum, AMs do not fully learn to undercut.

This reason for why the AMs' prices do not reach the Nash equilibrium is similar to what happens in Asker *et al.* (2023), Dou *et al.* (2023), or Abada *et al.* (2022). We can now push this logic further to explain how it interacts with the parameters of the model and with adverse selection, and how this explains our experimental results. As we just explained, algorithms learn to undercut each other by experimenting, but since experimentation is finite they may not have enough time to reach the Glosten-Milgrom price. An implication is that the final level of prices depends on how fast the algorithms learn. As explained previously, the Q-learning algorithm can be seen as a way to estimate the average true payoff of choosing a particular price. Intuitively, arriving at a good estimation takes more time when the AM's profit at this price is more uncertain, holding competitors' prices fixed (as in a Nash equilibrium, for instance).[30] More specifically, in the two-player example, if AM 2 is currently playing a_m above the Glosten-Milgrom price, then AM 1 will learn to undercut and play a_{m-1} if and only if observations in its Q-matrix accumulate such that $q_{m-1,1,t} > q_{m,1,t}$. Even though in expectation the profit from playing a_{m-1} is preferable to playing a_m, that is, $\mathbb{E}(\Pi(a_{m-1}, \bar{a}, \tilde{v}^C, \tilde{v})) > \mathbb{E}(\Pi(a_m, \bar{a}, \tilde{v}^C, \tilde{v}))$, AM 1 may end up having $q_{m-1,1,t} < q_{m,1,t}$

[30] In the experiments, AMs' prices vary over episodes (especially in the early episodes). For a given AM, variations in the price of its competitor is another source of variation in its profits, absent from the theory.

simply because both profits have a high variance and are hence estimated with noise. Thus, we conjecture that when the variance of dealers' profits at a given price is higher, outcomes become less competitive because it is more difficult for AMs to accurately rank the average true profits obtained at different prices and to realize that undercutting is profitable.

We now show that the results in the previous section are consistent with this conjecture. To this end, it is useful to compute the theoretical variance of an AM's profit $\text{Var}(\Pi(a_1, \bar{a}, \tilde{v}^C, \tilde{v}))$, *holding prices constant*. To provide the intuition in the simplest way, we do this only in the case with two AMs. In the no-adverse-selection case ('$n.as$') and $a_1 < a_2$, using (2) gives (see Appendix A.4 for derivations):

$$\text{Var}_{n.as}(\Pi(a_1, \bar{a}, \tilde{v}^C, \tilde{v})) = (a_1 - \mathbb{E}_{\frac{1}{2}}(v))^2 \underbrace{\mathbb{E}_{\frac{1}{2}}(V(a_1, \tilde{v}^C))(1 - \mathbb{E}_{\frac{1}{2}}(V(a_1, \tilde{v}^C)))}_{\text{Demand Risk}} + \underbrace{\frac{\Delta_v^2}{4}\mathbb{E}_{\frac{1}{2}}(V(a_1, \tilde{v}^C))}_{\text{Inventory Risk}}. \tag{13}$$

This is the variance of AM 1's profit if it undercuts its competitor without adverse selection. If instead AM 1 matches its competitor's price, the variance of its profit is as given in (13), divided by 4 because each AM fills only 50% of the client's order.

Thus, in the absence of adverse selection, *holding prices constant*, AM 1's profit is uncertain, even if it quotes the best price, for two reasons: (i) The client's demand is uncertain (the variance of this demand is $\mathbb{E}_{\frac{1}{2}}(V(a_1, \tilde{v}^C))(1-\mathbb{E}_{\frac{1}{2}}(V(a_1, \tilde{v}^C)))$ and (ii) conditional on the client trading, the value $-\tilde{v}$ of the dealer's short position is uncertain. We refer to these two sources of risk for dealers as "demand risk" and "inventory risk", respectively. When there is no adverse selection, both sources of risk increase with the dispersion of clients' liquidity shocks, σ and the volatility of the asset payoff, Δ_v (see Appendix A.4). This explains why prices become less competitive (realized spreads increase) when σ and Δ_v increase: as learning becomes noisier, AMs' ability to learn to undercut their competitor is slowed down. One way to attenuate this effect would be for AMs to experiment more when profits become more volatile. However, experimenting is costly and algorithms are supposed to be designed without prior knowledge of the environment (more on this in Section 4.3).

Now consider the case with adverse selection (as). In this case, the variance of dealer 1's profit

(when $a_1 < a_2$) is (see Appendix A.4):

$$\text{Var}_{as}(\Pi(a_1, a_2, \tilde{v}^C, \tilde{v})) = \text{Var}_{as} - \left[\frac{\Delta_v \times \Delta_D}{2}\left(a_1 - \mathbb{E}(v) - \frac{\bar{\Pi}_{as}(a_1, \bar{a}; \frac{1}{2}))}{2}\right)\right], \tag{14}$$

where $\text{Var}_{n.as}$ is given by(13) and $\bar{\Pi}_{as}$ is dealer 1's expected profit with adverse selection when she quotes $a_1 < a_2$ (equation (6) with $Z = 1$). The last term (in bracket) is positive because the dealer's quoted spread is always larger than her expected profit with adverse selection (see (6)). Thus, other things equal, adverse selection *reduces* the variance of dealers' profits and therefore the noise in AMs' learning process. The reason is that adverse selection shifts the mass from trades that generate relatively profits (those in which $v = v_L$) to trades that generate relatively low profits or even losses (those in which $v = v_H$). As a result, realized profits have a higher chance to be close or equal to zero (no trade) with adverse selection and the variance of profits declines (see Figure 7 in Appendix A.4 for an example). This "shifting effect" is captured by the term in bracket in eq.(14). As $\text{Var}_{as} < \text{Var}_{n.as}$, our conjecture implies that AMs' quotes should therefore be more competitive in environments with adverse selection, which is what we observe in our experimental results.

An increase in the dispersion of liquidity shocks, σ, raises the variance of AM 1's profit (see because (i) it increases $Var_{n.as}$ for reasons previously explained and (ii) it reduces the shifting effect previously discussed because it reduces adverse selection (see Appendix A.4 for a proof that the term in bracket in eq.(14) decreases with σ). According to our conjecture, an increase in σ should therefore also lead to less competitive prices (even though it reduces adverse selection costs) when there is adverse selection. This is also what we observe (see Figure 3).

In contrast, an increase in the volatility of the asset payoff has an ambiguous effect on the variance of AM 1's profit when there is adverse selection. Indeed, as previously explained, it increases $Var_{n.as}$ but it also increases adverse selection cost and therefore the shifting effect. Thus, the effect of the volatility of the asset payoff on the variance of AM 1's profit is weaker when there is adverse selection than when there is none. According to our conjecture, we should therefore see a smaller effect of an increase in the volatility of the asset payoff on AMs' average realized when there is adverse selection than when there is not. Again this is exactly what we observe (see 4).

In sum, when dealers are risk neutral and Bayesian (as assumed in standard analyses of the market making game, see Section 2.2), the variance of their profits plays no role. In contrast, it

seems to matter greatly when dealers use Q-learning algorithms to set their quotes, even though dealers are not penalized for taking risks in our experiments. The reason is that an increase in the variance of AMs' profits makes their estimates of the average payoffs of the various actions they can take less accurate because the feedback they receive is noisier. It is therefore more difficult for them to realize that by lowering their quoted spread, they can increase their profits by increasing trading volume. Interestingly, this effect leads AMs to require larger "risk premia" (average realized spreads) when their profits become more volatile, as if they were risk averse.

4.2 Alternative Explanations

Despite the simplicity of the Q-learning algorithm, strategic interactions between several such algorithms can lead to intriguing properties, some of which have been discussed in recent literature. We briefly discuss here some of these properties and explain why they cannot alone explain our experimental results or even do not apply in our setting.

Deficient Algorithm Design. The first and simplest explanation could be that our algorithms are simply not sophisticated enough or not well designed to play the market making game. However, as explained below, we parameterized the algorithms such that they still do a reasonable job at learning how to behave in their environment. The problem does not come from the design of each algorithm but from the interaction between these algorithms. To illustrate this point, it is useful to consider a different experiment with two AMs. In this experiment, we fix the price posted by AM 2 at 5.0 in every period, i.e., about the level of the average greedy price after T episodes in our baseline experiments (see Figure 2). We then test whether AM 1 is able to learn the best response to this price, which is 4.9. We report the results in Figure OA.1 in the Online Appendix. We find that it takes 46,043 episodes for the average greedy price over $K = 1,000$ experiments to reach 4.9. After $T = 1,000,000$ episodes the modal greedy price of AM 1 is indeed 4.9. There is only one experiment with a final greedy price above 4.9 (hence AM 1 has not learnt to undercut AM 2), and a few where the average greedy price is 4.82.

These findings show that the Q-learning algorithm performs well against an AM with a fixed pricing strategy. In particular, in this case, the other AM can learn to undercut with the parametrization used for the Q-learning algorithms in our environment. The reason why it does not in our experiments is that the other AM's price is not fixed over time and that by the time AM 2 reaches a price

of 5, AM 1 experiments only with a very low probability (as is the case for AM 2). This illustrates the importance of experimentation (and lack thereof) in our explanation.[31]

Coupling. A second explanation relies on the idea of "coupling" (Banchio and Skrzypacz (2022)): Even though the algorithms experiment at random times and independently of each other, they may end up playing in a correlated manner. For instance if AM 1 and AM 2 play the same price a_m and AM 1 learns to undercut to a_{m-1}, then AM 2 will eventually learn to play a_{m-1} or a_{m-2}, which will reduce the value associated with a_{m-1} in AM 1's Q-matrix. AM 1 will then eventually revert to a_m, and so will AM 2, giving rise to cycles of alternatively high and low prices (so that the average greedy price is above the competitive price). Banchio and Mantegazza (2022) illustrate this logic in a prisoner's dilemma and a Bertrand game with two prices and no uncertainty. It is unclear whether such outcomes can occur in an environment like ours, with many prices and stochastic payoffs. Intuitively, in this case, spontaneous synchronization of different algorithms seems more difficult. In any case, coupling requires the algorithms to experiment for an infinite number of episodes, which is not the case in our specification.

To explore this point in more details, we conduct the same experiment as the baseline case (Figure 2), but we let AMs explore with a positive probability forever, as in Banchio and Mantegazza (2022). Moreover, we change the updating rule (10) so that the Q-matrix simply records the empirical average of payoffs obtained with each strategy. This parameterization of the algorithm satisfies the assumptions of Watkins and Dayan (1992) ensuring convergence in a stationary environment with only one algorithm (see Footnote 22). Despite having two algorithms and hence a non-stationary environment, we observe in Figure OA.2 (Online Appendix) that the average greedy prices converge to 3.36, with the mode of the distribution at 3.0, only slightly above the least competitive Nash equilibrium of 2.8. We interpret this finding as confirmation that the distance between our experimental results and the Glosten-Milgrom benchmark is more likely to come from insufficient experimentation than from coupling.

Tacit Collusion. A third possibility is that AMs learn how to play a collusive equilibrium sustained by dynamic punishment strategies, as found in Calvano *et al.* (2020) and subsequent

[31] We repeated the same experiment with lower fixed prices. When the price becomes very close to the Glosten-Milgrom price, for instance 3.0 in the baseline setting, AM 1 fails to learn to play the best response of 2.9. The reason is that the profit from playing 2.9 or 3.0 has a very low expectation and a high variance, and the algorithm fails to detect that these strategies are more profitable than playing above 3.0 and getting 0.

papers (e.g., Dou *et al.* (2023)). However, this explanation cannot hold in our case for two reasons. First, in the market making game considered in our experiments, there are no non-competitive Nash equilibria (the game is static). Thus, AMs cannot settle on a non competitive Nash equilibrium. Second, while algorithms play the market making game many times with different clients in our experiments, they cannot condition their action on the trading history (in particular past prices and trade outcomes in the previous episode), unlike in Calvano *et al.* (2020). They cannot therefore learn and execute strategies similar to punishment strategies in repeated games. The crucial point here is the absence of any observable state on which the algorithm can condition, not the fact that the algorithm maximizes the one-shot profit of the game instead of the discounted value of future profits.[32]

4.3 Are Q-learning Algorithms Suboptimal?

As AMs' long run prices do not form a Nash equilibrium, they are - by definition - reacting suboptimally to their competitors' prices. As with any non-Nash outcome, a natural question is why are players "leaving money on the table"? That is, why wouldn't agents designing pricing algorithms adapt them so that they eventually learn to lower their prices when it is profitable to do so?

We believe that the answer most consistent with Q-learning is the uncertainty faced by the agents: They are assumed to use Q-learning algorithms precisely because they neither know the specifics of the game they are playing, nor the behavior of their competitors. As a result, they do not have the information necessary to realize that their behavior is suboptimal. If they did, they would probably not use such algorithms in the first place. Moreover, in the stochastic environment of our experiments, it is very difficult to empirically estimate expected profits and realize that better strategies are available.[33] As explained in Section 4.1, this is actually the reason why AMs reach an outcome far from the Glosten-Milgrom benchmark. In sum, the suboptimal nature of the

[32]The updating rule (10) is adequate for what Sutton and Barto (2018) call an "episodic task": an optimization problem with a clear beginning and end, here a one-period game. Other papers in the literature typically use the rule $q_{m,n,t} = \alpha[\pi_{n,t} + \gamma \max_{m'} q_{m',n,t}] + (1-\alpha)q_{m,n,t-1}$, which is meant for computing the value of an action in an infinite horizon problem like an infinitely repeated game. We can of course implement such a rule in our experiments and ask each dealer to maximize the discounted value of all future episodes. As long as the algorithm cannot condition on past history this makes no difference. However, because of the new term $\gamma \max_{m'} q_{m',n,t}$ there is less update after each episode, this makes learning slower and the final greedy price higher, keeping all other parameters constant.

[33]In deterministic contexts (e.g., a Bertrand game with deterministic demand) if the agents converge to a non-Nash outcome, experimenting a deviation once is enough to realize the outcome is individually suboptimal. This is not the case in our market making game (e.g., in our baseline case, undercutting a price of 5 by one tick can yield a profit of zero (no trade) even though its average payopff is strictly larger than the average profit at 5.

behavior of the algorithms is apparent only to the modeler, who has information that algorithms' designers not have. Accordingly and for the same reasons, the applied literature on Q-learning rarely discusses optimality, and usually focuses on reaching satisficing outcomes or "improving" over current methods in some baseline example.

