IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 9, 2021, accepted April 20, 2021, date of publication April 26, 2021, date of current version May 3, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3075700

Receding Horizon Optimization of Large Trade Orders

GERARDO ALFONSO[1,2], A. DANIEL CARNERERO[2], DANIEL R. RAMIREZ[2], AND TEODORO ALAMO[2]

[1]Shenwan and Hongyuan Securities Co., Ltd. (SWS), Shanghai 200031, China
[2]Departamento de Ingeniería de Sistemas y Automática, Universidad de Sevilla, 41092 Sevilla, Spain

Corresponding author: Daniel R. Ramirez (danirr@us.es)

This work was supported by the Agencia Estatal de Investigación (AEI), Spain, under Grant PID2019-106212RB-C41/AEI/10.13039/501100011033.

ABSTRACT The optimal execution of stock trades is a relevant and interesting problem as it is key in maximizing profits and reducing risks when investing in the stock market. In the case of large orders, the problem becomes even more complex as the impact of the order in the market has to be taken into account. The usual solution is to split large orders into a set of smaller suborders that must be executed within a prescribed time window. This leads to the problem of deciding when in the time window execute each suborder. There are popular ways of executing the trading of these split orders like those which try to track the ''Time Weighted Average Price'' and the ''Volume Weighted Average Price'', usually called TWAP and VWAP orders. This paper presents a strategy to optimize the splitting of large trade orders over a given time window. The strategy is based on the solution of an optimization problem that is applied following a receding horizon approach. This approach reduces the impact of prediction errors due to the uncertain market dynamics, by using new values of the price time series as they are available as time goes on. Suborder size constraints are taken into account in both market and limit orders. The strategy relies on price and traded volume forecast but it is independent of the prediction technique used. The performance index weighs not only the financial cost of the suborders, but also the impact on the market and the forecasting accuracy. A tailored optimization algorithm is proposed for efficiently solving the corresponding optimization problem. Most of the computations of the algorithm can be parallelized. Finally, the proposed approach has been tested through a case study composed by stocks of the Chinese A-share market.

INDEX TERMS Algorithmic trading, receding horizon optimization, large stock orders, limit orders, TWAP, VWAP.

I. INTRODUCTION

Stock trading is becoming an increasingly complex field as investors try to maximize their profits and reduce their risks with increased emphasis in recent years on optimal execution. After an investor has decided to buy or sell stocks, there are many options on how to execute this trade. In large stock orders, it is very frequent to split orders into smaller orders in an attempt not to impact the market. For instance, a buy order to purchase a large amount of stock could push the price of the stock up if executed in one single block. The market impact of that order can be potentially reduced by splitting that single order into smaller orders and executing them over time rather than in one go. There are many ways in which this can be accomplished. One of the simplest approaches is called ''Time Weighted Average Price'', commonly referred to as TWAP. A TWAP order splits the order in blocks of shares of the same size that are then executed at regular time intervals. A more sophisticated trade is a ''Volume Weighted Average Price'', commonly referred to as VWAP. This type of order is very common as it represents around 50% of all the institutional investors' trading [1]. In a VWAP order, rather than slicing the original order in smaller trades of equal size (equal number of shares), a forecasted traded volume for the desired interval is estimated. The size of each (sliced) transaction is proportional to the forecasted volume for the corresponding time bucket. Thus, a critical step in this approach is to be able to generate volume forecasts. Besides these two popular

The associate editor coordinating the review of this manuscript and approving it for publication was Hao Wang.

ways of trading sliced orders, there are basically two types of trading orders: market orders and limit orders. A market order is an order to be executed immediately at the prevailing market price. The focus on this type of order is speed rather than price optimization. Limit orders, on the other hand, focus on price efficiency. A limit order necessarily has an associated price over (below) which the buy (sell) order cannot be executed. There is a tradeoff between speed of execution and price optimization. Moreover, in the case of limit orders, there is no certainty of execution as it is possible to not have enough time buckets in the execution period for which the price meets the limit. Thus, in these orders, success is measured not only by the price attained, but also by the percentage of the order that has been executed.

