

# Revisiting Agent-Based Models of Algorithmic Trading Strategies

Natalia Ponomareva and Anisoara Calinescu

University of Oxford, UK

natalia\_ponomareva@yahoo.com, Ani.Calinescu@cs.ox.ac.uk

**Abstract.** Algorithmic trading (AT) strategies aim at executing large orders discretely, in order to minimize the order’s impact, whilst also hiding the traders’ intentions. Most AT evaluation methods range from running the AT strategies against historical data (backtesting) to evaluating them on simulated markets. The contribution of the work presented in this paper is twofold. First we investigated different types of agent-based market simulations and suggested how to identify the most suitable market simulation type, based on the specific market model to be investigated. Then we proposed an extended model of the Bayesian execution strategy. We implemented and assessed this model using our tool Al-TraSimBa (ALgorithmic TRAding SIMulation BACKtesting) against the standard Bayesian execution strategy and naïve execution strategies, for momentum, random and noise markets, as well as against historical data. On the basis of the results presented in the paper, we propose that *momentum market* is the most suitable model for testing algorithmic trading strategies, since it quickly fills the Limit Order book and produces results comparable to those of a liquid stock. Our experiments and analysis also revealed that: (i) the method of estimating priors proposed in this paper – within the Bayesian adaptive agent model – can be advantageous in relatively stable markets, when trading patterns in consecutive days are strongly correlated, and (ii) there exists a trade-off between the frequency of decision making and more complex decision criteria, on one side, and the negative outcome of lost trading on the agents’ side due to them not participating actively in the market for some of the execution steps.

**Keywords:** Algorithmic Trading; Bayesian adaptive agents; simulation; backtesting

## 1 Introduction

Algorithmic trading (AT) strategies, which model decisions made by the algorithms as opposed to human actors [28], have gained significant strength in the recent years [20]. According to [20], “for large stocks AT narrows spreads, reduces adverse selection, and reduces trade-related price discovery. Their findings indicate that AT improves liquidity and enhances the informativeness of quotes”. Similarly, [15] found AT to be a cost-effective technique for large orders.

Market activity is comprised of the execution of orders submitted by the market participants, and the most common types of orders are *market orders* and *limit orders* [31]. A *market order* is the order to be executed immediately, at the best available price on the market. A *Limit order* is defined by additionally providing a limit (or activation) price, *i.e.*, the price which, once reached by the market, will activate the order and lead to its execution. The limit price for the *Sell* order is above the current price, and for the *Buy* order it is below the market price.

Execution trading algorithms, which we will focus on, aim to execute an order submitted by a human, such as to minimize both the associated transaction cost and the market impact of the order. The two most common approaches for evaluating trading strategies are *backtesting* [21, 28] and *evaluation via artificial market simulation* [10, 21, 24]. In *backtesting* the algorithms are evaluated using historical market data [21, 28]. For each discrete time point  $t$ , the agents have access to historical data on the current price at time  $t$ . The agents submit two different types of orders: *market orders* and *limit orders*. Market orders are executed at the current price. Limit orders are either executed at the current price, if their limit price allows immediate execution, or are saved into the *Limit Book*. Then time is incremented,  $(t + 1)$ , the next current price is read from historical data, and the agents repeat the previous steps. This iterative process continues until all historical data items are processed. Backtesting of AT strategies is acceptable under the assumption that the agents' orders have little or no market impact, and would not have changed the historical prices if they were executed within the real market.

In *agent-based simulations* the agents representing humans and AT strategies compete with each other and determine the price of the stocks [24, 21, 10]. Since price fluctuations depend on the interactions of all the agents, and additional conditions that did not take place in historical data can occur, trading strategies can be arguably more fully evaluated by this approach than in the backtesting model. Bonabeau [10] state that an agent-based model that simulates the effect of regulatory changes on markets, under various market conditions and for different agent capabilities and strategies, has been used by NASDAQ to evaluate the impact of tick-size reduction. This model revealed a surprising behaviour, such as the fact that a reduction in the market tick's size can reduce the market's ability to perform price discovery, thus leading to an increase in the bid-ask spread.

The contribution of the work presented in this paper is twofold. First we investigated different types of agent-based market simulations and suggested how to identify the most suitable market simulation type, based on the specific market model to be investigated. Then we proposed an extended model of the Bayesian execution strategy. We implemented and assessed this model using our AlTraSimBa (ALgorithmic TRAding SIMulation BACKtesting) tool against the standard Bayesian execution strategy and naïve execution strategies, for momentum, random and noise markets, as well as against historical data. The results revealed valuable insights on the trade-offs between the frequency of

decision making and more complex decision criteria, on one side, and the negative outcome of lost trading on the agents' side due to them not participating actively in the market for some of the execution steps.

The remainder of this paper is organised as follows. Section 2 reviews agent-based simulation models within an AT context. Section 3 describes architectural challenges to consider and summarizes some of the features of our tool. Section 4 reviews algorithmic execution agents, performance metrics, the standard adaptive Bayesian execution algorithm, and presents the theoretical extension we proposed to this approach. Section 5 presents and analyses our results and assesses our approach. The last section summarises and concludes the paper, and suggests future work directions.

## 2 Agent-based models of financial markets

In order to build a realistic market model and simulation, one must decide the algorithm which will model how the price of the stocks will vary over time, and the type of agents that will participate in the market. Once these modelling decisions have been made, the model implementation and architectural decisions pose several challenges, too. This section reviews the different features that are relevant for agent-based modelling and simulation, within a financial market context.

### 2.1 Continuous Double Auction

Price determination specifies how the prices of securities change within a market context where supply and demand do not match each other [24]. The most realistic way to simulate real markets is the Continuous Double Auction (CDA) market [24, 33], where the agents are allowed to submit any type of orders (*Buy* or *Sell*), at any time.

CDA maintains an *Order Book* containing the list of the submitted limit orders. For each type of order (*Buy* and *Sell*) there is a separate queue. In the *Buy* queue the orders are sorted in decreasing order of limit price; if several limit orders have the same limit price, they are stored using a FIFO criterion. The *Sell* queue orders are sorted in increasing order of both limit price and their arrival time. The first order in the *Buy* queue determines *the bid price*, *i.e.* the highest price at which buyers are willing to purchase securities. The first order in the *Sell* queue dictates *the ask price* – the lowest price which sellers are willing to accept [23, 33].

A submitted market order is executed against the *Limit Order Book* of an opposite direction. If the market order is larger than the limit order at the head of the queue, the next limit order from the queue is retrieved. Similarly, if a limit order arrives, it is first checked against a queue of opposite direction, to see if its immediate execution is possible, *i.e.*, if a buy limit order has a limit price more or equal to the current ask price, or if a sell limit order's price is less or equal to the current bid price. If a limit order cannot be executed yet, it is saved into

the queue of the order's direction. Due to the nature of this mechanism, *the bid price is always lower than the ask price*.

CDA assumes order divisibility, *i.e.*, an order can be executed partially – any agent willing to buy (sell) stocks agrees to have only part of the order executed, if necessary [33].

## 2.2 Human-like agents

Human-like agents are used to model human actors on the market, to generate trading volume and liquidity, and fill the limit order book - *e.g.* initialize the simulation for testing AT strategies. These agents need to generate order flows comparable to those in real financial markets. For this purpose, four main human-like agent types are proposed in the literature: *noise agents*, *random trading agents*, *momentum trading agents* and *contrary trading agents* [21].

**Noise agents (zero intelligence)** Many researchers choose to use zero intelligence agents in their financial market simulations, for example in the models presented in [12, 13, 18, 28]. There is a strong empirical evidence backing up such a widespread use of noise agents - [18] has shown that noise agents with parameters derived from empirical data can explain up to 96% of the variance of spreads and 76% of the diffusion rates. There are different variations of zero-intelligence agents, the simplest ones ranging from submitting buy or sell and market or limit orders with equal probabilities, to extended models like those in [14].

Noise agents represent uninformed traders on the market. There exist a number of noise agent variations. Relatively complicated strategies for noise traders are used in [12–14]. In these papers, the agents can submit *Buy* and *Sell* orders with equal probabilities, and can submit *Limit* and *Market* orders, or can cancel existing an limit order if it was not yet executed. For limit orders, limit prices can lie inside the current spread, outside the current spread, or they can be equal to the current spot price. The inside-the-current-spread limit price is derived using the uniform distribution, and the outside-the-current-spread limit price is derived using the power law. The size of the market and limit orders follow power law distributions, with different parameters. The corresponding set of parameters suggested by [13] is presented in Table 1.

Equation 1 illustrates how a random value can be obtained from the power law distribution, with parameters  $A$  and  $\alpha$ .

$$y = A/x^{\frac{1}{\alpha-1}}, \quad x \sim \text{uniform}(0, 1) \quad (1)$$

**Random trading agents** Random trading agents can submit either a *Sell* or a *Buy* order, each with probability 0.5 [21]. In contrast with noise agents, random agents never submit market orders, instead they always submit limit orders for a fixed number of shares. The random trading agent's logic is summarized in Table 2, where  $\alpha_t$  is a random variable  $\sim \text{uniform}(-0.1, 0.1)$  and  $p_{t-1}$  is the previous price of the security [21].