Another approach, followed for instance by Compte (2023), is to assume that agents are constrained to use Q-learning algorithms, but that they can optimize the parameters of these algorithms (α and β in our setting). However, if the agents have no information about the environment, the only possibility for them to optimize the parameters of their algorithm is to try different ones and observe the outcome. This is basically a new Q-learning problem, and now the question is how to parameterize the algorithm to solve this new problem. Alternatively, the agents could be assumed to be able to compute the expected payoff associated with each parameterization of the algorithm and play a Nash equilibrium over the parameters. However, in our context, it is unclear why agents endowed with such detailed information about the environment would stick to using Q-learning algorithms.

Still, for robustness, we followed this line of reasoning and checked whether agents would change their experimentation rate (β) in order to increase their total profit. More specifically, we rerun the experiments in our baseline experiment ($\sigma = 5$, $\Delta_v = 4$), for different values of AM 1's β, holding AM 2's beta fixed at $\beta = 8.10^{-5}$ (its level in our experiments). We then compute AM 1's average total profits over various time windows and report the results in Figure OA.3 in the Online Appendix. We find that AM 1's average profit is *lower* if it unilaterally uses a lower β than AM 2 (and therefore experiments more). A contrario, increasing β (experimenting less) leads to slightly higher average profits than those obtained when $\beta = 8.10^{-5}$, though not significantly so. This suggests that no AM has an incentive to unilaterally deviate from $\beta = 8.10^{-5}$.

More generally, there are two reasons why agents have no incentive to experiment more: (i) experimentation involves playing random actions instead of actions that have proven more profitable in the past, which is costly on average ; (ii) experimenting more affects the update process of the competitors, in a direction which seems to lead to lower prices on average, and hence to lower profits for both agents.

5 Learning from Order Flow and Price Discovery

When there is asymmetric information, the "order flow" (the sequence of clients' trading decisions) is a signal about the asset payoff. In models of market making with bayesian learning (for instance, Glosten and Milgrom (1985), Kyle (1985), or Easley and O'Hara (1992)), market makers use this signal to update their forecast of the asset payoff and gradually learn it ("price discovery"). This implies that the order flow impacts prices. For instance, after a buy order, dealers revise upward their estimate of the asset payoff and therefore their quotes for the next trade. This implication is important. It is the cornerstone of the specification and interpretation of the so called "price impact regressions" (Glosten and Harris (1988)) that empiricists use to analyze the effect of trades on prices in securities markets.

In this section, we study how AMs' quotes react to the order flow and compare this reaction to that predicted by the Nash equilibrium of the market making game. For this analysis, we extend the market making game considered in Section 2.1 to two periods, focusing on the adverse-selection case. That is, we assume that before the asset payoff is revealed, dealers receive orders from two different buyers who arrive sequentially in periods $\tau = 1$ and $\tau = 2$. The valuation of the buyer in period τ is $\tilde{v}^C_\tau = \tilde{v} + \tilde{L}_\tau$, where $\tilde{L}_1$ and $\tilde{L}_2$ are independent and normally distributed with mean zero and variance σ^2. In this way, we can study how AMs' quotes for the second client depends on the order flow in the first period.

We proceed as in the case with one client. That is, in Section 5.1, we first derive the Nash equilibrium (Glosten and Milgrom prices) of the market making game with two periods and derive its implication for the dynamics of quotes. Then, in Section 5.2, we run experiments in which the two periods market making game is repeated with a large number of clients and played by AMs using Q-learning algorithms and test whether AMs revise their quotes as predicted by the Nash equilibrium.

5.1 Two-Period Glosten-Milgrom Benchmark

Let denote market makers' belief about the likelihood that $v = v_H$ prior to the arrival of the τ^{th} client by μ_τ. Thus, $\mu_1 = \mu$. At the end of the first trading round, there are two possible trading histories (H_1): (i) a trade at price $a_1^{\min}$ ($H_1 = \{1, a_1^{\min}\}$) or (ii) no trade at price $a_1^{\min}$

$(H_1 = \{0, a_1^{\min}\})$. The market makers' Bayesian beliefs about the likelihood that $v = v_H$ after both histories are:

$$\mu_2(1, a_1^{\min}) := \Pr(v = v_H \mid H_1 = \{1, a_1^{\min}\}) = \frac{D(a_1^{\min}, v_H)\mu_1}{\mathbb{E}_{\mu_1}(V(a_1^{\min}, \tilde{v}_1^C))} \tag{15}$$

$$\mu_2(0, a_1^{\min}) := \Pr(v = v_H \mid H_1 = \{0, a_1^{\min}\}) = \frac{(1 - D(a_1^{\min}, v_H))\mu_1}{1 - \mathbb{E}_{\mu_1}(V(a_1^{\min}, \tilde{v}_1^C))}. \tag{16}$$

It is easily checked that $\mu_2(1, a_1^{\min}) > \mu_1 > \mu_2(0, a_1^{\min})$ if (and only if) $\Delta_v > 0$. That is, Bayesian market makers revise their estimate of the expected payoff of the asset upwards after a buy in period 1 and downwards after no trade.

Conditionally on μ_τ, one can derive dealers' expected profits in periods $\tau = 1$ and $\tau = 2$ exactly as in the one-period case. Hence, dealer n's expected profit in period τ is $\bar{\Pi}(a_n, \bar{a}; \mu_\tau)$. As a result, the Glosten and Milgrom price in periods τ, a_τ^* is given by (9) with $\mu = \mu_\tau$ in period τ. The unique Nash equilibrium of the two period market making game is such that, in each period, at least two AMs post a_τ^*.[34] Thus, hypotheses H.1 to H.4 still hold within each period τ.

However, in this case, the Nash equilibrium of the market making game has another implication. Indeed, the Glosten and Milgrom price in the second period depends on the trading outcome (order flow) in the first period. As $\mu_2(1, a_1^{\min}) > \mu_1 > \mu_2(0, a_1^{\min})$, market makers charge a higher price to the second client than to the first client if a trade takes place in the first period, and a lower price otherwise (see Table 2 for a numerical example). This yield the following hypothesis:

H.5: Market makers' offer to the second client is higher (resp., smaller) than for the first client after a buy (resp., no trade).

In the next section, we study whether AMs' quotes satisfy this standard property even though they are not Bayesian.

[INSERT TABLE 2 ABOUT HERE]

5.2 Experimental Results

As in Section 3, we now conduct experiments in which Q-learning algorithms play the two-periods market making game over T episodes. The key difference is that each episode feature two clients

[34] There is no equilibria in which market makers post non competitive prices sustained via dynamic punishment strategies because the market making game has a finite horizon (two periods).

arriving sequentially with the same common value, $\tilde{v}$. To allow the algorithms to react to the occurrence of a trade in period 1, we let them keep track in each episode of the "state" they are in, and let them play an action that depends on the state. Q-learning algorithms were initially designed to solve dynamic stochastic optimization problems (both finite and infinite horizon), and are thus in principle well suited to optimizing prices in this environment.[35] In this section we sketch how we program the algorithms in the 2-player case. A more precise and general treatment is given in Appendix A.7.

For each AM $n \in \{1, 2\}$ and episode t, we denote $s_{n,t} \in \{\emptyset, NT, 0, \frac{1}{2}, 1\}$ the state the algorithm finds itself in. The states are defined as follows: (i) $s_n = \emptyset$ in the first period; (ii) $s_n = NT$ in the second period if "No Trade" took place in the first; (iii) $s_n = 0$ in the second period if there was a trade in the first period, but AM n did not trade; (iv) $s_n = \frac{1}{2}$ in the second period if there was a trade in the first period, and both AMs shared the market; (v) $s_n = 1$ in the second period if there was a trade in the first period, and AM n sold one share.

This partition of the state space implies that each algorithm keeps track both of (i) whether a trade took place (which is important to analyze the effect of order flow) and (ii) of its inventory after period 1 (e.g., $s_n = \frac{1}{2}$ indicates a short position of $-\frac{1}{2}$ for AM n). The latter is important: As $\tilde{v}$ is realized only at the end of the second period, the algorithm cannot know how profitable the first-period trade was before the end of the second period. Hence, the algorithm needs to keep track of its inventory, and learn what is the value of being in a state with a short position vs. a state with a zero inventory.[36] To do this, each AM relies on a Q-matrix $\mathbf{Q}_{n,t} \in \mathbb{R}^{M \times 5}$, in which each line corresponds to a different price and each column to a state, ordered as in the previous paragraph. We denote $q_{m,s,n,t}$ the (m, s) entry of matrix $\mathbf{Q}_{n,t}$.

We then extend the process described in Section 3.1 to this case with two periods and 5 states. The logic is exactly the same as in the first-period case (details can be found in Appendix A.7). There are two important differences worth mentioning. First, the updating of the Q-matrix is different from (10) and is now:

[35] See Leach and Madhavan (1993) for the analysis of a monopolist's optimal behavior in the two-period market making game.

[36] Using inventory levels as the state variable is common in other applications of Q-learning, in particular in dynamic pricing and revenue management. See, e.g., Rana and Oliveira (2014) for an example. The list of states used by the algorithms is an important parameter of the model. The list could be even richer (e.g., conditioning on prices in period 1 as well), or coarser (not distinguishing states NT and 0).

$$q_{m,\emptyset,n,t} = \begin{cases} \alpha[a^1_{n,t} I^1_{n,t} + \max_{m'} q_{m',s_{n,t},n,t-1}] + (1-\alpha) q_{m,\emptyset,n,t-1} & \text{if } a^1_{n,t} = a_m \\ q_{m,\emptyset,n,t-1} & \text{if } a_{n,t} \neq a_m, \end{cases} \tag{17}$$

$$\text{for } s_{n,t} \neq \emptyset, q_{m,s_{n,t},n,t} = \begin{cases} \alpha[a^2_{n,t} I^2_{n,t} - \tilde{v}_t (I^1_{n,t} + I^2_{n,t})] + (1-\alpha) q_{m,s_{n,t},n,t-1} & \text{if } a^2_{n,t} = a_m \\ q_{m,s_{n,t},n,t-1} & \text{if } a^2_{n,t} \neq a_m \end{cases} \tag{18}$$

This updating rule is best understood backwards. (18) is the updating done in period 2 when the state is $s_{n,t}$. At the end of period 2, we know the quantities $I^1_{n,t}$ and $I^2_{n,t}$ sold by AM n in periods 1 and 2, respectively. We count the revenues $a^2_{n,t} I^2_{n,t}$ generated by the period-2 sale, and subtract the cost $\tilde{v}_t(I^1_{n,t} + I^2_{n,t})$ of having sold $I^1_{n,t} + I^2_{n,t}$ units worth $\tilde{v}_t$ each. (17) is the updating done in period 1. The reward recorded by the algorithm has two components. First, the revenues $a^1_{n,t} I^1_{n,t}$ from selling $I^1_{n,t}$ units. As already mentioned, in period 1 the value of $\tilde{v}_t$ is still unknown and cannot be deducted from the revenues, this will be done at the end of period 2 only. To keep track of this cost, and following the standard specification of Q-learning, we add the term $\max_{m'} q_{m',s_{n,t},n,t-1}$: this term is the value associated with moving to state $s_{n,t}$ in period 2, which as we just saw incorporates the cost of selling the asset. For instance, if AM n sells one unit in period 1 we have $I^1_{n,t} = 1$ and revenues of $a^1_{n,t} \times 1$ are recorded in the first column of the Q-matrix. In addition, AM n will start period 2 in state $s_{n,t} = 1$, and the expected value of this state is $\max_{m'} q_{m',1,n,t-1}$, that is, the maximum of the 5th column of the Q-matrix. This value takes into account that in this state AM n starts with an inventory of 1, which will have a cost of $\tilde{v}_t$.

The second important difference with Section 3.1 is that the AMs can now play a different price in each state and there are potentially 5 different greedy prices for each AM. Our experiments focus again on the last episode T. As we are interested in testing whether AMs learn to react to the order flow, we will aggregate states $s_{1,T} \in \{0, \frac{1}{2}, 1\}$ as a state with a trade in period 1, compared to $s_{1,T} = NT$ which is a state with no trade. Formally, based on the realization of K experiments of T episodes each, we denote $a^{min,k}_\tau$ the best ask submitted and V^k_τ the realized volume in period

$\tau \in \{1, 2\}$ of episode T and experiment k. We then define:

$$\bar{V}_2 = \sum_{k=1}^{K} V_2^k \tag{19}$$

$$\bar{a}_1 = \frac{\sum_{k=1}^{K} a_1^{min,k}}{K} \tag{20}$$

$$\bar{a}_2^T = \frac{\sum_{k=1}^{K} a_2^{min,k} V_2^k}{V_2} \tag{21}$$

$$\bar{a}_2^{NT} = \frac{\sum_{k=1}^{K} a_2^{min,k} (1 - V_2^k)}{K - V_2}. \tag{22}$$

Thus, $\bar{a}_1$ is the average best quote in period 1 across the K experiments, $\bar{a}_2^T$ is the average best quote in period 2 conditionally on a trade occurring in period 1 (irrespective of who traded), and $\bar{a}_2^{NT}$ in the average best quote in period 2 conditionally on no trade occurring in period 1. Our hypothesis H.5 predicts that $\bar{a}_2^{NT} < \bar{a}_1 < \bar{a}_2^T$. To test this prediction, we run $K = 1,000$ simulations of the two-period game (always with adverse selection), for nine different values of σ. Figure 6 plots $\bar{a}_1$, $\bar{a}_2^T$, and $\bar{a}_2^{NT}$ for each σ, as well as the Glosten-Milgrom prices in each period (dashed lines).