There is relatively limited existing literature on the topic of optimal execution of split trade orders using learning techniques, with more papers covering stock forecasting by means of different techniques like neural networks [2] or deep learning [3], support vector machines [4], [5], adaptive line combiners [6] or local data-based techniques [7]. There are however some interesting articles in the field of trade execution optimization such as for instance [8]. In this article, the authors proposed a genetic algorithm to optimize a limit order book used for price formation in an artificial stock market. Genetic algorithms are also used in [9] to generate trading strategies, not based on forecasting, that are back-tested against historical data of the Australian Stock Exchange.

Recurrent neural networks have been used in [10] to predict price-flip events in limit order books by classifying sequences of observations of the book depths and market orders. Forecasting the traded volume has also been used in [11] to improve the execution of VWAP orders. That is, by forecasting the traded volume one can track the VWAP price matching it at the end of the chosen time window. On the other hand, [12] derives analytical solutions of a static optimal execution strategy of a VWAP trade, in which the optimal execution strategy can be calculated by an iteration of a single variable optimization, rather than by a multivariable optimization. In that work, the market is modelled using non-anticipating and Brownian motion processes.

Particle swarm optimization has been used by [13] in a high-frequency trading system based on moving averages, used to determine the trading sequence that maximizes the net returns over a series of consecutive time steps. This optimization technique has also been used by [14] to train a kernel-based nonlinear predictor that was also applied to forecast the VWAP price in the Shanghai market.

Optimal control methods have been also considered for generating the sequence of suborders in split large orders. For instance, [15] presented an optimal VWAP algorithm based on the linear quadratic regulator (LQR) subject to limits in the size of the suborders. The linear model for the stock prices is based on Brownian motion, a type of model that has also been used by [16] to find the stock prices also in the context of VWAP operations. Hamilton-Jacobi-Belmann methods and in general variational calculus have been used in [17]–[19] based on Brownian motion and random walks models. A recent paper [20] has studied optimal VWAP strategies using unconstrained optimization on models based on the assumption that the stock prices can be modelled as martingales and the traded volume as autoregressive processes.

Time series methods have been used in [21], where an autoregressive fractionally integrated moving average model is used to forecast intraday trading volumes in the Chinese equity market, and its application to VWAP tracking, obtaining better results than static approaches. VWAP tracking has also been tackled in [22], where an interesting combination of historical averages and SVM to forecast intraday trading volumes in the gold and S&P 500 futures markets was used. Another interesting approach to forecasting trading volumes in a VWAP tracking context has been presented in [23]. In this paper, the authors used the fast Fourier transform algorithm to identify the periodic and the non-periodic part of the trading volume, using historical values of 50 stocks contained in the Shanghai 50 stock index. A similar approach is followed in [24].

In this paper, we propose a data-based method that relies on price and volume forecasts, and the use of dynamic optimization over a finite horizon, to obtain optimal sequences of suborders to fulfil large trade orders. The optimal sequence is computed by minimizing a cost index, in which several terms are taking into account. Besides the price, the trading impact factor is considered along with a term related to forecasting accuracy. The technique is based on solving an optimization problem each time bucket of the time window in which the order has to be executed. Thus, at each time bucket, a complete sequence of suborders for the remaining time window is obtained, but only the first component of the sequence is effectively used. This strategy is similar to the feedback receding horizon or predictive control strategy used in automatic control [25], but with a shrinking prediction horizon. Using such an approach allows us to split the order into different suborders each of them computed with the most recent available information (and thus with improved forecastings). Therefore, this approach reduces the effects of the prediction errors that are bound to arise, especially if the prediction horizon is large.

The optimization is carried out by means of a tailored, efficient algorithm that can solve the optimization problem very fast. Furthermore, most of its computations can be done in parallel if necessary. This computational efficiency can be very useful if high-frequency trading is considered, or if learning or simulation-based strategies are used to tune the hyperparameters of the algorithm. Regarding the type of orders, both market and limit orders are considered in the paper and treated in a unified way. Finally, as a case study, stocks from the onshore Chinese A-share market are used to show the performance of the proposed strategy.