**Table 1.** Noise agent parameterization

Parameter	Value
Order cancellation probability	0.5
Market order probability	0.16
Market order size	$\sim \text{powerlaw}(10, 2.7)$
Limit order probability	0.35
Limit order in spread probability	0.32
Limit order at spot price probability	0.33
Limit order outside the spread probability	0.35
Limit price inside the spread	$\sim \text{uniform}(\text{bid}, \text{ask})$
Limit price outside the spread	$\sim \text{powerlaw}(50, 2.5)$
Limit order size	$\sim \text{powerlaw}(10, 2.1)$

**Table 2.** Random trading agents parameterization [21]

Order type	Action
Buy order	Submit Limit Buy order with quantity $N$ and limit price $(1 + \alpha_t)p_{t-1}$
Sell order	Submit Limit Sell order with quantity $N$ and limit price $(1 + \alpha_t)p_{t-1}$

**Momentum trading agents** Momentum trading agents are trend followers – they assume the persistence of an observed (so far) trend and act based *on this belief* [21]. For example, if the stock price went up compared to the previous observations, momentum agents buy this stock, assuming that it will continue to appreciate in price.

**Contrary trading agents** Contrary agents have an opposite opinion and assume that a trend is just a temporary inconsistency in the market, perhaps an overreaction to a new announcement, and thus act *against* the trend [21]. For example, they choose to sell stocks if there was an upwards trend, and buy if there was a downwards price trend. The strategies of momentum and contrary trading agents are presented in detail in [21], and are summarised in Table 3.

**Table 3.** Momentum and contrary agents parameterization [21]

Condition	Action
<i>Momentum agent</i>	
$p_{t-1} < p_t$	Submit Limit Buy order with quantity $N$ and limit price $(1 + \alpha_t)p_{t-1}$
$p_{t-1} > p_t$	Submit Limit Sell order with quantity $N$ and limit price $(1 + \alpha_t)p_{t-1}$
<i>Contrary agent</i>	
$p_{t-1} < p_t$	Submit Limit SELL order with quantity $N$ and limit price $(1 + \alpha_t)p_{t-1}$
$p_{t-1} > p_t$	Submit Limit BUY order with quantity $N$ and limit price $(1 + \alpha_t)p_{t-1}$

### 3 Implementation

#### 3.1 Essential features of the AT simulation tool

Once the features and parameters of the model are decided, the next important question that arises is what features should the simulation tool include, and how it should be implemented. Based on the analysis and evaluation of several financial market simulation tools [12, 19, 21, 29, 32] and available agent-based platforms like SWARM [6] and JASA [4], we identified the following list of features that a suitable tool for evaluating AT strategies should have:

- The ability to switch between backtest mode and CDA mode, like in [28], thus rendering the AT simulation tool suitable for both testing the performance of AT strategies against historical data, as well as allowing to explore how the agents and algorithms will behave and what patterns might emerge from the agent interactions for CDA-based simulations.
- The ability to simulate real market features – while this requirement adds complexity, these features might be essential to understand what strategy will perform best in a real market context. They include:
  - *Short selling*. Short selling is essential because allowing algorithms to model short trading can substantially extend their power, and thus more complicated algorithmic agents can be evaluated.
  - *Commissions* are also a prerequisite, because modelling them can have a substantial impact on the agents' performance.
  - *Lower latency* - as AT strategies can process information, make decisions and execute orders considerably faster than humans.
- Finally, as suggested in [32], the system should be of a pluggable architecture and easily extensible, to allow the addition of new agents and new modules.

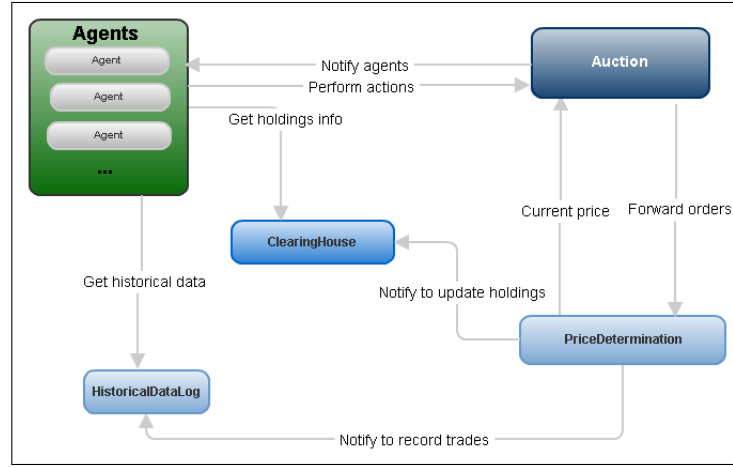
An important part of implementation is to decide how the agents will behave during simulations. Many implementations, like [19], choose an approach where at each point of time only one agent, selected randomly, will be able to perform any market actions. This is usually simulated by selecting an agent number from a distribution, and then giving this agent opportunity to act. This, however, makes it complicated to simulate the lower latency of algorithmic agents as, for this feature to be simulated, algorithmic agents should have a higher probability of being selected.

We, however, chose to adopt a different approach: at each time all agents can act simultaneously. In the implementation of our simulation tool, each agent is represented by a thread and can make decisions and submit orders at each auction time moment (*price tick*). Agents are synchronized only when they are submitting orders, because it is impossible to submit two orders simultaneously and there is always an order submitted earlier. When the next tick occurs, all agents are notified and can choose either to skip the step or act at this time. The probability of skipping the step should be higher for human-based agents than the probability of skipping the step for algorithmic agents, thus simulating the lower latency.

In the next section we briefly describe the architecture we used for our simulation tool, AlTraSimBa, in terms of its modules and how they are linked, and we explain the reasoning behind these design and implementation decisions.

### 3.2 AlTraSimBa implementation

There are two separate components of AlTraSimBa: a core component and a UI component, in charge of graphical user interface. In the core part the main modules are Auction, PriceDetermination, ClearingHouse and HistoricalDataLog. Figure 1 demonstrates the schematic view of the modules' interaction.



**Fig. 1.** Modules of AlTraSimBa system

**Auction** is the core module in charge of time iteration, notification of agents about auction events (tick time passed, auction day passed, etc.), collection of orders and directing them to the appropriate price determination module (either CDA or BackTest). It is implemented as a separate thread that communicates with the Agent threads via observer-like communication (all agents are listeners of auction events).

**PriceDetermination** is in charge of price formation and models the market mechanism. For CDA, price changes are determined by the executed orders on the market, while for BackTest prices follow prefixed fluctuations from the loaded data file. This module executes market orders, maintains the *Limit order book*, and executes limit orders when limit price conditions are met. The PriceDetermination module is connected to the ClearingHouse module, to be able to inform the latter that the order of an agent was executed, and therefore that the agent's holdings should be updated. Additionally, this module notifies HistoricalDataLog when trading takes place and forwards the details of the trade to be recorded.

**HistoricalDataLog** contains the information about the trades that took place on the market in the past. It also makes historical price movements and trading volumes available to all the agents in the model, for training and decision-making purposes.

**ClearingHouse** is in charge of keeping agent-specific information: the current holdings of an agent (the number of stocks owned, free cash and level of credit, number of shares shortsold), initial wealth level, and the list of submitted, cancelled, executed or still outstanding orders.

Even though we tried to implement as many key features as possible, due to the limited time AlTraSimBa suffers from a number of shortcomings which should be addressed in the future work. These limitations are:

- Borrowing is interest-free in our implementation. Ideally interest should be added for each period of having negative balance, however it requires maintaining a history of holdings and applying interest accordingly. This real market feature is left for later implementations.
- Short selling does not incur commissions. By contrast, in the real world short selling requires to have a margin account with holdings that should not fall below a certain level of the price of all borrowed securities, otherwise a margin call is encountered [31]. In the future versions of the tool interest and margin calls should be implemented.
- The AlTraSimBa agents can so far submit market and limit orders only, other types of real orders are not implemented. Adding agents who can submit, for example, stop and stop limit orders can result in more sophisticated and potentially profitable strategies.

Even though the AlTraSimBa simulation tool does not incorporate all real market features, it nevertheless provides a powerful and flexible functionality for the complex evaluation of different algorithmic trading strategies, on different types of markets, and can therefore be valuable for both researchers and practitioners.

## 4 Execution Algorithmic Agents

In this section we explain why execution strategies are important, review metrics used to assess these strategies, and list the main relevant execution strategies. Finally, we summarize the Bayesian execution agent with daily cycle strategy and describe our theoretical extension to this strategy.

### 4.1 Background

An order of almost any size can be executed via the market order. However, a large market order will consume limit orders from the limit order book, step by step, from the top of the queue, obtaining worse prices as deeper limit orders are executed to satisfy the market order. Execution algorithms aim at executing large orders in such a way as to minimize the market impact of the order.



They generally work by slicing large orders into smaller orders (suborders) and submitting the latter to the market at discrete times.

The problem of an optimized trade execution can be formulated as follows: the algorithm is supposed to buy or sell a specified number of shares of a specified stock, over a predetermined horizon, while taking into account the execution goals. Execution goals can be maximizing the revenue (for *Sell* orders) or minimizing the cost (for *Buy* orders), or aim to meet other investor performance metrics (*e.g.*, speedy execution, minimization of market impact). The trading strategy controls decisions about how to best split the order into smaller orders, the timing for submission of each suborder, what type of order to use, and how aggressively priced should the suborders be if limit orders are used [12].

Execution strategies can be divided into strategies that aim to meet market benchmarks and strategies aiming to meet criteria imposed by the order itself, at the moment of order submission [19].

#### 4.2 Execution strategy evaluation

The objectives of an execution strategy dictate the performance metrics to be used for its evaluation. The algorithms focused on meeting market benchmarks could be assessed using the *Volume Weighted Average Price* (*VWAP*) metric [13, 23].