[INSERT FIGURE 6 ABOUT HERE]

Figure 6 shows that, as predicted by H.5, AMs charge a larger quote to the second client after a buy from the first and a smaller quote to the second client after no trade from the first one. However, even though H.5 is qualitatively satisfied, there are important differences between the second period prices and those predicted by the Nash equilibrium. First, after a buy, AMs in the second period raise their quotes much more than what Bayesian behavior would imply. For instance, when $\sigma = 5$, the Glosten and Milgrom price should increase from 2.8 to 3.4 after a buy. Instead, in the experiments, the average price increases from 4.80 to 5.80. Moreover, the Nash equilibrium predicts that this revision should become smaller when the dispersion of clients' liquidity shocks increases while it becomes larger in the experiments. Conversely, when no trade occurs in the first period, the second-period price is almost equal to the first-period price while it is significantly smaller in the Nash equilibrium (compare the green and the blue dashed lines in Figure 6). Overall, these patters imply that AMs' extract even larger rents from the second period client than the first and these rents increase with the dispersion of liquidity shocks, as in the one period case.

Noisy learning is again important to understand the distance between the experimental results

and the Glosten-Milgrom benchmark. There are two effects. The first effect is that the variance of $\tilde{v}$ in period 2 conditional on the outcome of period 1 is lower than the unconditional variance in our setting.[37] This effect should reduce AMs' rents in the second period and therefore make their prices closer to the Glosten-Milgrom prices than in the first period. However, there is a countervailing effect, which seems to dominate in our experiments: the algorithms have fewer opportunities to learn about the average profits of their actions for the second client than for the first client. Indeed, they learn state by state. Now they face the first state in every episode while they face only one of the other states per episode. This means that they have far fewer observations to learn about their payoffs in these states than for the first state. At any price in the first period, the probability of a buy is strictly less than 50% and decreases quickly with the price (e.g., at $a_1 = 4.80$, the probability of a trade in the first period is only 30%). This means that there are relatively few episodes in which AMs get the opportunity to get feedback about the average profit they can obtain at a given price in the second period after a buy in the first period. Intuitively, this makes it more difficult for AMs to learn the average profit they can obtain at a given price in given state in the second period, which, as explained in the baseline case, makes it more difficult for them to learn to undercut (especially after a buy order). We believe that this explains why AMs' rents seem so large in the second period after a buy in the first period.

Interestingly, these experimental results give insights into how competition between AMs can be spotted in the data. They imply that quotes will tend to over-react to order flow (here a buy). This means that the change in prices following a buy or a sell should partially revert. Such patterns have been found for long in existing empirical studies and are usually attributed to order processing costs or inventory holding costs for market makers. Our experiments suggest that they could become more prevalent as quotes are posted by algorithms, reflecting algorithms' imperfect learning of the benefits of undercutting. More generally, spreads should tend to widen after histories that are more rarely observed, or even simply over time, irrespective of whether adverse selection is actually higher after these histories.[38]

[37] Indeed, the conditional variance of the asset payoff in period τ is $Var_\tau(\tilde{v}) = \mu_\tau(1-\mu_\tau)\Delta_v^2$. As $\mu_1 = 0.5$ in our experiments, we have $Var_2(\tilde{v}) < Var_1(\tilde{v})$.

[38] This type of behavior might lead to sudden evaporation of liquidity after events that have been rarely encountered by algorithms and potentially explain flash crashes.

6 Conclusion

We study the prices posted by market makers using Q-learning algorithms in a standard market making game with adverse selection (similar to Glosten and Milgrom (1985)) and compare them to those predicted by the Nash equilibrium of the market making game. We find that, despite their simplicity and the challenge of an environment with adverse selection, our algorithmic market makers (AMs) behave in a realistic way: their quoted spreads reflect adverse selection costs and they update their quotes in response to the observed order flow. However, they also deviate from the Nash equilibrium prices in many important ways. In particular, their quoted spread are larger than the competitive spreads and their rents increase when adverse selection costs decrease. Moreover, they over-react to the order flow.

We argue that these findings stem from the fact that AMs receive a noisy feedback about the average profit of their actions (because of uncertainty in their client's demand and the asset payoff) and this noise is larger when adverse selection is less intense. In response, AMs should experiment more in noisier environments. However, our experiments suggest that this would require very long training periods and that it may not even be optimal for agents designing AMs to do so (because experimentation is costly).

Overall, our results suggest that securities markets are a quite specific and particularly interesting application of recent research on competition between pricing algorithms. In particular, they raise the possibility that these algorithms may not lead to more competitive outcomes in assets that are risky but less exposed to adverse selection. They also suggest that these algorithms could be significantly less competitive when facing states that they rarely encounter (which may explain why variations in liquidity have become more extreme with the rise of algorithmic pricing). Future research could consider the robustness of our conclusions when more complex algorithms are used or when they are used in conjunction with some prior "model of the world".

References

Abada, I., Lambin, X. and Tchakarov, N. (2022). *Collusion by Mistake: Does Algorithmic Sophistication Drive Supra-Competitive Profits?* Working paper. 7, 24

Asker, J., Fershtman, C. and Pakes, A. (2023). The impact of artificial intelligence design on pricing. *Journal of Economics & Management Strategy*, **forthcoming**, 1–29. 7, 18, 24

Baldauf, M. and Mollner, J. (2020). High-frequency trading and market performance. *The Journal of Finance*, **75** (3), 1495–1526. 8

Banchio, M. and Mantegazza, G. (2022). *Adaptive Algorithms and Collusion via Coupling*. Working paper. 7, 28

— and Skrzypacz, A. (2022). Artificial intelligence and auction design. *Available at SSRN 4033000 9*. 7, 28

Biais, B., Foucault, T. and Moinas, S. (2015). Equilibrium fast trading. *Journal of Financial Economics*, **116** (2), 292–313. 8

Brain, D., De Pooter, M., Dobrev, D., Fleming, M., Johansson, P., Jones, C., Keane, F., Puglia, M., Reiderman, L., Rodrigues, T. and Or, S. (2018). Unlocking the Treasury Market through TRACE. *FED Notes*. 1

Brogaard, J. and Garriott, C. (2019). High-frequency trading competition. *Journal of Financial and Quantitative Analysis*, **54** (4), 1469–1497. 4, 22

—, Hendershott, T. and Riordan, R. (2014). High-Frequency Trading and Price Discovery. *The Review of Financial Studies*, **27** (8), 2267–2306. 1

Buchak, G., Matvos, G., Piskorski, T. and Seru, A. (2019). *Why is Intermediating Houses so Difficult? Evidence from iBuyers*. Tech. rep., NBER, Working Paper 28252. 1

Budish, E., Cramton, P. and Shim, J. (2015). The High-Frequency Trading Arms Race: Frequent Batch Auctions as a Market Design Response. *The Quarterly Journal of Economics*, **130** (4), 1547–1621. 8

Calvano, E., Calzolari, G., Denicolo, V. and Pastorello, S. (2020). Artificial intelligence, algorithmic pricing, and collusion. *American Economic Review*, **110** (10), 3267–97. 7, 28, 29

Cartea, A., Chang, P., Mroczka, M. and Oomen, R. (2022a). Ai-driven liquidity provision in otc financial markets. *Quantitative Finance*, **22** (12), 2171–2204. 8

Cartea, Á., Chang, P. and Penalva, J. (2022b). *Algorithmic Collusion in Electronic Markets: The Impact of Tick Size*. Working paper. 7, 8

Chaboud, A., Dao, A. and Vega, C. (2019). *What makes HFTs tick? Tick size changes and information advantage in a market with fast and slow traders*. Tech. rep., Available at SSRN: https://ssrn.com/abstract=3407970. 1

Competition Market Authority (2018). Pricing algorithms. pp. 3–62. 7

Compte, O. (2023). *Q-based Equilibria*. Working paper. 30

Cont, R. and Xiong, W. (2023). Dynamics of market making algorithms in dealer markets: Learning and tacit collusion. *Mathematical Finance*, **forthcoming**. 7

Dou, W., Goldstein, I. and Ji, Y. (2023). *AI-Powered Trading, Algorithmic Collusion, and Price Efficiency*. Tech. rep., Available at SSRN: https://ssrn.com/abstract=4452704. 7, 8, 24, 29

Easley, D. and O'Hara, M. (1992). Time and the process of security price adjustment. *The Journal of Finance*, **47** (2), 577–605. 31

Foucault, T., Pagano, M. and Röell, A. (2013). *Market Liquidity: Theory, Evidence, and Policy*. Oxford: Oxford University Press. 11

Glosten, L. R. and Harris, L. E. (1988). Estimating the components of the bid/ask spread. *Journal of financial Economics*, **21** (1), 123–142. 31

— and Milgrom, P. R. (1985). Bid, ask and transaction prices in a specialist market with heterogeneously informed traders. *Journal of Financial Economics*, **14** (1), 71–100. 1, 5, 8, 11, 31, 37

Goldstein, I., Spatt, C. S. and Ye, M. (2021). Big Data in Finance. *The Review of Financial Studies*, **34** (7), 3213–3225. 1

Guéant, O. and Manziuk, I. (2019). Deep reinforcement learning for market making in corporate bonds: Beating the curse of dimensionality. *Applied Mathematical Finance*, **26** (5), 387–452. 7

Hansen, K. T., Misra, K. and Pai, M. M. (2021). Frontiers: Algorithmic collusion: Supra-competitive prices via independent algorithms. *Marketing Science*, **40** (1), 1–12. 7

Hendershott, T., Jones, C. M. and Menkveld, A. J. (2011). Does algorithmic trading improve liquidity? *The Journal of Finance*, **66** (1), 1–33. 4

Jaakkola, T., Jordan, M. I. and Singh, S. P. (1994). On the convergence of stochastic iterative dynamic programming algorithms. *Neural Computation*, **6** (6), 1185–1201. 17

Kyle, A. S. (1985). Continuous auctions and insider trading. *Econometrica*, **53** (6), 1315–1335. 1, 5, 7, 8, 11, 31

Leach, J. C. and Madhavan, A. (1993). Price experimentation and security market structure. *Review of Financial Studies*, **6** (2), 375–404. 33

MacKay, A. and Weinstein, S. (2022). *Dynamic Pricing Algorithms, Consumer Harm, and Regulatory Response*. Working paper. 7

Menkveld, A. and Zoican, M. (2017). Need for speed? exchange latency and liquidity. *Review of Financial Studies*, **30** (4), 1188–1228. 8

OECD (2017). Algorithms and collusion: Competition policy in the digital age. pp. 1–72. 7

O'Hara, M. (2015). High frequency market microstructure. *Journal of Financial Economics*, **116** (2), 257–270. 8

Pouget, S. (2007). Adaptive traders and the design of financial markets. *The Journal of Finance*, **62** (6), 2835–2863. 8

Rana, R. and Oliveira, F. S. (2014). Real-time dynamic pricing in a non-stationary environment using model-free reinforcement learning. *Omega*, **47**, 116–126. 33

Sutton, R. and Barto, A. (2018). *Reinforcement Learning: An Introduction*. Cambridge (Mass.): MIT Press. 13, 29

Tsitsiklis, J. (1994). Asynchronous stochastic approximation and q-learning. *Machine Learning*, **16**, 185–202. 17

Waltman, L. and Kaymak, U. (2008). Q-learning agents in a cournot oligopoly model. *Journal of Economic Dynamics and Control*, **32** (10), 3275–3293. 7

Watkins, C. and Dayan, P. (1992). Q-learning. *Machine Learning*, **8**, 279–292. 17, 28

Wilk, E. (2022). *Pricing Under Pressure: The Effect of Signal Corruption on the Gameplay of Pricing Algorithms*. Working paper. 7

Wunder, M., Littman, M. L. and Babes, M. (2010). Classes of multiagent q-learning dynamics with epsilon-greedy exploration. In *ICML*, pp. 1167–1174. 7

A Appendix

A.1 Tables

σ	0.5	1	3	5	7
Quoted Spread	2.00	2.00	1.24	0.68	0.47
Likelihood Trade	25%	25%	37%	45%	47%
Adverse Selection Cost	0.5	0.49	0.45	0.3	0.22
Realized Spread	0	0	0	0	0
Δ_v	0	2	4	6	8
Quoted Spread	0	0.16	0.68	1.65	3.02
Likelihood Trade	50%	48%	45%	39%	32%
Adverse Selection Cost	0	0.07	0.3	0.64	0.99
Realized Spread	0	0	0	0	0

Table 1: **Benchmark**. Clients' liquidity shocks ($\tilde{L}$) are normally distributed with mean zero and variance σ^2. Moreover, $\mathrm{E}_{\mu}(v) = 2$ and $\mu = \frac{1}{2}$ ($v_H = 4$ and $v_L = 0$). The likelihood of a trade is $\mathrm{E}_{\frac{1}{2}}(D(a^{\min}, \tilde{v}))$ (see text) and the adverse selection cost is equal to $0.5\Delta_v \times \Delta_D$. **Panel A:** $\Delta_v = 4$. Quotes have been rounded up to two decimals (which explains why they are equal when $\sigma = 0.5$ and $\sigma = 1$). **Panel B:** $\sigma = 5$.

Panel A					
σ	0.5	1	3	5	7
a_1^c	4.00	4.00	3.24	2.68	2.47
a_2^{*T}	4	4	3.82	3.26	2.93
a_2^{NT}	4	4	2.44	2.08	2.02
Panel B					
Δ_v	0	2	4	6	8
a_1^c	2	2.16	2.68	3.65	5.03
a_2^{*T}	2	2.5	3.26	4.6	5.86
a_2^{*NT}	2	1.8	2.08	205	3.66

Table 2: **Learning from Order Flow.** Clients' liquidity shocks ($\tilde{L}$) are normally distributed with mean zero and variance σ^2. Moreover, $\mathbb{E}(v) = 2$ and $\mu = \frac{1}{2}$ ($v_H = 4$ and $v_L = 0$). The table shows the Glosten-Mligrom prices at date 1 (a_1^*) and at date 2 after (i) a trade at date 1 (a_2^{*T}) or (ii) no trade at date 1 (a_2^{*NT}). **Panel A:** $\Delta_v = 4$. Quotes have been rounded up to two decimals (which explains why they are equal when $\sigma = 0.5$ and $\sigma = 1$). **Panel B:** $\sigma = 5$.

A.2 Figures

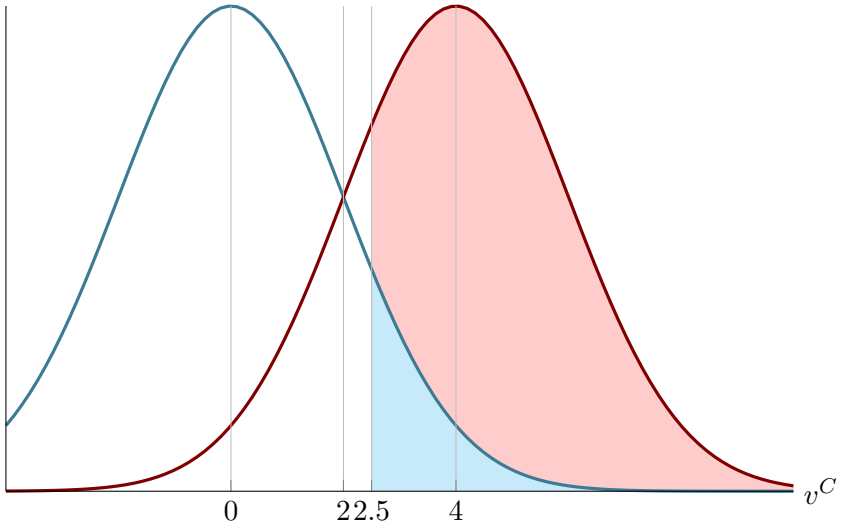
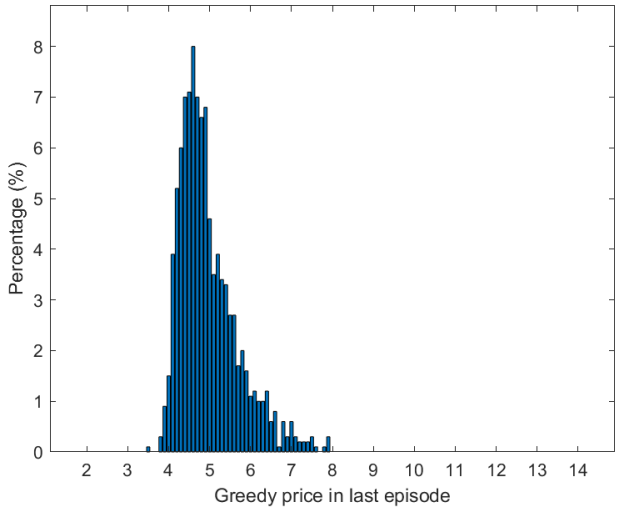


Figure 1: Distribution of Clients' Valuation conditional on $w^C = 0$ (blue) ($\sigma = 2$). If the best offer price is 2.5, the likelihood of a trade is the blue plus the red area when $w^C = 4$ and the blue area only when $w^C = 0$. When $\tilde{w}^C = \tilde{v}$ (adverse selection case), a client is therefore more likely to buy the asset when its payoff is high than when its payoff is low. The difference between the likelihood of a buy when $v = v_H$ and $v = v_L$ (denoted Δ_D in the text) is then equal to the red area. When $\tilde{w}^C$ and $\tilde{v}$ are i.i.d, the likelihood of a trade is equal to the blue plus the red area weighted by the likelihood that $w^C = 4$, independently of the asset payoff. See the text for more explanations.

Panel A: Distribution of the greedy price of AM 1 in the last episode.

This panel shows a histogram of the greedy price of AM 1 in episode T: For each possible price a between 1.10 and 14.90 the bar indicates the percentage of the 1,000 experiments conducted in which $a^*_{1,T} = a$.



Panel B: Dynamics of the average greedy price of AM 1 for episodes 1 to T.

This graph shows for each episode t the average of AM 1's greedy price $a^*_{1,t}$ across the 1,000 experiments conducted. As a measure of dispersion, we also compute the standard deviation of $a^*_{1,t}$ across experiments and plot the average of $a^*_{1,t}$ plus/minus one standard deviation (with a 500-episode moving average for better readability).

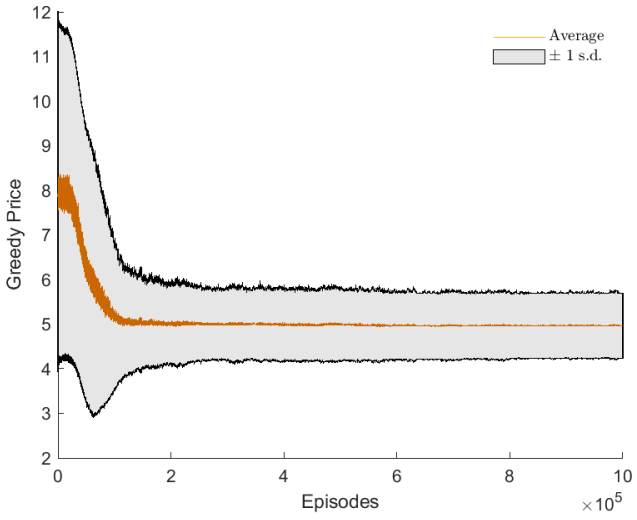
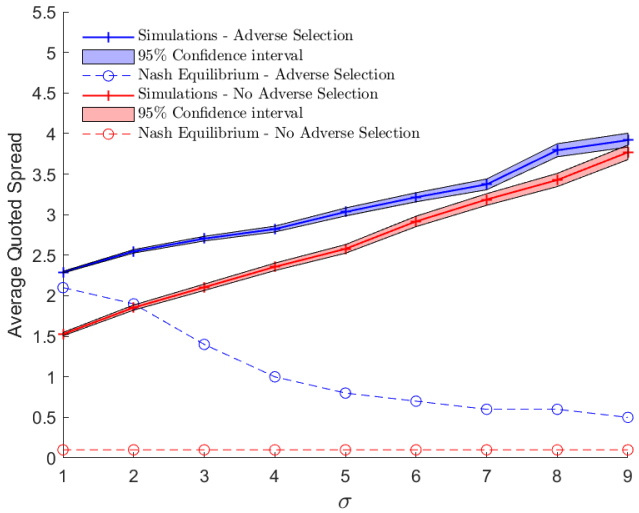


Figure 2: **Greedy price of AM 1 in the adverse-selection case, baseline parameters:** $\sigma = 5$, $\Delta_v = 4$, $N = 2$, $\mu = \frac{1}{2}$, $\mathbb{E}(v) = 2$, $T = 1,000,000$, and $K = 1,000$.

Panel A: Average Quoted Spread.

This graph plots the average over 1,000 experiments of the quoted spread, with 95% confidence intervals, both in the adverse-selection case and the no-adverse-selection case. The graph additionally plots the values of the quoted spread in both cases, in the Glosten-Milgrom benchmark of Section 2.2 (accounting for price discreteness).



Panel B: Average Realized Spread.

This graph plots the average over 1,000 experiments of the realized spread, with 95% confidence intervals, both in the adverse-selection case and the no-adverse-selection case. The graph additionally plots the values of the realized spread in both cases, in the Glosten-Milgrom benchmark of Section 2.2 (accounting for price discreteness).

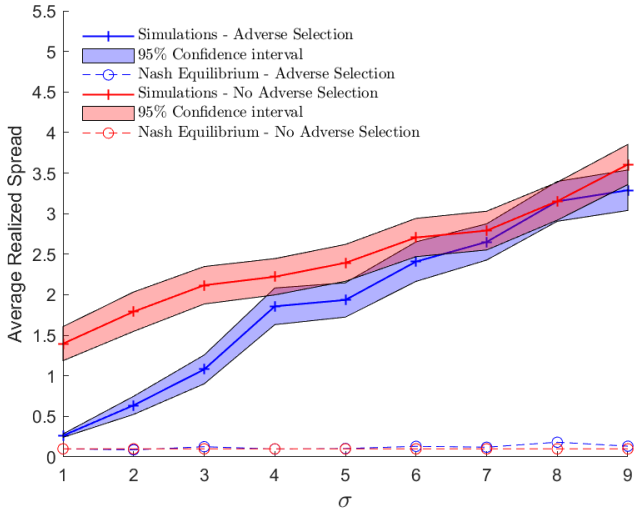
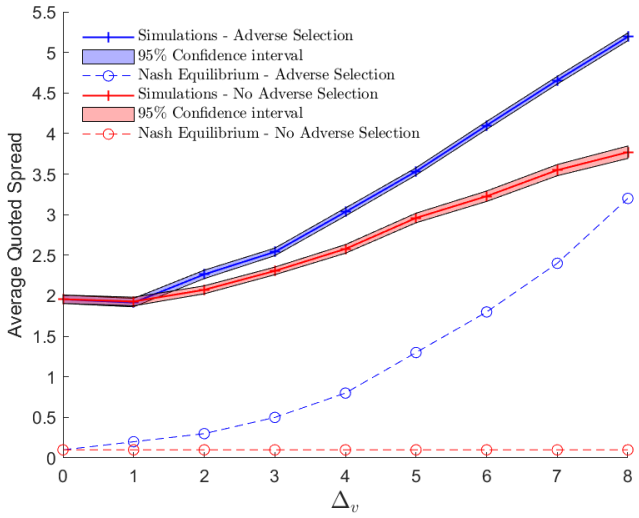


Figure 3: **Average Quoted Spread $\bar{QS}$ and Average Realized Spread $\bar{RS}$ in the adverse-selection case and the no-adverse-selection case, for different values of the dispersion of clients' liquidity shocks σ.** The other parameters are $\Delta_v = 4$, $N = 2$, $\mu = \frac{1}{2}$, $\mathbb{E}(v) = 2$, $T = 1,000,000$, and $K = 1,000$.

Panel A: Average Quoted Spread.

This graph plots the average over 1,000 experiments of the quoted spread, with 95% confidence intervals, both in the adverse-selection case and the no-adverse-selection case. The graph additionally plots the values of the quoted spread in both cases, in the Glosten-Milgrom benchmark of Section 2.2 (accounting for price discreteness).



Panel B: Average Realized Spread.

This graph plots the average over 1,000 experiments of the realized spread, with 95% confidence intervals, both in the adverse-selection case and the no-adverse-selection case. The graph additionally plots the values of the realized spread in both cases, in the Glosten-Milgrom benchmark of Section 2.2 (accounting for price discreteness).

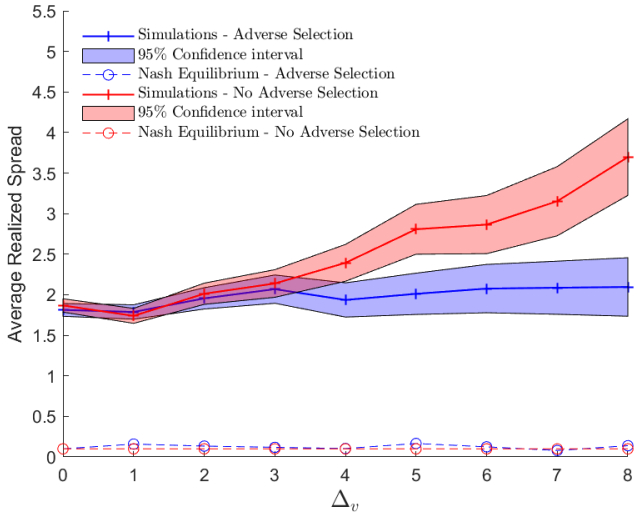
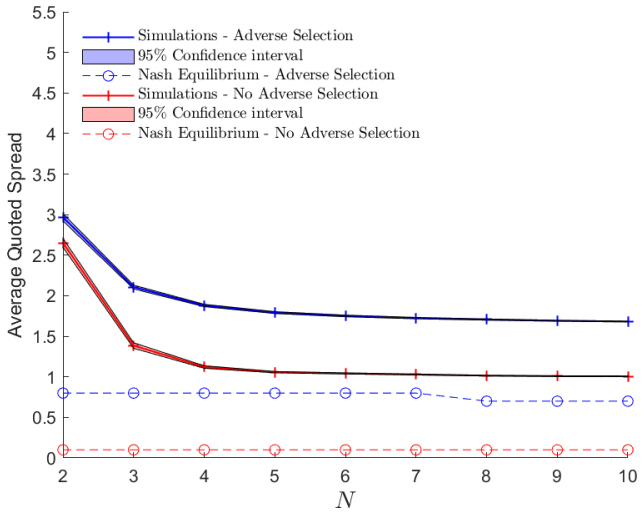


Figure 4: **Average Quoted Spread $\bar{QS}$ and Average Realized Spread $\bar{RS}$ in the adverse-selection case and the no-adverse-selection case, for different values of the asset volatility Δ_v.** The other parameters are $\sigma = 5$, $N = 2$, $\mu = \frac{1}{2}$, $\mathbb{E}(v) = 2$, $T = 1,000,000$, and $K = 1,000$.

Panel A: Average Quoted Spread.

This graph plots the average over 1,000 experiments of the quoted spread, with 95% confidence intervals, both in the adverse-selection case and the no-adverse-selection case. The graph additionally plots the values of the quoted spread in both cases, in the Glosten-Milgrom benchmark of Section 2.2 (accounting for price discreteness).



Panel B: Average Realized Spread.

This graph plots the average over 1,000 experiments of the realized spread, with 95% confidence intervals, both in the adverse-selection case and the no-adverse-selection case. The graph additionally plots the values of the realized spread in both cases, in the Glosten-Milgrom benchmark of Section 2.2 (accounting for price discreteness).

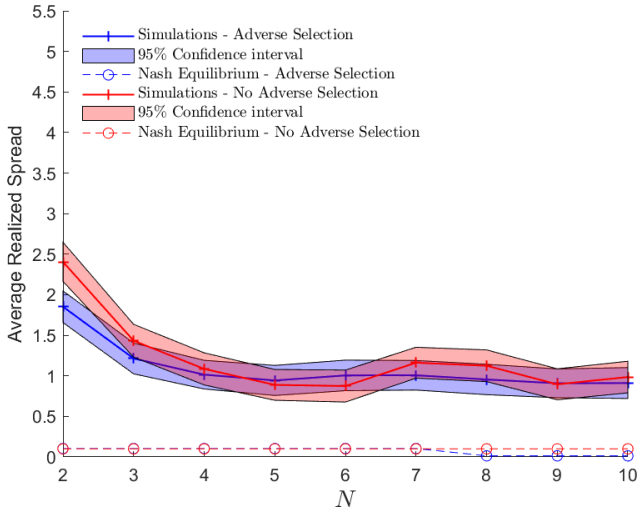


Figure 5: **Average Quoted Spread $\bar{QS}$ and Average Realized Spread $\bar{RS}$ in the adverse-selection case and the no-adverse-selection case, for different values of the number N of AMs.** The other parameters are $\sigma = 5$, $\Delta_v = 4$, $\mu = \frac{1}{2}$, $\mathbb{E}(v) = 2$, $T = 1,000,000$, and $K = 1,000$.