The *Volume Weighted Average Price* (*VWAP*) is the average price on a per share basis over a specified period  $H$ , for example over one trading day [13, 23]. If during the horizon  $H$ ,  $T$  trades took place on the market  $\{(p_1, v_1), \dots, (p_T, v_T)\}$ , where  $p_i$  denotes the price at which the trade was executed and  $v_i$  the traded volume, then the market  $VWAP_M$  is defined as:

$$VWAP_M = \sum_{t=1}^T p_t v_t / \sum_{t=1}^T v_t \quad (2)$$

The *VWAP* of some algorithm  $A$  is defined analogously using the sequence of orders algorithm executed. To compare the market and algorithm's metrics, the *VWAP Ratio* (*VWAPR*) is used [23]:

$$VWAPR = \begin{cases} \frac{10000 \times (VWAP_A - VWAP_M)}{VWAP_M} & \text{for a } \textit{Buy} \text{ order} \\ \frac{10000 \times (VWAP_M - VWAP_A)}{VWAP_M} & \text{for a } \textit{Sell} \text{ order} \end{cases} \quad (3)$$

The smaller the *VWAPR*, the better is considered the strategy. A negative *VWAPR* means that the  $VWAP_A$  was better than  $VWAP_M$  [23]. Apart from just the *VWAPR*, [13] also considers the algorithm's *VWAPR* mean and standard deviation over  $N$  trading days. Low *VWAPR* mean and standard deviation values are expected from a good execution strategy aimed at achieving the *VWAP* metric.

Another metric, the *Mid-Spread Metric* (*MSM*) is calculated by comparing the overall execution price achieved by an execution strategy to the average of

the bid-ask spread at the time when the order was submitted to the strategy for execution [26]. *MSM* provides a good baseline for the comparison of different execution algorithms that do not aim at achieving market-specific metrics. In general, any real strategy is expected to do worse than this idealized policy (*MSM* is also referred to as *underperformance* or *trading cost*) [25], unless market prices will move in a direction that favours the trader. *MSM* is measured in basis points (1/100 unit of percent); the *MSM* equation is given below:

$$MSM = \begin{cases} \frac{10^4 * (\sum_{t=1}^K p_t n_t + C(K-1) - p_{mid} * X)}{p_{mid} * X} & \text{for a Buy order} \\ \frac{10^4 * (p_{mid} * X - \sum_{t=1}^K p_t n_t - C(K-1))}{p_{mid} * X} & \text{for a Sell order} \end{cases} \quad (4)$$

where  $\{(p_1, n_1), (p_2, n_2), \dots, (p_K, n_K)\}$  is the sequence of suborders executed by the strategy,  $p_{mid}$  is equal to  $(p_{ask} - p_{bid})/2$ ,  $X = \sum_{t=1}^K n_t$  is the volume of original (large) order, and  $C \geq 0$  is the commission size per order.

If during a strategy execution the suborders submitted by the strategy exhaust all limit orders offering to buy (or sell) at the spot price, the spot price will go down for *Sell* orders and up for *Buy* orders, and thus the *MSM* value will be positive. If the trader's suborders are executed with a price below the current mid-spread price for *Buy* and higher for *Sell* orders, then the *MSM* value will be negative and that strategy will be considered successful [26].

### 4.3 Naïve strategies

To estimate the performance of complex execution strategies, the main execution strategy types, such as those described in [13, 25], can be used. Below we briefly summarize these strategies; for full details refer to [12, 13, 25, 26].

- Strategies that submit only market orders:
  - Market Order Agent – submits 10 equal suborders at equal time intervals;
  - VWAP Agent – submits suborders at each tick, with  $size = fullOrderSize/N$ , where  $N$  is the number of tick updates received during the execution horizon  $H$ .
- Strategies that use limit orders:
  - Naïve Limit Order Agent – submits one limit order for the full quantity and amends the limit price to the current spot price at each tick;
  - Chunked Limit Order Agent – divides each order into 10 equal suborders, submits each chunk with equal intervals, and returns to chunks and amends them after a fixed time interval (less than the interval between the chunks' submissions).

### 4.4 Bayesian adaptive trading agents

**Theory** Bayesian adaptive trading with a daily cycle is a dynamic strategy for executing large orders [7]. This strategy is adjusted at each arrival of a price update (at each *price tick*). The derivation of the Bayesian-based strategy is based on the following assumptions [7]:

- (i) The stock's price can be described as an arithmetic random walk:  $S(t) = S_0 + \alpha t + \sigma B(t)$ , where  $S(t)$  is the price of the stock at time  $t$ ,  $S_0$  is the price of the stock at the beginning of the day,  $\alpha$  is a drift, thought to be determined by the actions of the informed traders,  $\sigma$  is the market volatility, and  $B(t)$  models standard Brownian motion.

Almgren *et al.* believe that large investors decide on their trading strategy in the morning and thus they determine the overall direction of the price movements [7]. This approach is in contrast with many execution strategies which treat price fluctuations caused by other traders as noise in the data [7]. The drift rate is assumed to be constant throughout the whole day, however, its value is unknown and has to be estimated via observations of the price movements. This assumption is supported by another assumption, that informed traders use strategies that have *VWAP* as their benchmark. The volatility  $\sigma$  is determined from the actions of informed traders, as their trades are of a noise type and thus can be predicted on average [7]. Almgren *et al.* suggest to estimate  $\sigma$  from the short period of observations of price movements [7].

- (ii) The true drift value is unknown, but it is assumed to be drawn from the Normal distribution.

$$\alpha \sim \mathcal{N}(\bar{\alpha}, \nu^2) \quad (5)$$

This drift estimate will be updated as more stock price updates arrive during the day. According to Almgren *et al.* [7], the best estimate of  $\alpha$  at time  $t$  is given by:

$$\alpha_*(t) = \frac{\bar{\alpha}\sigma^2 + \nu^2(S(t) - S_0)}{\sigma^2 + \nu^2 t} \quad (6)$$

- (iii) It is assumed that the market impact of order execution can be described as a linear function of the execution speed.

The formulation of the problem is as follows:  $X$  represents the number of shares to be bought by the end of the trading horizon  $H$ ,  $x(t)$  represents the number of shares left to be bought at time  $t$  and is called *trading trajectory*. The trading rate (or speed of execution),  $v(t)$ , is defined as:

$$v(t) = -\frac{dx}{dt}; \quad \forall t \quad v(t) \geq 0 \quad (7)$$

The requirement of non-negativity means that the strategy should never sell to execute a *Buy* order.

Using assumption (iii), the price after the execution of a part of the order can be defined by:

$$\bar{S}(t) = S(t) + \eta v(t) \quad (8)$$

where  $\bar{S}(t)$  is the price of the stock after executing the strategy at moment  $t$ ,  $S(t)$  is the initial price of the stock at moment  $t$ , before any suborder was executed as a part of the strategy, and  $\eta$  is the market impact coefficient. For *Buy* orders  $\eta \geq 0$ , as every *Buy* order can only either drive the price up or leave it at the

same level (according to the *Limit Order Book* mechanism). Taking into account Equation (7) and adding the requirement that the trading rate  $v(t)$  should be continuous, Almgren *et al.* [7] derived equation (9) for  $x(\tau)$ , where  $t \leq \tau \leq H$ :

$$x(\tau) = \begin{cases} \begin{cases} X & \tau < t^* \\ \frac{H-\tau}{H-t}x(t) - \frac{\alpha_*}{4\eta}(\tau-t)(H-\tau) & \tau \geq t^* \end{cases} & \alpha_* < -\alpha_c \\ \frac{H-\tau}{H-t}x(t) - \frac{\alpha_*}{4\eta}(\tau-t)(H-\tau) & |\alpha_*| \leq \alpha_c \\ \begin{cases} \frac{H^*-\tau}{H^*-t}x(t) - \frac{\alpha_*}{4\eta}(\tau-t)(H^*-\tau) & \tau \leq H^* \\ 0 & \tau > H^* \end{cases} & \alpha_* > \alpha_c \end{cases} \quad (9)$$

$$t^* = H - \sqrt{\frac{4\eta x(t)}{-\alpha_*}}; \quad H^* = t + \sqrt{\frac{4\eta x(t)}{\alpha_*}} \quad (10)$$

$\alpha_c$  is the critical value of the drift, estimated using  $\alpha_c = 4\eta x(t)/(H-t)^2$ . Overall, the above equations mean that, if  $\alpha_* < -\alpha_c$ , the trading does not start until the  $t^*$  period. If  $\alpha_* > \alpha_c$ , then the trading finishes before the end period  $H$  – the last trade is expected to occur no later than at time  $H^*$ . For exact details of the derivations refer to [7].

**Extending the standard Bayesian agent approach** In this section we propose three theoretical extensions to the standard Bayesian adaptive trading model presented in the previous section. We assume that trades will happen at discrete points of time  $\{1, 2, \dots, H\}$ , time 0 is the beginning of the algorithm, and  $x(0) = X$ . The trade should be executed during one business day, so:

$$H = \frac{\text{End of market day} - \text{beginning of market day}}{\text{tick step}} - 1 \quad (11)$$

where *tick step* indicates the time interval between two consecutive price updates (*e.g.* every minute or 10 minutes).

- (i) The maximum number of orders an agent can submit is  $H$ . [25] mentions that the execution strategy may not consider the commissions, as execution strategies are usually employed by large institutional investors who receive considerably lower commission rates compared to the regular investors. However, it still seems appropriate to include the commissions into the calculation of the optimal strategy. If the algorithm trades too often, the commission incurred can be too high, as it is paid for every order executed. We can select  $H$  in such a way that the overall commission incurred for all orders will be no more than a certain percentage of the overall spot price of the order. We define the number of steps as:

$$\text{step} = \frac{H}{\min(\max(1, \maxStepsNum), H)} \quad (12)$$

where  $maxStepsNum = S_0 X * \epsilon / ComSize$ ,  $\epsilon$  is a commission threshold (e.g., 0.001) and  $ComSize$  is the size of the commission per order. Then our strategy will act only when  $t \bmod step \equiv 0$ .

- (ii) In [7] the schedule  $x(\tau)$  (Equation (9)) was derived for *Buy* orders. To adjust it to *Sell* orders, we analyse equation (8). Any *Buy* order can only worsen the spot price of a stock. For a *Sell* order, large orders will dive deeper into the *Limit Order Book*, and executing limit orders will gradually reduce prices the deeper the *Limit Order Book* is traversed, so the market impact equation becomes  $\bar{S}(t) = S(t) - \eta v(t)$ . Then the equation for the trading rate becomes:

$$x(\tau) = \frac{H - \tau}{H - t} x(t) + \frac{\alpha_*}{4\eta} (\tau - t)(H - \tau) \quad (13)$$

From here we can derive the *Sell* schedule  $x(\tau)$ :

$$x(\tau) = \begin{cases} \begin{cases} X & \tau < t^* \\ \frac{H - \tau}{H - t} x(t) + \frac{\alpha_*}{4\eta} (\tau - t)(H - \tau) & \tau \geq t^* \end{cases} & \alpha_* > \alpha_c \\ \frac{H - \tau}{H - t} x(t) + \frac{\alpha_*}{4\eta} (\tau - t)(H - \tau) & |\alpha_*| \leq \alpha_c \\ \begin{cases} \frac{H^* + \tau}{H^* - t} x(t) - \frac{\alpha_*}{4\eta} (\tau - t)(H^* - \tau) & \tau \leq H^* \\ 0 & \tau > H^* \end{cases} & \alpha_* < -\alpha_c \end{cases} \quad (14)$$

Further, by examining equations (9) and (14) we can notice, that if in (14) we replace  $\alpha^*$  to  $\alpha^{*'} = -\alpha^*$ , then (14) becomes exactly (9). This means that the only difference in our algorithm for the *Sell* order is in the calculation of the drift estimate:

$$\alpha_*(t) = \begin{cases} \frac{\bar{\alpha}\sigma^2 + \nu^2(S(t) - S_0)}{\sigma^2 + \nu^2 t} & \text{for a Buy order} \\ -\frac{\bar{\alpha}\sigma^2 + \nu^2(S(t) - S_0)}{\sigma^2 + \nu^2 t} & \text{for a Sell order} \end{cases} \quad (15)$$

- (iii) How do we set  $\bar{\alpha}$ ,  $\nu^2$  and  $\sigma$  for the current day in Equation (6)? According to [7], if a trader believes that his prior belief  $\bar{\alpha}$  is correct, then he needs to set the variance  $\nu$  to 0; in this case price updates will not modify the value of  $\alpha$ . If no reliable information about prior  $\alpha$  is available, setting variance  $\nu$  to  $\infty$  will result in the estimate of the drift  $\alpha$  to be formed only by price observations during the day. If we assume that the investors' overall drift for the previous day can be a good prior for the drift in the current day and that the absolute volatility  $\sigma$  is the same for both days, then the estimated value for the drift from the previous day can be evaluated using historical data for the previous day. Thus  $S(t) = S_0 + \alpha t + \sigma B(t)$

and  $S(t+1) = S_0 + \alpha(t+1) + \sigma B(t+1)$ . Subtracting one from another, we obtain:

$$S(t+1) - S(t) = \alpha + \sigma(B(t+1) - B(t)) \quad (16)$$

where  $S_0$  is the stock price at the beginning of the trading period on the previous day, and  $S(t)$  and  $S(t+1)$  are the stock prices at times  $t$  and  $t+1$ , respectively, on the previous day. Since  $B(t)$  models the Brownian motion,  $B(t+1) - B(t)$  is a random variable  $\sim \mathcal{N}(0, t+1-t) = \mathcal{N}(0, 1)$  and  $\sigma(B(t+1) - B(t))$  can be considered a random variable drawn from  $\sim \mathcal{N}(0, \sigma)$ . Now if we use the MLE estimate [9]:

$$0 = \frac{1}{N-1} \sum_{t=1}^{N-1} (S(t+1) - S(t) - \alpha) \quad (17)$$

$$\sigma = \frac{1}{N-1} \sum_{t=1}^{N-1} (S(t+1) - S(t) - \alpha)^2 \quad (18)$$

where  $N$  is the number of price observations and  $S(t)$  the stock price at time  $t$  on the previous day. From equation (17) we can derive:

$$\alpha = \frac{1}{N-1} \sum_{t=1}^{N-1} (S(t+1) - S(t)) \quad (19)$$

and then we can calculate  $\sigma$  using equation (18).

The assumption that the drift value from the previous day can be a good prior for the belief of the drift value of the current day means that *we believe that the intentions of the investors did not change much compared to the previous day*. The results of our experiments presented in Section 5 will show whether these assumptions lead to an improved performance of the trading algorithms, on backtesting and on market simulation.

## 5 Results

In this section we present the quantitative results and qualitative insights from the experiments we conducted with human-like and execution agent models implemented and assessed using the AlTraSimBa tool. Then we present the results of testing the Bayesian execution strategy and we compare it with naïve execution strategies. We also test whether the Bayesian agent assumptions we presented in Section 4.4 hold for backtesting and simulation.

### 5.1 Human-like agent experiments

In this simulation model, human-like agents will be required to fill in a Limit Order book and simulate trade flows corresponding to real market trade flows. To assess whether the human-like agents generate a realistic flow of orders, from

an economic perspective, we used two metrics frequently used in real markets – the *annualized daily returns volatility* and *bid-ask spread* [8, 17, 30].

Annualized daily returns volatility (*ADRV*) is a common metric used to report the volatility of different stocks [16] and is calculated using the following equations:

$$\text{Daily return} = 100 * \frac{\text{today closing price} - \text{yesterday closing price}}{\text{yesterday closing price}} \quad (20)$$

$$\text{ADRV} = \sqrt{252} * \text{standard deviation of daily returns for period} \quad (21)$$

$$\text{Spread} = 100 \frac{\text{ask price} - \text{bid price}}{\frac{\text{ask price} + \text{bid price}}{2}} \quad (22)$$

Reference [2] provides month-by-month statistics of annualized daily returns for 2893 stocks with different capitalizations. We downloaded these files and processed them in order to derive the "normal" levels of annualized volatility we would like our simulations to exhibit. The *ADRV* mean and standard deviation results we obtained for the 2010 data are illustrated in Table 4. This analysis showed that 95% of stocks had an annualized volatility between 1 and 80, and the interval of possible volatility values is [0.7250, 251.6504].

**Table 4.** Statistics of stocks ADRV for Year 2010

Volatility	#	%	Example of stocks
0.7250-13.9316	2000	7.9261%	[AON, CDNS, BCE]
13.9316-27.1382	7238	28.6847%	[GXP, DBD, FLIR]
27.1382-40.3448	6953	27.5552%	[BRF, IM, PPD]
40.3448-53.5514	4472	17.7228%	[WYNN, VNO, LEN]
53.5514-66.7580	2369	9.3885%	[AEIS, EXK, HMIN]
66.7580-79.9646	1096	4.3435%	[BGZ, GMO, MEL]
<b>Annualized volatility mean: 37.4928</b>			
<b>Annualized volatility S.D: 22.2068</b>			

Regarding the bid-ask spread, [22] suggests that for liquid stocks the average bid-ask spread is 2.0%, and [11] reports average bid-ask spreads on NASDAQ and NYSE being 1.0133% and 0.5773% respectively, with 95% of the stocks having spread of no more than 2.6486% for NASDAQ and no more than 1.4442% for NYSE. Thus, we concluded that, to validate our assumptions and modelling approach, our simulation experiments should produce price fluctuations with bid-ask spread no more than 3% of the stock's mid-quote price.

In all our human-agents experiments we used a tick step of 1 minute (every minute agents can submit new orders), a 6-hour trading day (from 13 to 19), a tick size of 1 cent; each market setting is run for 1 day to allow agents to fill the *Limit Order Book*. Simulations were run for a 10-day testing period, for which metrics are reported. We first calculated the average *Daily return* and

then converted it to the annualized volatility using equations (20) and (21). The *Spread* is calculated for all tick steps over the testing period using equation (22), and then the Annualized volatility mean and standard deviation are reported. The sizes of *Buy* and *Sell* queues are reported at the end of the testing period.

For each type of the market we ran several experiments with a different number of agents, to test whether a larger number of agents leads to more realistic results. For each market we reported *ADRV*, the bid-ask *Spread* mean and standard deviation, and the sizes of the *Buy* and *Sell* queues.

The conclusions of our analysis of the impact of different agent markets on the market validity are briefly presented next.

**Markets with random trading agents only** In addition to the markets described in [21] (contrary, momentum and half-and-half) we decided to also investigate markets where all the agents are random trading agents (described in Section 2.2). This market type can be used if the no intelligence assumption is sufficient for the purpose of the research, and the experimenter believes that stock price fluctuations follow a random pattern.