This graph plots the average over 1,000 experiments of the first-period and second-period prices, with 95% confidence intervals. The graph additionally plots the values of these prices in the Glosten-Milgrom benchmark of Section 5.1 (accounting for price discreteness).

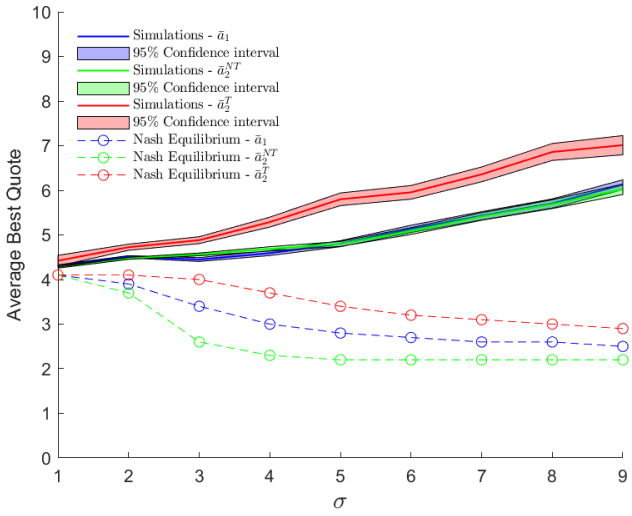


Figure 6: **Average first-period price $\bar{a}_1$ and second-period price after a trade $\bar{a}_2^T$ and after no trade $\bar{a}_2^{NT}$, for different values of the dispersion of clients' liquidity shocks σ.** The other parameters are $\Delta_v = 4$, $N = 2$, $\mu = \frac{1}{2}$, $\mathbb{E}(v) = 2$, $T = 1,000,000$, and $K = 1,000$.

A.3 Testable Hypotheses

We just consider H.1., as the other hypotheses are straightforward. As explained in the text, the Glosten and Milgrom price solves:

$$a^* = \mathrm{E}_\mu(\tilde{v} \mid \tilde{v}^C > a^*), \tag{A.1}$$

We define $F(a; \sigma, \Delta_v) := a - \mathrm{E}_\mu(\tilde{v} \mid \tilde{v}^C > a)$. The Glosten and Milgrom price is the smallest root of:

$$F(a^*; \sigma, \Delta_v) = 0. \tag{A.2}$$

We first show that there is always a solution to (A.2). Thus, a competitive price always exists in our setting.

Existence of the Glosten and Milgrom price. Let $\Pr_\mu(\tilde{v} = v_H \mid \tilde{v}^C > a)$ be the probability that the asset payoff is high ($v = v_H$) conditional on a trade, given dealers' beliefs (μ) about the payoff of the asset. Thus:

$$\mathrm{E}_\mu(\tilde{v} \mid \tilde{v}^C > a) = \Pr_\mu(\tilde{v} = v_H \mid \tilde{v}^C > a)v_H + (1 - \Pr_\mu(\tilde{v} = v_H \mid \tilde{v}^C > a)v_L. \tag{A.3}$$

Therefore, as $\mathrm{E}_\mu(\tilde{v}) = \mu v_H + (1 - \mu)v_L$, we have

$$\mathrm{E}_\mu(\tilde{v} \mid \tilde{v}^C > a)) - \mathrm{E}_\mu(\tilde{v}) = [\Pr_\mu(\tilde{v} = v_H \mid \tilde{v}^C > a) - \mu](v_H - v_L). \tag{A.4}$$

It follows that:

$$F(a; \sigma, \Delta_v) = a - \mathrm{E}_\mu(\tilde{v}) + (\Pr_\mu(\tilde{v} = v_H \mid \tilde{v}^C > a) - \mu)(v_H - v_L), \tag{A.5}$$

where $\Pr_\mu(\tilde{v} = v_H \mid \tilde{v}^C > a)$ is the probability that the asset payoff is high ($v = v_H$) conditional on a trade, given dealers' beliefs (μ) about the payoff of the asset. Standard calculations yield:

$$\Pr_\mu(\tilde{v} = v_H \mid \tilde{v}^C > a) = \frac{D(a, v_H)}{\mu D(a, v_H) + (1 - \mu)D(a, v_L)}\mu, \tag{A.6}$$

where $D(a,v))$ is defined in (3). As $D(a,v_L)>0$, $\Pr_\mu(\tilde{v}=v_H \mid \tilde{v}^C>a)<1$ when $\mu<1$ and a finite.

Observe that: (i) $F(.)$ is continuous, (ii) $F(a;\sigma,\Delta_v)<0$ for any $a\leq \mathrm{E}_\mu(\tilde{v})$ and (iii) $F(v_H;\sigma,\Delta_v)>0$ since $\Pr_\mu(\tilde{v}=v_H \mid \tilde{v}^C>a)<1$ for all finite a (in particular $a=v_H$). Thus, there is at least one solution in $(\mathrm{E}_\mu(v),v_H)$ to (A.2). If there are multiple solutions, the competitive one is the smallest. Observe that as $F(\mathrm{E}_\mu(\tilde{v});\sigma,\Delta_v)<0$, the Glosten and Milgrom price must be such that:

$$\frac{\partial F(a^*;\sigma,\Delta_v)}{\partial a}\mid_{a=a^*}>0. \tag{A.7}$$

If it were not the case, there would be another solution to (A.2) in $(\mathrm{E}_\mu(v),a^*)$. A contradiction since a^* is the smallest solution to (A.2).

Effect of σ on a^*. We deduce from (A.2) that:

$$\frac{\partial a^*}{\partial \sigma}=-\frac{\frac{\partial F}{\partial a}\mid_{a=a^*}}{\frac{\partial F}{\partial \sigma}\mid_{a=a^*}}. \tag{A.8}$$

As $\frac{\partial F}{\partial a}\mid_{a=a^*}>0$, we have that $\frac{\partial a^*}{\partial \sigma}<0$ if and only if $\frac{\partial F}{\partial \sigma}>0$. We now show that this is the case.

Observe, using (A.5), that $\frac{\partial F}{\partial \sigma}>0$ iff $\Pr_\mu(\tilde{v}=v_H \mid \tilde{v}^C>a^*)$ decreases with σ. Using (A.6), we obtain

$$\frac{\partial \Pr_\mu(\tilde{v}=v_H \mid \tilde{v}^C>a^*)}{\partial \sigma}=\mu\left[\frac{\frac{\partial D(a^*,v_H)}{\partial \sigma}\mathrm{E}_\mu(D(a^*,\tilde{v}))+\frac{\partial \mathrm{E}_\mu(V(a^*,\tilde{v}^C))}{\partial \sigma}D(a^*,v_H)}{(\mathrm{E}_\mu(V(a^*,\tilde{v}^C)))^2}\right]. \tag{A.9}$$

It follows, after simplifying the numerator of the previous expression, that $\frac{\partial \Pr_\mu(\tilde{v}=v_H \mid \tilde{v}^C>a^*)}{\partial \sigma}$ has the same sign as:

$$D(a^*,v_L)\frac{\partial D(a^*,v_H)}{\partial \sigma}-D(a^*,v_H)\frac{\partial D(a^*,v_L)}{\partial \sigma}.$$

Now remember that $D(a^*,v)=1-G(a^*-v)$ where $G(.)$ is the c.d.f of a Gaussian variable with mean zero and variance σ^2. It follows that $\frac{\partial D(a^*,v)}{\partial \sigma}=(\sqrt{2\pi}\sigma^2)^{-1}exp(-\frac{(a^*-v)^2}{2\sigma^2})(a^*-v)$. Hence, the previous expression is negative since $a^*\in(v_L,v_H)$. Hence, a^* decreases with σ.

Effect of Δ_v on a^*. We can proceed in the same way for analyzing the effect of Δ_v on a^*. The

same reasoning as before shows that a^* increases with Δ_v if and only if $\Pr_\mu(\tilde{v} = v_H \mid \tilde{v}^C > a^*)$ increases with Δ_v. After some algebra, one obtains that $\frac{\partial \Pr_\mu(\tilde{v}=v_H|\tilde{v}^C>a^*)}{\partial \Delta_v}$ has the same sign as:

$$D(a^*, v_L)\frac{\partial D(a^*, v_H)}{\partial \Delta_v} - D(a^* - v_H)\frac{\partial D(a^*, v_L)}{\partial \Delta_v}.$$

Now remember that (i) $D(a^*, v) = 1 - G(a^* - v)$ where $G(.)$ is the c.d.f of a Gaussian variable with mean zero and variance σ^2 and (ii) $v_H = \mu + \frac{\Delta_v}{2}$ and $v_L = \mu - \frac{\Delta_v}{2}$. It follows that $\frac{\partial D(a^*, v_H)}{\partial \Delta_v} > 0$ while $\frac{\partial D(a^*, v_L)}{\partial \Delta_v} < 0$. We deduce that $\frac{\partial \Pr_\mu(\tilde{v}=v_H|\tilde{v}^C>a^*)}{\partial \Delta_v} > 0$. Hence, we have shown that a^* increases with Δ_v.

A.4 The Variance of AMs' Profits

To simplify notations, in this section, we define: $p(a) := \mathrm{E}_{\frac{1}{2}}(V(a, \tilde{v}^C)) = \frac{D(a_1, v_H)}{2} + \frac{D(a_1, v_L)}{2}$. Consider the case without adverse selection first. The distribution of AM 1's profit $\Pi(a_1, a_2, \tilde{v}^C, \tilde{v})$ when $a_1 < a_2$ (the case assumed in the text) is as follows:

1. $(a_1 - v_H)$ with probability $\frac{D(a_1, v_H)}{4} + \frac{D(a_1, v_L)}{4} = \frac{p(a)}{2}$.
2. $(a_1 - v_L)$ with probability $\frac{D(a_1, v_H)}{4} + \frac{D(a_1, v_L)}{4} = \frac{p(a)}{2}$.
3. 0 with probability $1 - p(a)$.

Denote by $\bar{\Pi}_{n.as} = p(a_1)(a_1 - \mathrm{E}_{\frac{1}{2}}(v))$, AM 1's expected profit in this case (remember again that $a_1 < a_2$). By definition (index $n.as$ refers to "no adverse selection"):

$$\mathrm{Var}_{n.as}(\Pi(a_1, a_2, \tilde{v}^C, \tilde{v})) = \mathrm{E}((\Pi(a_1, a_2, \tilde{v}^C, \tilde{v}) - \bar{\Pi}_{n.as})^2). \tag{A.10}$$

That is:

$$\mathrm{Var}_{n.as} = p(a_1)(a_1 - \mathrm{E}_{\frac{1}{2}}(v) - \bar{\Pi}_{n.as})^2 + p(a_1)\frac{\Delta_v^2}{4} + \bar{\Pi}_{n.as}^2 - 2p(a_1)\bar{\Pi}_{n.as}(a_1 - \mathrm{E}_{\frac{1}{2}}(v)) \tag{A.11}$$

Hence, as $\bar{\Pi}_{n.as} = p(a_1)(a_1 - \mathrm{E}_{\frac{1}{2}}(v))$, we deduce that:

$$\mathrm{Var}_{n.as} = p(a_1)(1 - p(a_1))(a_1 - \mathrm{E}_{\frac{1}{2}}(v))^2 + p(a_1)\frac{\Delta_v^2}{4}, \tag{A.12}$$

which is (13) in the text since $p(a_1) := \mathrm{E}_{\frac{1}{2}}(V(a, \tilde{v}^C))$.

Now consider the case with adverse selection. The distribution of AM 1's profit is then as follows:

1. $(a_1 - v_H)$ with probability $\frac{D(a_1, v_H)}{2}$.
2. $(a_1 - v_L)$ with probability $\frac{D(a_1, v_L)}{2}$
3. 0 with probability $1 - p(a_1)$.

Observe that, holding a_1 constant, adverse selection does not reduce the likelihood of a trade ($p(a_1)$ in either case). However, it shifts the distribution of profits when there is a trade to the left because $(a_1 - v_H) < (a_1 - v_L)$ and $D(a_1, v_H) > p(a_1) > D(a_1, v_L)$. Intuitively, as $(v_H - a_1)$ is closer to zero than $(a_1 - v_L)$ (when $a_1 > \mathrm{E}_{\frac{1}{2}}(v)$), this shift reduces the dispersion of trading profits (there is more mass overall close to zero).

As an example, Figure 7 compares the distribution of realized profits for AM 1 when it posts a price of 4.9 while AM 2 posts a price of 5 in the baseline case ($\sigma = 5$ and $\Delta_v = 4$) in the case without adverse selection (red) and the case with adverse selection (blue). In this case, AM 1's realized profit can be 0 (the client does not trade), 0.9 (the client buys and the asset payoff is $v_H = 4$) or 4.9 (the client buys and the asset payoff is is $v_L = 0$). As the figure shows, in the presence of adverse selection, the distribution of profits is more skwewed to the left, toward zero, due to adverse selection (the likelihood of a buy when the asset payoff is large is higher than when the payoff is small). As a result, the variance of profits is smaller when there is adverse selection (1.78 vs 2.93).