The results of experiments for markets containing 5, 11, 22, 55 and 110 agents are presented in Table 5.

**Table 5.** Random agent market Annualized Volatility (*AV*) and bid-ask *Spread*

# Agents	AV	Spread mean	Spread S.D.	Buy queue	Sell queue
5	168.4439	3.8802	2.3468	1788	1584
11	73.0448	4.1120	2.5591	3801	4025
22	138.6540	4.1075	2.4528	7431	7761
55	109.9939	3.9487	2.3586	17905	17855
110	139.5403	3.7866	2.3630	33376	33430

From Table 5 we can notice that random trading agents fill the *Limit Order Book* quickly enough – after 1 day neither the *Buy* nor the *Sell* queue are ever empty (otherwise the *Spread* reported would have been *NaN*). Additionally, they fill limit books almost equally.

Daily returns for a different number of random agents are shown in Figure 2. It appears that there is no pattern between the number of agents and daily returns fluctuations. That also explains why we cannot derive a pattern of *Annualized Volatility* – *AV* seems to be independent of the number of agents. However, as reported in Table 4, *AV* is within (not very common) real-market values. Overall, markets generated by random trading agents can represent a market where a very volatile stock is traded.

By analysing the results, we saw that the *Spread* values for the 11-agent markets and for the 110-agent markets are the most volatile, however we see no reasonable explanation for this phenomenon. This behaviour is most likely caused by the underlying random nature of the agents used. The results in Table



5 show that the overall *Spread* mean and standard deviation are unchanged when increasing the number of agents. Spread values are slightly larger than expected for liquid stocks, because, as stated previously, for liquid stock we would expect spread values to be less than 3%.

Overall, the market simulation with random agents seem to create an economically plausible situation, however slightly on the volatile side and more for illiquid stocks. The number of agents does not have a significant impact on the parameters of the market, and they should be chosen based only on volume (sizes of limit queues) a market should have.

**Momentum markets** [21] in their experiments used a market with 50 momentum agents and 5 random agents. We additionally tested a *Llimit Order Book* filled by 10, 20 and 100 momentum agents, while maintaining the proportion of momentum and random agents. The summary of the experimental results is presented in Table 6.

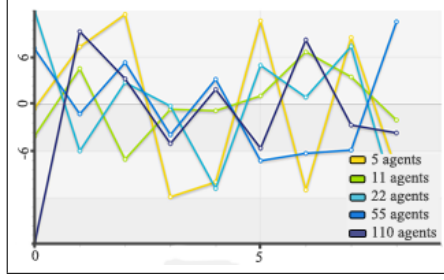
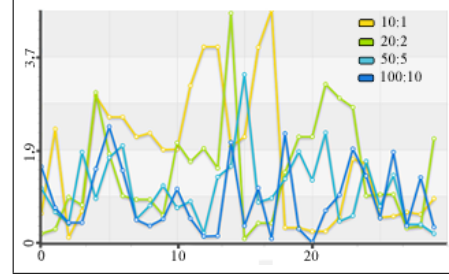
**Table 6.** Momentum market Annualized Volatility (*AV*) and bid-ask *Spread* (The # Agents column contains the number of momentum : random agents.)

# Agents	AV	Spread mean	Spread S.D	Buy queue	Sell queue
10:1	118.1072	1.6752	1.2114	6512	7303
20:2	103.7858	1.4024	1.0369	14320	13483
50:5	149.2243	1.0546	0.7812	33613	34058
100:10	94.3928	0.8391	0.6262	59964	69918

For the momentum markets we observe again no connection between the Annualized Volatility and the number of the agents in the simulation model, however both *Spread* mean value and standard deviation decrease with increasing the overall number of agents. The *AV* values are plausible and comparable with the corresponding random agent market values. However the *Spread* values are within the target spread values of 3%, and therefore suggest a more liquid stock. From Figure 3 we can see that, indeed, for the 100:10 configuration market the *Spread* fluctuations are less volatile than for the markets with fewer agents. Both limit books in the momentum markets seem to be filled equally.

**Contrary markets** Contrary markets according to [21] are the markets filled with contrary and random agents in proportion 10:1. The results we observed in experiments at the first sight seem perplexing (Table 7).

The values *NaN* in the columns for the bid-ask *Spread* standard deviation indicate that, during the testing period, one of the limit order books became empty. Indeed, we can see that the *Sell* queue is severely underfilled in comparison with the *Buy* queue. Additionally, the *AV* values indicate an extremely volatile stock, from an unrealistic (hopefully – unless another major crisis strikes) world. This

**Fig. 2.** Daily returns for random agents**Fig. 3.** Bid-ask spread for the last 30 mins of the momentum market**Table 7.** Contrary market Annualized Volatility (*AV*) and bid-ask *Spread*  
(The # Agents column contains the number of contrary : random agents.)

# Agents	AV	Spread mean	Spread S.D	Buy queue	Sell queue
10:1	46017.1713	NaN	NaN	0	14749
20:2	478285.9925	NaN	NaN	16	22448
50:5	2063384.608	NaN	NaN	0	3666
100:1	1529582.742	NaN	NaN	1669	24768

is in accordance with [21], which, via experiments, showed that the contrary agents' actions quickly reverse any trend and simulate very volatile markets.

Even if such agents are allowed more time to fill the *Limit Order Book*, the antisymmetry feature of limit order books filling will result in one queue being periodically exhausted. Overall, it seems that contrary markets are not suitable for testing algorithmic trading strategies.

**Half and half markets** Half and half markets are comprised of contrary, momentum and random agents in proportion 5:5:1, respectively [21].

**Table 8.** Half-and-half market Annualized Volatility (*AV*) and bid-ask *Spread*  
(The # Agents column contains the number of contrary : momentum : random agents)

# Agents	AV	Spread mean	Spread S.D	Buy queue	Sell queue
5:5:1	186.8334	4.3441	2.8278	3631	3847
10:10:2	260.533	7.1614	5.3655	7330	8087
25:25:5	236.0432	5.123	4.8293	16631	16636
50:50:10	187.67	3.5133	2.3815	32607	32855

The half-and-half market is more volatile than the momentum market, and the simulated stock is more illiquid (*Spread* of up to 7%, with a high standard deviation). There is no observed connection between the number of agents and the *AV* and *Spread* values; both queues are filled approximately equally. This

market type seems to be truly random, beating in randomness even the pure random agent market. Indeed, according to the experiments carried out in [21], the half-and-half market was the least stable market, with larger price fluctuations.

**Noise markets** [13] claims that, for noise agents with the parameters described in Table 1, the flow of orders generated is economically plausible and can be compared with the real market activity. Additionally, [18] has shown that zero intelligence agents with slightly different parameters could mimic real order flow in CDA simulations and are able to explain 96% of the variance for bid-ask price fluctuations. Thus we would expect that zero intelligence agents can create a flow of orders which will result in realistic price and bid-spread fluctuations. However, it still remains a question of how many agents should we include into the market simulation and how many days should they first run before their overall activity simulates a realistic market setting.

Nevertheless, running the experiments with parameters from Table 1 we found out that the market thus generated is extremely volatile, and 1 full day is not enough to fill the *Limit Order Book* – over a 10-day testing period one of the limit books was exhausted each day, resulting in a *NaN* bid-ask *Spread*. We ran a sequence of experiments to see whether we can find better parameters for the agents. During testing it turned out that an extremely high probability of order cancellation (and only limit orders can be cancelled) in conjunction with a high market order probability resulted in queues being emptied very quickly by the arriving market orders. We ran a series of experiments with different probabilities for market, limit and order cancellation (where all probabilities ranging from 0.3 up to 0.9 were tried, with a 0.05 step) and the following different parameters were found to give more plausible results (Table 9) (different values were obtained only for the market order, limit order and order cancellation probabilities; the amended values are in red).

**Table 9.** The modified noise agent parameterization

Parameter	Value
Order cancellation probability	0.05
Market order probability	0.3
Market order size	$\sim \text{powerlaw}(2.7)$
Limit order probability	0.65
Limit order in spread probability	0.32
Limit order at spot price probability	0.33
Limit order outside the spread probability	0.35
Limit price inside the spread	$\sim \text{uniform}(\text{bid}, \text{ask})$
Limit price outside the spread	$\sim \text{powerlaw}(2.5)$
Limit order size	$\sim \text{powerlaw}(2.1)$

The results of the set of experiments with the parameters in Table 9 are summarized in Table 10. Interestingly, if we extend the period agents are given

to fill the *Limit Order Book* before we test the market characteristics from 1 day to 5 days, the results obtained are economically more plausible, since annualized volatility is lower. Also, spread diminishes as more agents are used. This can be explained by more limit orders within the spread being entered, which diminishes the spread and increases the number of limit orders which offer to buy or sell at prices close to the spot prices, thus making incoming market orders execute at closer prices. Overall, noise agents seem to fill queues unequally, but as time elapses (or more agents are used), the spread and volatility diminishes.

**Table 10.** Noise markets Annualized Volatility (*AV*), bid-ask *Spread*, and *Buy* and *Sell* queues sizes. (Agents are given 1 and 5 days to fill the limit order book. The # Agents column contains the number of noise agents.)

# Agents	Annualized Volatility	Spread mean	Spread S.D	Buy	Sell
Agents are given <b>1</b> day before testing					
5	381.676	2.2934	6.5205	42423	408863
11	349.094	0.954	3.7674	65048	227519
22	492.48	0.1997	1.0485	223342	155818
55	27.8309	0.0339	0.3631	442579	851887
Agents are given <b>5</b> days before testing					
5	170.4793	0.6734	1.8316	55514	65256
11	161.0567	0.6092	1.7014	120106	109189
22	172.1393	0.4141	1.6055	211893	711861
55	64.3345	0.0836	0.5424	992644	922889

**Human-like agents markets conclusion** Based on the results and analysis presented in Section 5.1, we concluded that the momentum market is the best model for testing algorithmic trading strategies, since it quickly fills the *Limit Order Book* and produces results comparable to those of a liquid stock. On the other hand, random markets can simulate a more volatile and less liquid stock, whereas markets containing noise agents should be used with more time (5 and more days) given to fill the *Limit Order Book*, and they can represent a very liquid and volatile stock.

## 5.2 Bayesian adaptive learners

Execution strategies aim at minimizing the market impact of the order to be executed, therefore they should be mainly tested on the simulation part of the software, since the market impact of each order could thus be measured. Backtesting assumes infinite liquidity and zero impact of any order submitted by an artificial agent, but it also considers all agents evenly, which cannot be achieved with the market simulation, due to the random nature of the simulation of price movements. So backtesting can still be used to check whether a strategy adjusts its aggressiveness of order submission as a consequence of price variations.

**Experiment setup** At first we filled the *Limit Order* books with the human-like agents of a specified market type, and saved the configuration to file. Then we ran each test on this initial configuration, to ensure the tests have the same initial conditions. We thus eliminated the situation where one of the strategies is launched on a more favourable market. Each agent was then given to execute a large order and statistics were calculated. In theory, the starting price for the security should be the same, since the market configuration is loaded from the configuration file for each test. However, due to the random order in which agents act, initial spot prices can differ slightly.

Simulations were run for five different configurations for two market types: momentum markets and random agent markets. For each agent, we reported the *MSM* metric with commissions considered (Equation (4)), *VWAPR* (Equation (3)), *the achieved price* of an overall execution and *the ideal price* of an order – the price of an order at its creation time.

For each saved initial market configuration, we estimated the daily trading volume. [13] claims that execution strategies are used to execute the orders of size 5–15% of the Average Daily Volume (*ADV*). Orders less than 5% are believed to have a small market impact and can be executed either as market orders or using simple executional strategies. Orders bigger than 15% of *ADV* should be traded over several days.

For momentum markets tests, we used 50 momentum agents and 5 random agents, whereas for random markets we chose a configuration of 55 random agents. All markets were first run for 5 days before tests of different types of agents were launched.

Each Bayesian agent is initialised with two parameters: *Bayesian(considerCommissions, learnPriors)*. The parameter *considerCommissions* indicates whether commissions should be considered when calculating the trading schedule. When it is set to *true*, the Bayesian agent might do less trading, in an attempt to minimize the commission costs; *learnPriors* indicates whether the agent should try to estimate  $\sigma$  and  $\bar{\alpha}$  (Equation (5) and Section 4.4) from previous days observations.

The simulation parameters we used are described in Table 11. We will consider the results for each market type separately and then will draw global conclusions.

**Momentum market results** Table 12 summarizes the results for the momentum markets. We remind the reader that for both *MSM* and *VWAPR*, the lower their values, the better is considered the strategy. We can see that estimating prior values for  $\sigma$  and  $\bar{\alpha}$  for both *Buy* and *Sell* orders improves the *MSM* metric but decreases the *VWAPR* values. This indicates that estimating parameters from previous day observations provides a better starting point than using their default values. The improvement in *MSM* is more pronounced for the *Buy* order. The decrease in *VWAPR* is a side effect of the change in the trading speed, however the Bayesian strategy is not a VWAP-benchmark following strategy, since it bases its decisions only on sequences of price updates. Therefore poor

**Table 11.** Simulation parameters for Bayesian agent experiments  
(Parameters marked with \* are derived from previous day observations if *learnPriors* is *true*. Parameters  $\sigma$ ,  $\bar{\alpha}$  and  $\nu^2$  are used to estimate the drift, using Equation (15)).

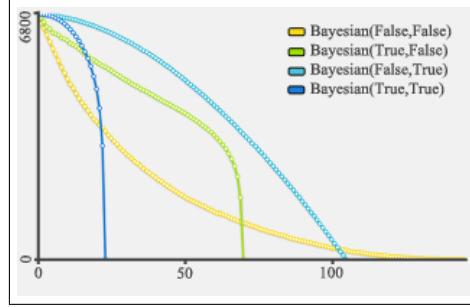
Parameter	Value
Commission	8 units per order
Commission threshold	0.001 of initial order price
Tick step (updates every)	1 minute
Market open time	13:00
Market close time	19:00
Default* Market Volatility $\sigma$	1.5
Default* prior estimate of drift $\bar{\alpha}$	0.7
Default prior STD of drift $\nu^2$	1.0

**Table 12.** Bayesian versus naïve execution strategies for momentum markets.  
(The parameters correspond to *Bayesian(considerCommissions, learnPriors)*).

Order	Agent	Ideal	Achieved	MSM	MSM Rank	VWAPR
Buy	Bayesian (false,false)	564131.06	584106.45	354.0912	4	-9.1662
	Bayesian (true,false)	564139.06	594657.66	540.9765	7	96.6631
	Bayesian (false,true)	564131.06	581117.8	301.1133	3	449.5647
	Bayesian (true,true)	564139.06	589819.53	455.2152	6	321.1232
	VWAP	564131.06	589342.46	446.9067	5	39.5673
	NaïveLimitOrder	564131.06	575703.94	205.1452	2	-293.1889
	ChunkedLimitOrder	564131.06	560338.74	-67.224	1	-0.2298
Sell	Bayesian (false,false)	599083.8	559206.91	665.6312	7	173.0038
	Bayesian (true,false)	604936.52	568520.81	601.9757	5	304.0506
	Bayesian (false,true)	604928.52	565012.03	659.8546	6	332.0176
	Bayesian (true,true)	604936.52	565265.26	655.7921	4	-27.1817
	VWAP	608685.84	611498.17	-46.2033	1	-1.5198
	NaïveLimitOrder	608685.84	604873.3	62.6355	2	281.1482
	ChunkedLimitOrder	608685.84	588004.18	339.7756	3	104.0207

*VWAPR* values in all types of Bayesian agents is an illustration that this strategy should not be used if a good *VWAPR* is needed. Not surprisingly, Bayesian agents receive the worst *VWAPR* in ranking when compared to naïve executional strategies (Table 12).

For *Sell* orders Bayesian agents (BA) that consider commissions and thus act less frequently than the standard BA improve the *MSM* only slightly, but deteriorate it for the *Buy* orders. This suggests that, even though such agents do less trading, they miss the important price changes updates that can result in an earlier adjustment of their execution strategies. This further suggests that perhaps setting a larger commission threshold may result in a bigger improvement of the BA performance, since they would receive more price updates and will have the opportunity to act more often. The trading trajectories (the sequence of the values of the number of shares an agent still has to buy or sell over time) for four Bayesian agents are presented in Figure 4.



**Fig. 4.** The trading trajectories for Bayesian agents, in momentum markets, for *Buy* orders. The  $x$ -axis is the number of price updates, and the  $y$ -axis is the number of shares the agents still have to buy at each moment of time. The parameters correspond to *Bayesian(considerCommissions, learnPriors)*. (The agents that consider commissions are given fewer opportunities to act, and thus they are allowed to execute orders in a lower number of steps than the no-commission agents.)

We can see that the smoothest trajectory is of the agent that does not consider commission and uses default prior values (*Bayesian(false,false)*). Its trading speed increases slightly towards the end, when the agent trades less aggressively, thus minimizing the possibility of exhausting limit orders with prices close to the spot price and obtaining worse prices. Two agents which consider commissions have to buy a large number of shares at the end, since they did not acquire them during the trading, and this ought to drive the price in negative for an agent direction. This suggests that perhaps setting a larger commissions threshold may result in a bigger improvement of the Bayesian agents performance since they will receive more price updates and will thus have opportunity to act more often. The agent that estimates prior from previous day observations seems to have more even liquidation speed in comparison with the agent that uses default values and trades more slowly at the beginning.

Chunked limit order and naïve limit order strategies outperform all Bayesian agents in terms of *MSM*, however this might be due to the fact that they engage in much less trading than even Bayesian agents that consider commissions. The trading trajectory for VWAP, naïve limit order and chunked limit order strategies are presented in Figure 5. The VWAP agent steadily submits orders for equal quantity at each price update it receives, whereas naïve limit order and chunked limit order agents execute orders much faster, thus they incur less commissions.

**Random market results** Random market results and ranking of the agents according to the *MSM* metric are presented in Table 13.

Estimating prior values in random markets considerably improves the *MSM* metric for the *Sell* orders, but deteriorates it slightly for the *Buy* orders. Since a random market is more volatile and less liquid than a momentum market, this suggests that estimates of priors based on previous days observations have less power in more volatile markets.

**Table 13.** Bayesian versus naïve execution strategies and their MSM rank, for random markets.(The parameters correspond to *Bayesian(considerCommissions, learnPriors)*).