More formally, denote $\bar{\Pi}_{as}$ AM1's expected profit with adverse selection ('as')'when it quotes $a_1 < a_2$ (this is given by (6) with $Z = 1$). By definition:

$$\mathrm{Var}_{as}(\Pi(a_1, a_2, \tilde{v}^C, \tilde{v})) = \mathrm{E}((\Pi(a_1, a_2, \tilde{v}^C, \tilde{v}) - \bar{\Pi}_{as})^2). \tag{A.13}$$

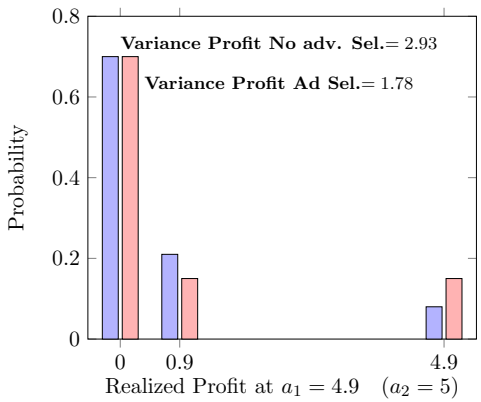


Figure 7

That is, using the fact that $\Delta_D := D(a_1, v_H) - D(a_1, v_L)$:

$$\begin{aligned} \text{Var}_{as} = \frac{p(a_1)}{2}(a_1 - v_H - \bar{\Pi}_{as})^2 + \frac{p(a_1)}{2}(a_1 - v_L - \bar{\Pi}_{as})^2 + \\ (1 - p(a_1))\bar{\Pi}_{as}^2 - \frac{\Delta_D}{4}[(a_1 - v_L - \bar{\Pi}_{as})^2 - (a_1 - v_H - \bar{\Pi}_{as})^2)], \end{aligned} \tag{A.14}$$

and therefore, after some straightforward algebra:

$$\text{Var}_{as} = \text{Var}_{n.as} - \frac{\Delta_D}{4}[(a_1 - v_L - \bar{\Pi}_{as})^2 - (a_1 - v_H - \bar{\Pi}_{as})^2)], \tag{A.15}$$

The last term in bracket is negative because $v_H - a_1 > a_1 - v_L$ for $a_1 > \text{E}_{\frac{1}{2}}(v)$. Thus, $\text{Var}_{n.as} < \text{Var}_{as}$. Moreover, we can rewrite the term in bracket to obtain:

$$\text{Var}_{as} = \text{Var}_{n.as} - \frac{\Delta_D \Delta_v}{2}[(a_1 - \text{E}_{\frac{1}{2}}(v)) - \bar{\Pi}_{as}/2], \tag{A.16}$$

as claimed in the text.

To analyze the effect of σ on Var_{as}, observe first that $p(a_1) = \text{E}_\mu(V(a, \tilde{v}^C))$ increases with σ for $a > E(v)$. Indeed:

$$\frac{\partial \text{E}_\mu(V(a, \tilde{v}^C))}{\partial \sigma} = \mu \frac{\partial D(a, v_H)}{\partial \sigma} + (1 - \mu)\frac{\partial D(a, v_L)}{\partial \sigma}. \tag{A.17}$$

As $D(a, v) = 1 - G(a - v)$ and $G(.)$ is the c.d.f of a Gaussian variable with mean zero and variance

σ^2, we have $\frac{\partial D(a,v)}{\partial \sigma} = (\sqrt{2\pi}\sigma^2)^{-1} exp(-\frac{(a-v)^2}{2\sigma^2})(a-v)$. As $a - v_H < a - v_L$, we deduce that:

$$\frac{\partial \mathrm{E}_\mu(V(a,\tilde{v}^C))}{\partial \sigma} = \mu \frac{\partial D(a,v_H)}{\partial \sigma} + (1-\mu)\frac{\partial D(a,v_L)}{\partial \sigma} > (\sqrt{2\pi}\sigma^2)^{-1} exp(-\frac{(a-v_L)^2}{2\sigma^2})(a - E(v)) > 0, \tag{A.18}$$

for $a > E(v)$. Thus, for $a > E(v)$, $\mathrm{E}_\mu(V(a,\tilde{v}^C))$ is maximal when σ goes to infinity and therefore $\mathrm{E}_\mu(V(a,\tilde{v}^C)) < \frac{1}{2}$ (since $D(a,v)$ goes to $\frac{1}{2}$ when σ goes to infinity). It follows from (13) that Var_{as} increases with σ. A similar reasoning shows that Var_{as} increases with Δ_v.

Now consider the effect of σ on Var_{as}. Substituting $\bar{\Pi}_{as}$ by its expression (equation (6) with $Z = 1$) in(A.16) and rearranging, we obtain:

$$\mathrm{Var}_{as} = \mathrm{Var}_{n.as} - \left[(\frac{\Delta_D \Delta_v}{2})((a_1 - \mathrm{E}_{\frac{1}{2}}(v))(1 - p(a_1)) + \frac{\Delta_D \Delta_v}{2})\right]. \tag{A.19}$$

The first term (Var_{as}) increases with σ (as shown before) while the second term in brackets decreases with σ for $a_1 \geq \mathrm{E}_{\frac{1}{2}}(v)$ (the relevant case in our experiments) since Δ_D decreases with σ and $p(a_1)$ increases with σ. As this term is multiplied by -1, we deduce that Var_{as} also increases with σ.

A.5 Convergence

As explained in the text, the environment in which AMs operate implies that the Q-matrices do not converge to a single value. More precisely, suppose AMs keep playing the same price profile $a \in \mathcal{A}^N$ at every episode t. Let a_m be the best price in a, and suppose it is played by AM n. Let $q_{m,n,t}$ denote the m-th entry in AM n's Q-matrix at time t, i.e., the value that at time t, AM n attaches to playing price a_m. We show that for any t,

$$\exists \Delta_q > 0, \text{ and } \epsilon > 0 \text{ s.t. } Pr(|q_{m,n,t} - q_{m,n,t+1}| \geq \Delta_q) \geq \epsilon.$$

That is, each entry of the Q-matrix cannot converge in probability to a single value.[39] Thus, no matter how large is the number of episodes T, there is a strictly positive probability bounded away from 0 that the Q-matrix of the dealers posting the best price in episode t changes by more than a

[39] $q_{m,n,t}$ converges in probability to a real number $q \in \mathbb{R}$ if for any $\varepsilon > 0$, one has $\lim_{t\to\infty} Pr(|q_{m,n,t} - q| \geq \varepsilon) = 0$.

fixed amount $\Delta_q > 0$.

Formally, let define

$$\Delta_m^* := \frac{\alpha}{2} \max \left\{ v_H - v_L, \frac{v_H - v_L}{2} + \left| a_m - \frac{v_H + v_L}{2} \right| \right\},$$

that is strictly positive, as long as $v_H \neq v_L$ or $a_m \neq \frac{v_H + v_L}{2}$. Let

$$P_m^* := \min \left\{ \frac{1}{2N} D(a_m, v_L), 1 - \frac{1}{2}(D(a_m, v_L) + D(a_m, v_H)) \right\},$$

that is strictly positive because for any finite a_m and $v \in \{v_l, v_H\}$, one has $0 < D(a, v) < 1$.

Lemma 1. *For any given t and $a_m \in \mathcal{A}$, if $a_{n,t} = a_m = a_t^{\min}$, then ,*

$$\Pr\left(|q_{m,n,t} - q_{m,n,t+1}| \geq \Delta_m^*\right) \geq P_m^*,$$

Proof. Fix a price a_m and a dealer n. Suppose that at episode t the dealer's price is $a_{n,t} = a_m$ and it is the lowest price among dealers, i.e. $a_{nt} = a_m = a_t^{\min}$. Then three outcomes are possible: either the dealer does not trade, the dealer sells the asset worth v_H, or the dealer sells the asset worth v_L. In all cases the Q-matrix is updated. If the dealer does not trade then $\pi_{n,t} = 0$ and $q_{m,n,t+1} = (1 - \alpha) q_{m,n,t}$, implying

$$|q_{m,n,t} - q_{m,n,t+1}| = \alpha |q_{m,n,t}|$$

If the dealer trades then $q_{m,n,t+1} = \alpha(a_m - \tilde{v}) + (1 - \alpha) q_{m,n,t+1}$, and thus if $\tilde{v} = v_H$,

$$|q_{m,n,t} - q_{m,n,t+1}| = \alpha |a_m - v_H - q_{m,n,t}|$$

whereas if if $\tilde{v} = v_L$,

$$|q_{m,n,t} - q_{m,n,t+1}| = \alpha |a_m - v_L - q_{m,n,t}|.$$

Denote $\Delta_m(q) := \alpha \max\{|q|, |a_m - v_H - q|, |a_m - v_L - q|\}$. This is the maximum possible value that $|q_{m,n,t} - q_{m,n,t+1}|$ can take, given that $q_{m,n,t} = q$. Note that three situations are possible.

If $a_m < v_L$, then

$$\Delta_m(q) = \begin{cases} \alpha(-q + v_H + a_m) \text{ for } q \leq \frac{v_H - a_m}{2} \\ \alpha q \text{ for } q > \frac{v_H - a_m}{2} \end{cases}$$

that implies

$$\Delta_m(q) \geq \frac{\alpha(v_H - a_m)}{2} \geq \frac{\alpha(v_H - v_L)}{2}, \frac{\alpha(v_L - a_m)}{2}$$

If $v_L \leq a_m \leq v_H$, then

$$\Delta_m(q) = \begin{cases} \alpha(-q + v_H - a_m) \text{ for } q \leq \frac{v_H + v_L}{2} - a_m \\ \alpha(q - v_L + a_m) \text{ for } q > \frac{v_L + v_H}{2} - a_m \end{cases}$$

that implies

$$\Delta_m(q) \geq \frac{\alpha(v_H - v_L)}{2} \geq \frac{\alpha(v_H - a_m)}{2}, \frac{\alpha(v_L - a_m)}{2}$$

If $a_m > v_H$, then

$$\Delta_m(q) = \begin{cases} -\alpha q \text{ for } q \leq \frac{v_L - a_m}{2} \\ \alpha(q - v_L + a_m) \text{ for } q > \frac{v_L - a_m}{2} \end{cases}$$

that implies

$$\Delta_m(q) \geq \frac{\alpha(a_m - v_L)}{2} \geq \frac{\alpha(v_H - v_L)}{2}, \frac{\alpha(v_H - a_m)}{2}$$

Hence we can write

$$\min_q \Delta_m(q) = \frac{\alpha}{2} \max\{a_m - v_L, v_H - a_m, v_H - v_L\} = \frac{\alpha}{2} \max\left\{v_H - v_L, \frac{v_H - v_L}{2} + \left|a_m - \frac{v_H + v_L}{2}\right|\right\} = \Delta_m^*$$

In words, no matter the value of $q_{m,n,t}$, at least one of the three possible outcomes mentioned above leads to $|q_{m,n,t} - q_{m,n,t+1}| \geq \Delta_m^*$. Thus the probability that $|q_{m,n,t} - q_{m,n,t+1}| \geq \Delta_m^*$ cannot be smaller than the smallest of the probabilities of these three events.

Now, given $a_{n,t} = a_m = a_t^{\min}$, the probability that the dealer sells the asset worth v_H, is at least $\frac{1}{2N} D(a_m, v_H)$. The probability that the dealer sells the asset worth v_L, is at least $\frac{1}{2N} D(a_m, v_L) < \frac{1}{2N} D(a_m, v_H)$. The probability that the dealer does not trade is $1 - \frac{1}{2}(D(a_m, v_L) + D(a_m, v_H))$,

hence the expression for P_m^*. Q.E.D.

A.6 Nash Equilibria

In this section we analyze the Nash equilibria of the one-shot game when market makers are constrained to choose prices from a finite grid (a positive tick size). We first show that when the tick size is small enough or the number N of market makers is large enough the game has a unique Nash equilibrium. However it is possible that for N relatively small and tick size relatively large the game has more than one pure Nash equilibrium. Namely for the value of the parameters in the range of our experiment we find that the game has either 1 or 2 pure Nash equilibria. We show that for $N = 2$, if the game has 2 pure Nash equilibria then it also has one mixed strategy equilibrium where market makers independently randomize their quotes over the two prices that form the two pure equilibria.

Denote $\Pi(a)$ the expected payoff of a monopolistic market maker who sets a price a. Namely

$$\Pi(a) = \mu D(a, v_H)(a - v_H) + (1 - \mu)D(a, v_*L)(a - v_L)$$

Let a^* be the smallest solution of the equation $\Pi(a) = 0$. This is the equilibrium price in the game where market makers can chose their prices on the real line. We know that a^* exists because $\Pi(a)$ is continuous in a, strictly negative for $a < v_L$ and strictly positive for $a > v_H$.

Let us now consider the game with $N > 1$ market makers that have to chose their prices on a grid $\mathcal{A}$, and denote δ the tick size. Without loss of generality we can assume that a^* is not on the price grid, i.e., $a^* \notin \mathcal{A}$.

Lemma 2. *In the game with N market makers, an action profile $\{a_1, a_2, \dots a_N\} \in \mathcal{A}^N$ is a pure Nash equilibrium if and only if the following two conditions are satisfied,*

1. *All market makers set the same price $a \in \mathcal{A}$*

2. *The price a is such that $\Pi(a) \geq 0$ and*

$$\frac{1}{N}\Pi(a) \geq \max_{a' \in \mathcal{A}, a' < a} \{\Pi(a')\} \tag{A.20}$$

Proof. Sufficient condition: suppose that all market makers except market maker i set a price equal to a, and that a satisfies condition (A.20). Let us consider the best response of market maker i. The expected payoff from playing $a' = a$ is $\frac{1}{N}\Pi(a) \geq 0$, as the market maker i has to share the payoff $\Pi(a)$ with the other $n-1$ market makers. The expected payoff from undercutting the other market makers by playing some $a' < a$ is $\Pi(a') \leq \frac{1}{N}\Pi(a)$, by condition (A.20). Whereas the expected payoff from from playing some $a'' > a$ is $0 \leq \frac{1}{N}\Pi(a)$, as no client trades with market maker i. Hence $a_i = a$ for all i is a pure Nash equilibrium.

Necessary condition: Suppose the action profile $\{a_1, a_2, \dots a_N\} \in \mathcal{A}^N$ forms a Nash equilibrium, and let $a^{\min}$ be the lowest offered price. If $\Pi(a^{\min}) < 0$, then the MM playing $a^{\min}$ gets a negative profit, and he has a profitable deviation by setting any $a' > v_H$. Hence, because prices belong to a discrete grid without loss of generality it must be that $\Pi(a^{\min}) > 0$. If there is a market maker i such that $a_i \neq a^{\min}$, then $a_i > a^{\min}$ and the market maker's payoff is nil. But then market maker i has a profitable deviation by playing $a^{\min}$ that provides him with a fraction of the strictly positive payoff $\Pi(a^{\min})$. Hence all market makers must play $a^{\min}$ and get a payoff of $\frac{1}{N}\Pi(a^{\min})$. If there is $a' < a^{\min}$ such that $\Pi(a') > \frac{1}{N}\Pi(a^{\min})$ then playing such a' would be a profitable deviation. Hence conditions 1. and 2. in the Lemma are necessary conditions for an equilibrium. □

Denote $\hat{a} \in \mathcal{A}$ the smallest price in the grid larger than a^*. Formally

$$\hat{a} = \min\{a \in \mathcal{A}, s.t. \Pi(a) \geq 0\}.$$

Note that if the price grid δ is small enough, then $\hat{a}$ is the price on the grid closest to a^* weakly greater than a^*. Then we have

Corollary 1. *If N is large enough or δ is small enough then all MMs playing $\hat{a}$ is the unique Nash equilibrium of the game.*

Proof. Let first show that playing $\hat{a}$, is an equilibrium. Because of the definition of $\hat{a}$, all $a' < \hat{a}$ on the grid $\mathcal{A}$ provide strictly negative payoff, and generically we have $\Pi(\hat{a}) > 0$. Thus $\hat{a}$ satisfies condition 2 in the Lemma 2 and thus playing a is a Nash equilibrium.