Order	Agent	Ideal	Achieved	MSM	Rank	VWAPR
Buy	Bayesian (false,false)	365764.64	380432.2	401.0109	6	346.1609
	Bayesian (true,false)	365772.64	369826.34	110.8256	2	660.0392
	Bayesian (false,true)	365764.64	385983.38	552.7801	7	522.92
	Bayesian (true,true)	365772.64	380246.36	395.7026	5	660.8258
	VWAP	365764.64	375378.25	262.8359	4	318.1245
	NaïveLimitOrder	365764.64	370027	116.5329	3	228.4315
	ChunkedLimitOrder	365764.64	361184.24	-125.228	1	-10.9637
Sell	Bayesian (false,false)	430498.36	417768.65	295.697	7	313.3508
	Bayesian (true,false)	417752.08	434243.34	-394.76188	4	730.0739
	Bayesian (false,true)	417744.08	411427.53	151.2062	6	437.2886
	Bayesian (true,true)	417752.08	411954.97	138.7691	5	292.4836
	VWAP	434805	482559.75	-1098.3026	1	278.2538
	NaïveLimitOrder	434805	466496.36	-728.8637	3	227.9997
	ChunkedLimitOrder	434805	475926.36	-945.7425	2	-131.7396

For both *Sell* and *Buy* orders the Bayesian agents that consider commissions considerably improve the *MSM* metric in comparison with the Bayesian agents that trade more frequently. However, it would be wrong to attribute such an improvement in *MSM* only to the influence of the commission parameter, as the market is volatile and less frequent trading can by chance lead to better results.

For *Sell* orders we can see that a simple strategy such as *VWAP* achieved an extremely good performance by trading very often. Naïve Limit Order and Chunked Limit Order strategies, just as in the Momentum market, outperformed all Bayesian agents, apart from the Bayesian agent with commissions, which was able to do slightly better than NaïveLimitOrder agents for a *Buy* order (Table 13).

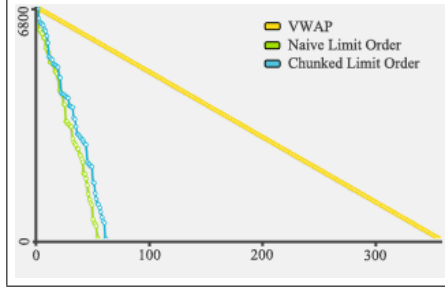
The trading trajectories for *Buy* orders are presented in Figure 6.

**Noise market results** Noise market results are summarized in Tables 14 and 15.

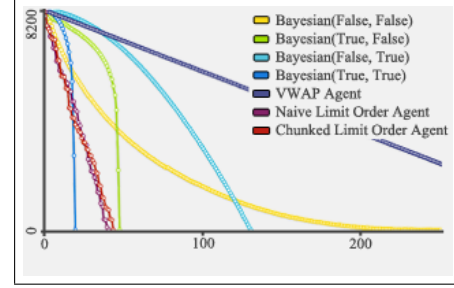
This market is the most volatile and represents the least liquid stock. For this market considering commission estimates improved the performance only for the *Buy* orders, and significantly deteriorated performance for *Sell* orders. Prior parameters estimated did not bring any improved performance, which is explainable, since in such market trends do not last long enough and thus previous days values have no bearing on the current day price fluctuations. Chunked limit order and naïve limit order strategy again outperformed all the types of Bayesian agents.

**BackTesting results** We performed a backtest for a very liquid stock Amazon (AMZN) [5] for the period between 29 June–06 July 2010, with a one-minute tick





**Fig. 5.** Trading trajectory for simple executional strategies in momentum market for *Buy* orders. The  $x$ -axis is the number of price updates, and the  $y$ -axis is the number of shares agents still have to buy.



**Fig. 6.** Trading trajectory for executional strategies in random market for *Buy* orders. The  $x$ -axis is the number of price updates and actions of the agent, and the  $y$ -axis is the number of shares agents still have to buy.

**Table 14.** Bayesian agent results in comparison with naïve execution strategies for noise markets.

(The parameters correspond to *Bayesian(considerCommissions, learnPriors)*).

Order	Agent	Ideal	Achieved	MSM	VWAP Ratio
Buy	Bayesian(false,false)	790132.3	832538.73	536.7003	119.1093
	Bayesian(true,false)	790140.3	808998.53	238.6693	498.9715
	Bayesian(false,true)	790132.3	848776.9	742.2124	654.0119
	Bayesian(true,true)	790140.3	832450.65	535.4789	786.9869
	VWAP	790132.3	818957.2	364.8111	292.8517
	NaïveLimitOrder	790132.3	805074.84	189.1144	-148.7392
	ChunkedLimitOrder	790132.3	793131.89	37.9631	-118.1761
Sell	Bayesian(false,false)	509783.85	510380.53	-11.7046	470.7504
	Bayesian(true,false)	503590.1	497954.73	111.9039	426.3716
	Bayesian(false,true)	503582.1	501832.83	34.7365	435.5821
	Bayesian(true,true)	503590.1	506548.04	-58.737	165.9824
	VWAP	541536.81	591678.12	-925.9076	164.197
	NaïveLimitOrder	541536.81	576314.39	-642.2015	44.4981
	ChunkedLimitOrder	541536.81	576979.55	-654.4844	-29.5074

**Table 15.** Ranking of agents by the *MSM* metric for the noise market

Order	Rank	Agent	MSM
Buy (8264)	1	ChunkedLimitOrder	37.9631
	2	NaïveLimitOrder	189.1144
	3	Bayesian(true,false)	238.6693
	4	VWAP	364.8111
	5	Bayesian(true,true)	535.4789
	6	Bayesian(false,false)	536.7003
	7	Bayesian(false,true)	742.2124
Sell (8282)	1	VWAP	-925.9076
	2	ChunkedLimitOrder	-654.4844
	3	NaïveLimitOrder	-642.2015
	4	Bayesian(true,true)	-58.737
	5	Bayesian(false,false)	-11.7046
	6	Bayesian(false,true)	34.7365
	7	Bayesian(true,false)	111.9039

step. We chose Amazon because it is actively traded and has large capitalization, thus it is possible that institutional investors will trade in this stock and that Almgren *et al.*'s assumption about the daily drift being formed by actions of the traders can hold for this data [7]. We can see in Figure 7 that during this period there were two general trends - a downward trend from 25th June to 2nd July, followed by an upward trend until 9th July. Since backtesting does not allow to replicate a market impact of a large order, the size of the order does not matter, so we gave the agents the task to *Buy/Sell* 5000 shares. The results for 5 trading days for *Buy* and *Sell* orders are summarized in Table 16.

**Fig. 7.** Price fluctuations for AMZN stock (*Picture taken from Google Finance* [3])

One of the problems in backtesting can be understood looking at the NaïveLimitOrder agent's performance indices. This agent submits a Limit Order for the full quantity at the current spot price. In real markets, if there is sufficient liquidity, an order is executed at the current spot price. However usually a Limit Order is only partially executed and the spot price goes down — for *Sell* orders, and up — for *Buy* orders. The remaining part of that Limit Order is amended

**Table 16.** Backtesting: Bayesian agent versus naïve execution strategies for *Buy* (top) and *Sell* (bottom) orders

$E_1$  and  $SD_1$  are the expectation and the standard deviation of  $MSM$ , respectively;  $E_2$  and  $SD_2$  are the expectation and the standard deviation of  $VWAPR$ , respectively.

Agent	29/6	30/6	1/7	2/7	5/7	$E_1$	$SD_1$	$E_2$	$SD_2$
Bayesian(false,false)	-9.75	202.88	-1.27	-115.97	-0.04	15.16	103.49	15.28	86.04
Bayesian(true,false)	-8.45	258.94	86.02	-150.45	12.06	39.62	133.75	29.53	69.36
Bayesian(false,true)	-9.75	279.89	100.22	-167.26	-0.04	40.61	147.11	40.03	66.52
Bayesian(true,true)	-8.45	257.72	118.67	-169.77	12.06	42.04	141.91	33.30	68.92
VWAP	-9.84	273.91	32.15	-144.35	0.02	30.37	135.93	30.02	74.75
NaïveLimitOrder	0	0	0	0	0	0	0	1.43	151.87
ChunkedLimitOrder	-9.84	258.45	25.49	-134.51	109	49.71	130.50	27.67	75.79
Bayesian(false,false)	9.93	-264.39	-47.80	155.82	-0.03	-29.29	135.85	-28.91	72.32
Bayesian(true,false)	8.63	-227.31	-105.28	162.86	-9.62	-34.14	129.32	-26.12	71.21
Bayesian(false,true)	9.93	-223.08	52.56	93.77	-0.03	-13.36	110.02	-13.54	100.02
Bayesian(true,true)	8.63	-230.64	-63.87	138.38	-9.62	-31.42	119.71	-22.55	73.36
VWAP	9.84	-273.91	-32.15	144.35	-0.02	-30.38	135.93	30.02	74.75
NaïveLimitOrder	0	0	0	0	0	0	0	1.435	151.87
ChunkedLimitOrder	9.84	-258.45	-25.49	134.51	0	-27.91	127.86	27.671	75.79

subsequently to the spot price, at specified time intervals. Backtesting assumes unlimited liquidity, so the Limit Order is fully executed without modifying the price, as would happen in a real market. Actually, in backtesting, the Naïve execution strategy becomes the "Ideal (reference) strategy" that the  $MSM$  metric compares the agents' behaviour against.

Another problem is revealed when comparing metrics for *Buy* and *Sell* orders (Table 16). Generally speaking, if the price of a security goes down, the best *Buy* strategy is the one that buys closer to the end of the period. If the price goes up, then any *Sell* order strategy that breaks an order into chunks will benefit. This is why we can observe in backtesting that, if an agent has a negative  $MSM$  value for a *Buy* order, it will have a positive  $MSM$  value for a *Sell* order on this day. Further, simpler strategies like Naïve Limit order, VWAP agent and Chunked Limit Order have exactly opposite in sign values for *Buy* and *Sell* orders, since they do not adjust their behaviour based on the order's direction.

Considering commissions for Bayesian agents deteriorates the  $MSM$  metric for both *Buy* and *Sell* orders on all days apart from 5th July for the *Sell* order and 2nd July for the *Buy* order (Table 16). Actually, on the trading days before 5th July the closing price was less than the closing price of the previous day, the trend was downwards, as we can see in Figure 7, and on 5th July the trend was already upwards. So it seems that when the trend is upwards, the option to consider commissions results in an improvement of the  $MSM$  metric. Overall, less frequent trading (the one we obtain when considering commissions) should be favourable for *Buy* orders – when the price trend is downwards – and for *Sell* orders – when the price trend is upwards. However, daily fluctuations of

prices can result in this rule not working, as we can see for other days for *Buy* and *Sell* orders (Table 16).

Estimating priors increases the performance of an agent only on 2nd July, for both *Buy* and *Sell* orders. This is probably due to the fact that the price fluctuations of the 1st and 2nd of July are indeed correlated. Interestingly, on the 1st July the news "Amazon bought wacky retailer" were announced [1], which probably resulted in investors selling stocks over 1st and 2nd July, which then created correlated overall drifts in price.

Finally, if we do not consider Naïve Limit order strategy which leads to a 0 *MSM* metric value due to backtest infinite liquidity, we can notice that the Bayesian agents obtain the best *MSM* mean value for the period considered.

**Bayesian agent conclusions** Bayesian agents assume that market price fluctuations are defined by the bulk number of intelligent institutional investors which determine the overall direction of price movements, thus the market created by such investors is not supposed to be very volatile. Estimating priors ( $\bar{\alpha}$  and  $\sigma$ ) of Bayesian agents based on previous day observations is advantageous only in stable markets or if there is a strong correlation between price fluctuations for two days, for example when these days are close to the day a financial report was released or stock news were announced.

Considering commissions may result in large orders being executed at the end of the period and thus in a deterioration in the *MSM* metric values. This could perhaps be mitigated by increasing the commission threshold, so the agents trade more frequently. The more frequent trading seems to improve the overall performance since, even though the agents encounter higher commissions, they are given more frequently the opportunity to react to price movements and can therefore adjust their aggressiveness at the earliest possible moment. Finally, as Bayesian agents are not focused on achieving good VWAP Ratio values, they should not be used to model a VWAP-achieving strategy market.

We would like to conclude by noting that the core assumption of the Bayesian agents is that there is a daily drift in prices caused by the investors' intelligent actions. Almgren *et al.* in [7] mention that this assumption cannot be proven by empirical studies, but it probably does take place, since some real traders express interest in this strategy. From our observations, it seems that, for real data, this assumption does not necessarily hold each day, otherwise Bayesian agents would have consistently outperformed other strategies in the backtesting configuration.

## 6 Conclusions and Future work

On the basis of the results presented in this paper, we propose that *momentum market* is the most suitable model for testing algorithmic trading strategies, since it quickly fills the Limit Order book and produces results comparable to those of a liquid stock (Section 5.1). The standard Bayesian execution agent model (Section 4.4) was implemented and its performance assessed. Additionally, we

proposed an extended Bayesian model that models commissions when generating the schedule of order executions, and also models the ability to estimate prior values for the daily price drift and market volatility, based on previous day observations. The schedule for *Sell* orders was analytically derived using a methodology similar with that employed in [7] for *Buy* orders. Our experiments revealed that the method of estimating priors proposed in this paper can be advantageous in relatively stable markets, when trading patterns in consecutive days are strongly correlated. The results also showed that adjusting the execution strategy according to a reducing commission criterion can result in a worse performance, since agents trading less frequently seem to miss important price trend changes and act with a delay. Additional experiment results and insight analysis, which could not be included in this paper due to space constraints, are included in [27].

In terms of future work, in the Bayesian agent analysis it would be worth experimenting with different commission thresholds, to find the best proportion of price updates frequencies and trading.

### Acknowledgements

Natalia Ponomareva would like to gratefully acknowledge the Hill Foundation for supporting her study for an MSc degree at the Department of Computer Science of the University of Oxford.

### References

1. Amazon buys wacky retailer. [http://www.theregister.co.uk/2010/07/01/amazon\\_buys\\_woot/](http://www.theregister.co.uk/2010/07/01/amazon_buys_woot/).
2. Cboe historical stock volatilities. <http://www.cboe.com/data/historicalvolatility.aspx>.
3. Google finance. <http://www.google.com/finance>.
4. Jasa - java auction simulator api, home page. <http://www.essex.ac.uk/ccfea/research/software/jasa/>.
5. Online tick-level dataset ducascopy. <http://freeserv.dukascopy.com/exp/>.
6. A resource for agent- and individual-based modelers and the home of swarm. [http://www.swarm.org/index.php/Main\\_Page](http://www.swarm.org/index.php/Main_Page).
7. R. Almgren and J. Lorenz. Bayesian Adaptive Trading with a Daily Cycle. *The Journal of Trading*, 1(4):38–46, Jan. 2006.
8. T. G. Andersen, T. Bollerslev, F. X. Diebold, and H. Ebens. The distribution of realized stock return volatility. *Journal of Financial Economics*, 61:43–76, 2001.
9. C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
10. E. Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *PNAS*, 99:7280–7287, 2002.
11. K. H. Chung and Y. Kim. Volatility, market structure, and the bid-ask spread\*. *Asia-Pacific Journal of Financial Studies*, 38(1):67–107, 2009.
12. W. Cui, A. Brabazon, and M. O’Neill. Efficient trade execution using a genetic algorithm in an order book based artificial stock market. In *Conference Companion on Genetic and Evolutionary Computation*, pages 2023–2028, 2009.

13. W. Cui, A. Brabazon, and M. O'Neill. Evolving dynamic trade execution strategies using grammatical evolution. In *Proceedings of the International Conference on Applications of Evolutionary Computation*, pages 192–201, 2010.
14. G. Daniel. *Asynchronous Simulations of a Limit Order Book*. PhD thesis, University of Manchester, 2007.
15. I. Domowitz and H. Yegerman. The Cost of Algorithmic Trading. A First Look at Comparative Performance. *The Journal of Trading*, 2005.
16. L. H. Ederington and W. Guan. Measuring Historical Volatility. *Journal of Applied Finance*, 16, 2006.
17. R. Engle and A. Patton. What good is a volatility model? *Quantitative Finance*, 1:237–245, 2001.
18. J. D. Farmer, P. Patelli, and I. I. Zovko. The Predictive Power of Zero Intelligence in Financial Markets. *SSRN eLibrary*, 2004.
19. M. Gsell. Assessing the impact of algorithmic trading on markets: A simulation approach. *16th European Conference on Information Systems*, pages 587–598, 2008.
20. T. Hendershott, C. Jones, and A. Menkveld. Does algorithmic trading improve liquidity? *The Journal of Finance*, 66:133, 2011.
21. K. Izumi, F. Toriumi, and H. Matsui. Evaluation of automated-trading strategies using an artificial market. *Neurocomput.*, 72:3469–3476, October 2009.
22. C. X. Jiang, J.-C. Kim, and R. A. Wood. A comparison of volatility and bidask spread for NASDAQ and NYSE after decimalization. *Applied Economics*, 43(10):1227–1239, 2011.
23. S. M. Kakade, M. Kearns, Y. Mansour, and L. E. Ortiz. Competitive algorithms for VWAP and limit order trading. In *Proceedings of the 5th ACM conference on Electronic commerce*, pages 189–198, 2004.
24. B. Lebaron. Agent-based computational finance. In *Handbook of Computational Economics, Agent-based Computational Economics*, pages 166–209, 2006.
25. Y. Nevmyvaka, Y. Feng, and M. Kearns. Reinforcement learning for optimized trade execution. In *Proceedings of the 23rd international conference on Machine learning*, pages 673–680, 2006.
26. Y. Nevmyvaka, M. S. Kearns, A. Papandreou, and K. P. Sycara. Electronic trading in order-driven markets: Efficient execution. In *CEC'05*, pages 190–197, 2005.
27. N. Ponomareva. Using agent-based modelling and backtest to evaluate algorithmic trading strategies. Master's thesis, Department of Computer Science, University of Oxford, 2011.
28. S. Raghavendra and D. Paraschiv. A framework for testing algorithmic trading strategies. Technical report, 2008.
29. A. Rashid. Using a service oriented architecture for simulating algorithmic trading strategies. In *Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services, iiWAS '10*, pages 925–929, New York, NY, USA, 2010. ACM.
30. R. Roll. A simple implicit measure of the effective bid-ask spread in an efficient market. *The Journal of Finance*, 39:1127–1139, 1984.
31. W. Sharpe, G. J. Alexander, and J. W. Bailey. *Investments*. Prentice Hall, 6 edition, 1998.
32. F. Wang, K. Dong, and X. Deng. Algorithmic trading system: design and applications. *Frontiers of Computer Science in China*, 3:235–246, 2009.
33. P. R. Wurman, W. E. Walsh, and M. P. Wellman. Flexible double auctions for electronic commerce: theory and implementation. *Decision Support Systems*, 24(1):17 – 27, 1998.