Let now show that if N is large or δ small, then there are no other equilibria. Suppose that

there is another equilibrium, where all MMs play $a \neq \hat{a}$. Then $\Pi(a) > 0$ and hence $a > \hat{a}$. If a player deviates from this equilibrium and plays $\hat{a}$ instead, then his payoff is $\Pi(\hat{a}) > \frac{1}{N}\Pi(a)$ for $N > \pi(a)/\pi(\hat{a})$. Thus for N large enough playing $a \neq \hat{a}$ cannot be an equilibrium as it violates condition (A.20).

Now fix N and suppose that there is an equilibrium where $a > \hat{a}$, and consider the deviation to the largest price on the grid that is smaller than a. This is $a' = a - \delta$. If a MM deviates and plays a', then his payoff is $\Pi(a-\delta)$ that tends to $\Pi(a)$ as δ goes to 0. Thus for δ small enough and $n > 1$, $\Pi(a') > \frac{1}{N}\Pi(a)$, and thus a cannot be a Nash equilibrium as it violates condition (A.20). □

Because the δ we use in our experiments is relatively large, for N relatively small we find that depending on the value of the parameters, the game has either 1 or 2 pure Nash equilibria. In the next Lemma we show that if the 2-payer game has two Nash equilibria, then it also has a third equilibrium in mixed strategies.

Lemma 3. *Suppose that for $N = 2$, thare are two pure Nash equilibria: $\hat{a}$ and $a > \hat{a}$. Then the game also has a mixed strategy equilibrium where market makers independently randomize between setting a price of $a_i = a$ with probability $\eta = \frac{\Pi(\hat{a})}{\Pi(a)-\Pi(\hat{a})}$ and $a_i = \hat{a}$ with the complementary probability.*

Proof. Note first that both a and $\hat{a}$ must satisfy condition (A.20). Namely condition (A.20) applied to $\hat{a}$ implies $\Pi(\hat{a}) > 0$, and if applied to a, implies $\Pi(\hat{a}) < \frac{1}{2}\Pi(a)$. These two inequalities imply that $0 < \eta < 1$.

Not that if the other MM j plays a and $\hat{a}$ with probability η and $1-\eta$, respectively, then MM i is indifferent between playing a or $\hat{a}$. This because η is the solution of the following indifference condition

$$\underbrace{\eta\Pi(\hat{a}) + \frac{1}{2}(1-\eta)\Pi(\hat{a})}_{\text{expected payoff from playing } \hat{a}} = \underbrace{\frac{1}{2}\eta\Pi(a)}_{\text{expected payoff from playing } a}$$

Both actions lead to an expected payoff of

$$\Pi^{mix} = \frac{\Pi(a)\Pi(\hat{a})}{2(\Pi(a) - \Pi(\hat{a}))} > 0$$

Let show that by unilaterally deviating to any $a' \notin \{\hat{a}, a\}$ MM i cannot gain more than Π^{mix}. For

$a' < \hat{a}$ the deviation payoff is strictly negative, by definition of $\hat{a}$. For $\hat{a} < a' < a$ the deviation payoff is

$$\eta\Pi(a') = \frac{\Pi(a')\Pi(\hat{a})}{\Pi(a) - \Pi(\hat{a})} \leq \frac{\Pi(a)\Pi(\hat{a})}{2(\Pi(a) - \Pi(\hat{a}))} = \Pi^{mix},$$

where the inequality follows from condition (A.20) applied to equilibrium a, that implies $\frac{1}{2}\Pi(a) \geq \Pi(a')$. For $a' > a$, MM does not trade and gets $0 < \Pi^{mix}$. □

A.7 Algorithm used in the two-period case

We formally define the algorithms and the process we simulate in the two-period case.

For each AM n, we define $(N + 3)$ states, denoted s_n, as follows: (i) $s_n = \emptyset$ in the first trading round; (ii) $s_n = NT$ in the second trading round if no trade takes place in the first; (iii) $s_n \in \mathcal{S} = \left\{0, \frac{1}{N}, \frac{1}{N-1}, ..., \frac{1}{2}, 1\right\}$ is the number of shares sold by AM n if a trade took place in period 1 (depending on how many AMs shared the market). Each AM then relies on a Q-matrix $\mathbf{Q}_{n,t} \in \mathbb{R}^{M\times(N+3)}$, in which each line corresponds to a different price and each column to a state, ordered as in point (iii). We denote $q_{m,s,n,t}$ the (m, s) entry of matrix $\mathbf{Q}_{n,t}$.

We then modify the process described in Section 3.1 as follows. For any experiment k, we initialize the matrices $\mathbf{Q}_{n,0}$ with random values: Each $q_{m,s,n,0}$ (for $1 \leq m \leq M$, $1 \leq n \leq N$, and $s \in \mathcal{S}$) is i.i.d. and follows a uniform distribution over $[\underline{q}, \overline{q}]$. Then, in each episode t, we do the following:

Period 1:

1. For each AM n, we define $m^{1,*}_{n,t} = arg\max_m q_{m,\emptyset,n,t-1}$ the index associated with the highest value in matrix $\mathbf{Q}_{n,t-1}$ in state $s = \emptyset$, and we denote $a^{1,*}_{n,t} = a_{m^{1,*}_{n,t}}$ the corresponding greedy price.

2. For each AM n, with probability $\epsilon_t = e^{-\beta t}$ the AM "explores": it draws a random integer $\tilde{m}^1_{n,t}$ between 1 and M, all values being equiprobable, and plays $a^1_{n,t} = a_{\tilde{m}^1_{n,t}}$. With probability $1 - \epsilon_t$, the AM "exploits" and plays the greedy price $a^1_{n,t} = a^{1,*}_{n,t}$. The random draws leading to exploring or exploiting are i.i.d. across all AMs in a given trading round of a given episode.

3. We compute $a_t^{1,min} = \min_n\{a_{n,t}^1\}$, and draw $\tilde{v}_t$ and $\tilde{L}_{1,t}$. This determines the position $I_{n,t}^1$ taken by each AM in period 1 and the state $s_{n,t}$ it will be in when period 2 starts. Formally, denote $\mathcal{D}_t^1$ the set of AMs who quote $a_t^{1,min}$ and z_t^1 the size of this set. Then, if $\tilde{v}_t + \tilde{L}_{1,t} \geq a_t^{1,min}$ we have $I_{n,t}^1 = s_{n,t} = \frac{1}{z_t^1}$ for every $n \in \mathcal{D}_t^1$, and $I_{n,t}^1 = s_{n,t} = 0$ for $n \notin \mathcal{D}_t^1$. If $\tilde{v}_t + \tilde{L}_{1,t} < a_t^{1,min}$ then $I_{n,t}^1 = 0$ and $s_{n,t} = NT$ for every n.

4. We update the first column of the Q-matrix of each AM n as follows:

$$q_{m,\emptyset,n,t} = \begin{cases} \alpha[a_{n,t}^1 I_{n,t}^1 + \max_{m'} q_{m',s_{n,t},n,t-1}] + (1-\alpha) q_{m,\emptyset,n,t-1} & \text{if } a_{n,t}^1 = a_m \\ q_{m,\emptyset,n,t-1} & \text{if } a_{n,t} \neq a_m \end{cases} \tag{A.21}$$

Period 2:

1. At the beginning of period 2 we know the state $s_{n,t}$ in which AM n finds itself. We define $m_{n,t}^{2,*} = arg\max_m q_{m,s_{n,t},n,t-1}$ the index associated with the highest value in matrix $\mathbf{Q}_{n,t-1}$ in state $s = s_{n,t}$, and we denote $a_{n,t}^{2,*} = a_{m_{n,t}^{2,*}}$ the corresponding greedy price.

2. With probability ϵ_t the AM plays a random price $a_{n,t}^2$, following the same process as in period 1. With probability $1 - \epsilon_t$, the AM plays $a_{n,t}^2 = a_{n,t}^{2,*}$.

3. We compute $a_t^{2,min} = \min_n a_{n,t}^2$ and draw $\tilde{L}_{2,t}$. This determines the position $I_{n,t}^2$ taken by each AM in period 2, following the same rules as in period 1.

4. For each AM n, we only update the column corresponding to state $s_{n,t}$, as follows:

$$\forall 1 \leq n \leq N, q_{m,s_{n,t},n,t} = \begin{cases} \alpha[a_{n,t}^2 I_{n,t}^2 - \tilde{v}_t(I_{n,t}^1 + I_{n,t}^2)] + (1-\alpha) q_{m,s_{n,t},n,t-1} & \text{if } a_{n,t}^2 = a_m \\ q_{m,s_{n,t},n,t-1} & \text{if } a_{n,t}^2 \neq a_m \end{cases} \tag{A.22}$$

We repeat this process for T episodes, after which the experiment ends.

Online Appendix to "Algorithmic Pricing and Liquidity in Securities Markets"

Jean-Edouard Colliard, Thierry Foucault, and Stefano Lovo

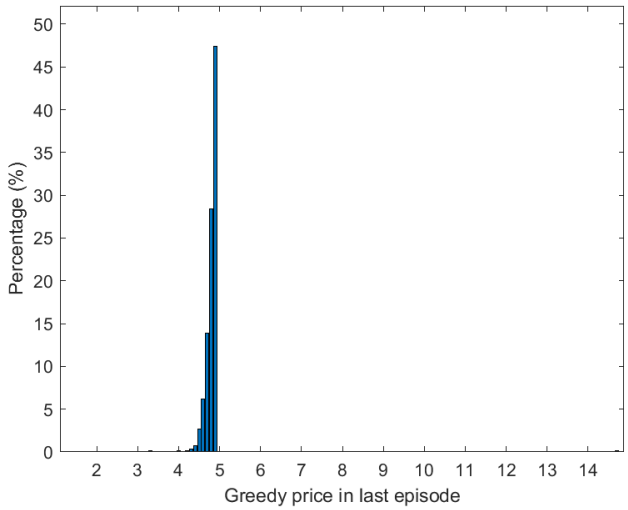
This Online Appendix provides additional robustness tests and experiments.

OA.1 Learning against a fixed-strategy opponent

We run the same experiments as in Figure 2, with the same parameters. The only difference is that only AM 1 is doing Q-learning. AM 2 plays a constant strategy $a_2 = 5.0$ in every episode. Figure OA.1 replicates Figure 2 in that case, with a histogram of the greedy price of AM 1 in episode T, and a plot of how the average greedy price of AM 1 evolves over episodes.

Panel A: Distribution of the greedy price of AM 1 in the last episode.

This panel shows a histogram of the greedy price of AM 1 in episode T: For each possible price a between 1.10 and 14.90 the bar indicates the percentage of the 1,000 experiments conducted in which $a^*_{1,T} = a$.



Panel B: Dynamics of the average greedy price of AM 1 for episodes 1 to T.

This graph shows for each episode t the average of AM 1's greedy price $a^*_{1,t}$ across the 1,000 experiments conducted. As a measure of dispersion, we also compute the standard deviation of $a^*_{1,t}$ across experiments and plot the average of $a^*_{1,t}$ plus/minus one standard deviation (with a 500-episode moving average for better readability).

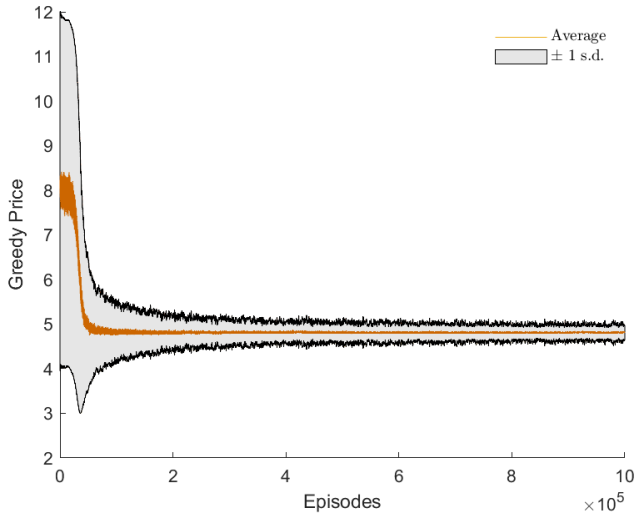


Figure OA.1: **Greedy price of AM 1 when AM 2 plays a constant price:** adverse-selection case, baseline parameters $\sigma = 5$, $\Delta_v = 4$, $N = 2$, $\mu = \frac{1}{2}$, $\mathbb{E}(v) = 2$, $T = 1,000,000$, and $K = 1,000$. AM 2 plays a constant price of 5.0 in every episode, while AM 1 uses a Q-learning algorithm with $\alpha = 0.01$ and $\beta = 0.00008$.

OA.2 Infinite experimentation and empirical average

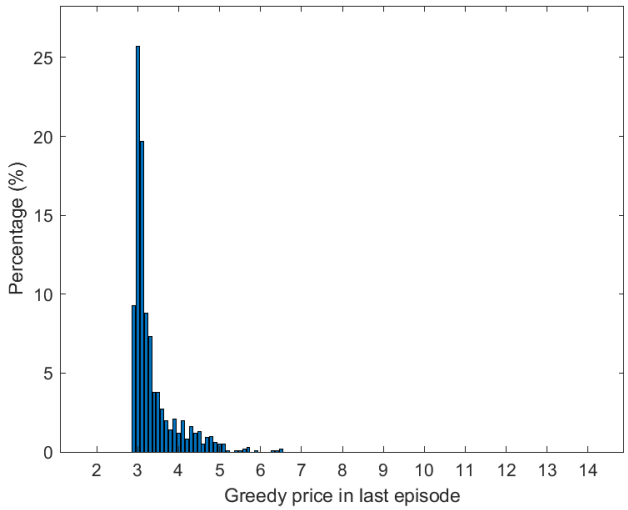
We run the same experiments as in Figure 2, with the same parameters, but we change the parameterization of both algorithms. We now have $\epsilon_t = 0.05 + 0.95 \exp^{-\beta t}$: for early episodes the experimentation probability ϵ_t will be high and then decrease exponentially like in the baseline case, but it will converge towards 0.05 instead of 0. Thus, in the long-run the algorithms will still experiment once every 20 episodes on average. Moreover, we change the updating rule (10) so that now the entries in the Q-matrix correspond to the empirical average of the profit obtained with each price. Formally, denoting $\nu_{m,n,t}$ the number of times price m has been tried by AM n before episode t, we update $q_{m,n,t}$ as:

$$q_{m,n,t} = \begin{cases} \frac{\pi_{n,t} + \nu_{m,n,t} q_{m,n,t-1}}{1 + \nu_{m,n,t}} & \text{if } a_{n,t} = a_m \\ q_{m,n,t} & \text{if } a_{n,t} \neq a_m \end{cases} \tag{OA.1}$$

We initialize each Q-matrix as in the baseline case, and start with $\nu_{m,n,1} = 1$ for every m and n. Figure OA.2 replicates Figure 2 in that case, with a histogram of the greedy price of AM 1 in episode T, and a plot of how the average greedy price of AM 1 evolves over episodes.

Panel A: Distribution of the greedy price of AM 1 in the last episode.

This panel shows a histogram of the greedy price of AM 1 in episode T: For each possible price a between 1.10 and 14.90 the bar indicates the percentage of the 1,000 experiments conducted in which $a^*_{1,T} = a$.



Panel B: Dynamics of the average greedy price of AM 1 for episodes 1 to T.

This graph shows for each episode t the average of AM 1's greedy price $a^*_{1,t}$ across the 1,000 experiments conducted. As a measure of dispersion, we also compute the standard deviation of $a^*_{1,t}$ across experiments and plot the average of $a^*_{1,t}$ plus/minus one standard deviation (with a 500-episode moving average for better readability).

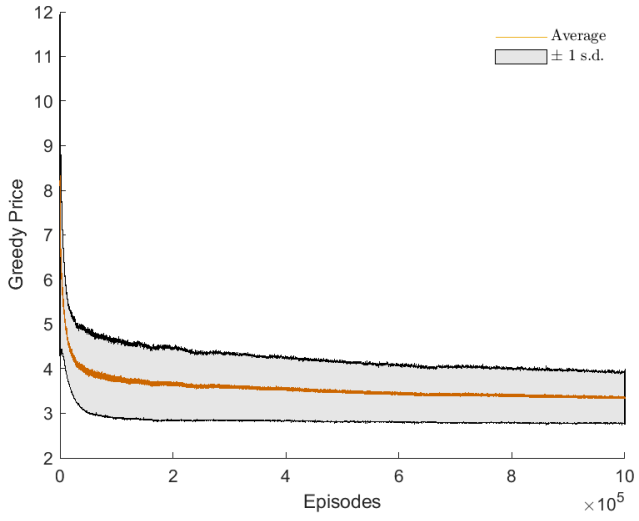


Figure OA.2: **Greedy price of AM 1 when AM 1 and AM 2 keep experimenting in the long-run:** adverse-selection case, baseline parameters $\sigma = 5$, $\Delta_v = 4$, $N = 2$, $\mu = \frac{1}{2}$, $\mathbb{E}(v) = 2$, $T = 1,000,000$, and $K = 1,000$. Both AMs use $\epsilon_t = 0.05 + 0.95\exp^{-\beta t}$ and the Q-matrix records the empirical average of the profit obtained with each price in past episodes.

OA.3 Experimenting more is not profitable

We run the same experiments as in Figure 2, with the same parameters, except that we allow AM 1 to use a different parameter β. We denote β_1 the experimentation parameter for AM 1 and β_2 the parameter for AM 2. We fix β_2 at its baseline value of 0.00008 and we make β_1 vary between 1/16 of β_2 and 16 times β_2. For each β_1 we compute the average profit realized by AM 1 across 1,000 experiments and over 100,000 episodes, 500,000 episodes, and 1,000,000 episodes. We observe that the average profit is non-monotonic in β_1: while experimenting allows AM 1 to get more information and adjust its price to the behavior of AM 2 for more episodes, it also leads AM 2 to choose lower prices in the long-run, and it is also costly as AM 1 will more often play a random, often suboptimal, price. We observe that the point $\beta_1 = 0.00008$ in the middle of the x-axis, which corresponds to our baseline parameterization, is close to being a best response to β_2.

This graph shows the average profit for AM 1 over the first 100,000 episodes, the first 500,000 episodes, and the entire 1,000,000 episodes, for different parameters β_1 used by AM 1, keeping β_2 constant. Note that the scale on the x-axis is not linear.

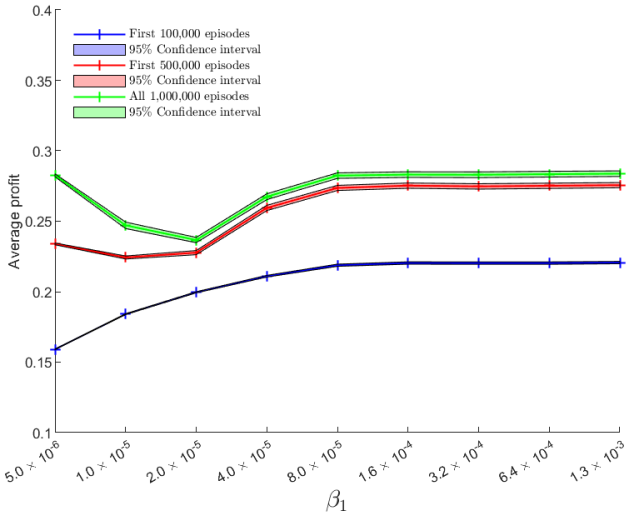


Figure OA.3: **Average profit of AM 1 for different values of β_1.** Adverse-selection case, baseline parameters $\sigma = 5$, $\Delta_v = 4$, $N = 2$, $\mu = \frac{1}{2}$, $\mathbb{E}(v) = 2$, $T = 1,000,000$, and $K = 1,000$. AM 1 uses β_1 and AM 2 uses $\beta_2 = 8.10^{-5}$.

OA.4 Waiting for the experiment to "converge" can be misleading

In this section we explain why we choose to run experiments in which algorithms interact for a large but fixed number T of episodes, instead of waiting for the algorithms to play the same actions for a certain number of times, as is done in other papers in the literature.

Consider the following two procedures for the numerical experiments:

- Fixed stopping time procedure: the algorithms play for a fixed number T of episodes.

- Random stopping time procedure: the algorithms play until they have both taken the same action for κ episodes in a row, then the procedure stops. The final episode is denoted $\tilde{T}$.

The random stopping procedure is in principle the appropriate thing to do if we know theoretically that the algorithms will eventually converge, in the sense that with probability 1 they will both play the same actions for every period after some random period. Then one can wait for the same actions to be repeated a large number of times κ, and if κ is large enough it is likely that the algorithms have indeed converged.

However, as we showed in Section A.5, the probability that our Q-learning algorithms converge in this sense is zero: there is a probability of 1 that an AM will change its optimal action if one waits for long enough. Then, the random stopping procedure implies that we are conditioning experimental observations on a specific path having been taken in the experiment. This may in principle bias the results.

To better understand this point, we consider a very simple example in which the correct quantity to estimate can be computed theoretically. Assume there is only one Q-learning algorithm that can take two actions a_1 and a_2. Action a_i gives a payoff π_i^h with probability p_i, and $\pi_i^l = 0$ with probability $1 - p_i$. Assume $\pi_1^h > \pi_2^h$. The algorithm does not experiment (or the probability of experimentation decays exponentially, so that in the long-run it becomes null), and updates with a rule similar to (10), with $\alpha = 1$.

Because $\alpha = 1$, the Q-matrix can only take four values:

$$Q_1 = \begin{pmatrix} \pi_1^h \\ \pi_2^h \end{pmatrix}, Q_2 = \begin{pmatrix} 0 \\ \pi_2^h \end{pmatrix}, Q_3 = \begin{pmatrix} \pi_1^h \\ 0 \end{pmatrix}, Q_4 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{OA.2}$$

Given that $\pi_1^h > \pi_2^h$, when the Q-matrix is Q_2 the algorithm will play a_2. With probability p_2 the next value of the Q-matrix will be Q_2 again, and with probability $1-p_2$ it will be Q_4. Similarly, when the Q-matrix is Q_3 the algorithm will play a_1, then the next value will be Q_3 with probability p_1 and otherwise Q_4. When the Q-matrix is Q_4 the algorithm will play a_1 with probability $1/2$, leading to either Q_3 or Q_4, and a_2 with probability $1/2$, leading to either Q_2 or Q_4. Note that the only state of the Q-matrix that can lead to Q_1 is Q_1 itself, and only with a probability lower than 1. Hence, in the long-run the probability that the Q-matrix is Q_1 is zero.

The Q-matrix then follows a Markov process with 3 states Q_2, Q_3, and Q_4, and the transition probabilities just described. It is easy to compute the stationary probability of each state, and then the stationary probability that the algorithm plays a_1 is:

$$\Pr(a = a_1) = \frac{1-p_2}{2-p_1-p_2}. \tag{OA.3}$$

Now we can test how each procedure will estimate $\Pr(a = a_1)$. We take $p_1 = 0.1$ and $p_2 = 0.9$, which gives $\Pr(a = a_1) = 0.1$. In words, the algorithm will constantly alternate between a_1 and a_2, but in the long-run it will play a_1 10% of the time and a_2 90% of the time.

To implement the fixed stopping time procedure, we take $T = 50,000$. We simulate T periods for $K = 1,000$ experiments, and we record the percentage of experiments in which the algorithm plays a_1 or a_2 in the last episode.

To implement the random stopping time procedure, we let the algorithm run for 50,000 episodes, and then wait until the algorithm has played the same action for 100 episodes. We then stop the algorithm and record the action played in the last episode. We run $K = 1,000$ experiments and record the percentage of experiments in which the algorithm plays a_1 or a_2 in the last episode.

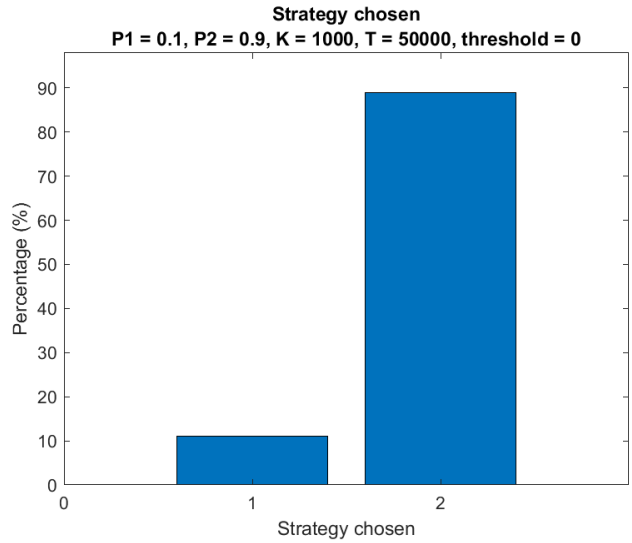
Figure OA.4 shows the outcome of our experiments. On Panel A we see that, using the fixed stopping time procedure, the percentage of experiments that end with action a_1 is very close to the theoretical value of 10%. On Panel B instead, with the random stopping time procedure the percentage of experiments that end with action a_1 is 0%, so that the estimate of $\Pr(a = a_1)$ is significantly biased downwards.

The reason for this bias is that the second procedure conditions the observation on having the

same action taken 100 times in a row. Conditionally on being in state Q_2 and playing a_2, the probability of remaining in Q_2 is 0.9. The probability to remain in Q_2 for 100 episodes in a row is $0.9^{100} \simeq 2.65 \times 10^{-5}$, so that on average it will take $1/(2.65 \times 10^{-5}) \simeq 37,648$ repetitions of a sequence of 100 episodes to observe a constant action. For action a_1, the probability of remaining in Q_3 is only 0.1, and the probability to remain in Q_3 for 100 episodes in a row is $0.1^{100} = 10^{-100}$, which is virtually zero. Hence, the random stopping time procedure picks up very particular histories, heavily biased towards action a_2.

This example is clearly extreme and meant only for illustration. With lower values of α and actions that are less different we do not expect the two procedures to lead to radically different results. However, given that the random stopping procedure is in principle biased and is also typically much more computationally intensive, we recommend using the fixed stopping time procedure instead.

Panel A: Fixed stopping time procedure.



Panel B: Random stopping time procedure.

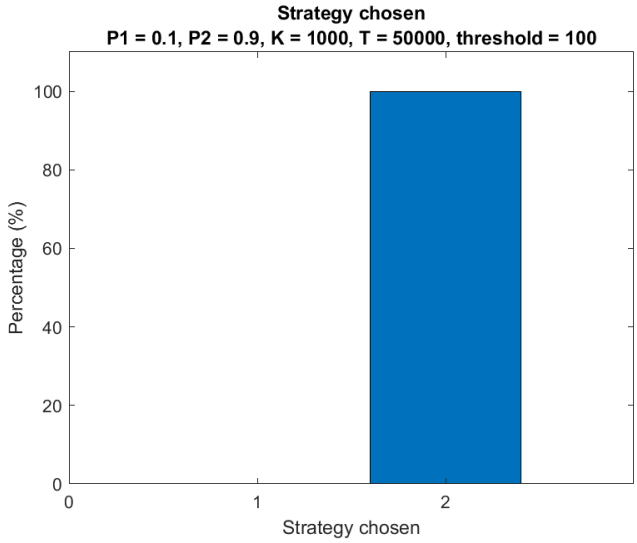


Figure OA.4: **Percentage of experiments ending with actions a_1 and a_2, using either the fixed stopping time procedure or the random stopping time procedure.**