The rest of the paper is outlined as follows: Section II describes the problem statement for each type of order. Section III presents the optimization algorithm. Section IV presents the results of applying the proposed technique to the case study. Finally, section V presents the conclusions.

II. PROBLEM STATEMENT

The objective of this paper is to design a strategy to execute large stock orders that, in order to limit their impact on the market, have to be split into a number of smaller suborders. Thus, an order to buy $M \in \Re^+$ shares will be executed by splitting the order into up to N suborders $m(t+k) \in \Re^+$, with $k \in [1, N]$, such that the sum of all the suborders is equal to M or, in the case of limit orders, the closest possible to M. Note that in practice, the number of shares to be traded must be a natural number, so the number of shares in the suborders that are sent to the market are rounded to the nearest smaller integer, that is $\lfloor m(t+k) \rfloor$. Given that the number of shares considered in these orders is quite large, often hundreds of thousands or millions of shares per order, the difference in financial cost between considering real or integer-valued suborders is negligible. However, from a computational point of view, the difference is quite significant, being much more efficient an algorithm based on real-valued suborders.

The proposed strategy computes the splitting of the original order in an optimal way, that is, achieving the most convenient price. The strategy is based on forecasting both the price of the stock and its total traded volume over the time window defined by N, i.e., from $t+1$ to $t+N$. The price and trade volume forecasts are used to compute a performance index that should be optimized. Thus the strategy will rely on an optimization problem in which a performance index, denoted as V_N, is optimized. The formulation of V_N must weigh certain aspects of concern, such as the total cost of the order, but also the impact on the market of each of the suborders. Besides that, given that forecasts are used to compute the performance index, a term penalizing the potential degradation of forecasts with the prediction horizon is also included in V_N. Let $\hat{p}(t+k|t)$ be the price forecast for $t+k$, $\hat{v}(t+k|t)$ the total traded volume forecast[1] for $t+k$ and $m(t+k|t)$ the number of shares to be bought at $t+k$, meaning the notation $t+k|t$ that these values are computed at time t. Then the proposed performance index is:

$$V_N(\mathbf{m}_N(t), \hat{\mathbf{p}}_N(t), \hat{\mathbf{v}}_N(t)) = \sum_{k=1}^{N} \left(a\hat{p}(t+k|t) + \alpha \left(\frac{m(t+k|t)}{\hat{v}(t+k|t)} \right)^{\beta} \right) m(t+k|t) + \mu \sum_{k=1}^{N} \sigma^k m(t+k|t), \quad (1)$$

where $a = 1$ in buy orders or $a = -1$ in sell orders, the tuning parameters $\alpha, \mu, \sigma, \beta$ are nonnegative with $\beta \geq 1$, and

$$\mathbf{m}_N(t) = [m(t+1|t), \ldots, m(t+N|t)],$$
$$\hat{\mathbf{p}}_N(t) = [\hat{p}(t+1|t), \ldots, \hat{p}(t+N|t)],$$
$$\hat{\mathbf{v}}_N(t) = [\hat{v}(t+1|t), \ldots, \hat{v}(t+N|t)],$$

the sequences of suborders, price and traded volume forecastings respectively. Note that the first term represents the forecasted amount or economic figure of the buy or sell operation, and that in this term an impact factor correction has been included. Impact factor correction represents the influence of the suborder volume in the price for that time bucket. Impact factors can be modelled as linear terms, like in [15] or be more elaborate like the exponential form used here based on [26]. The second term assigns a greater cost to suborders that are far in the future, because the prediction error grows with the prediction horizon. Note that for this effect the exponential weight σ^k must have $\sigma > 1$.

The proposed strategy aims to find the optimal sequence of suborders

$$\mathbf{m}_N^*(t) = [m^*(t+1|t), m^*(t+2|t), \ldots, m^*(t+N|t)], \quad (2)$$

that minimizes the performance index V_N over the time window N. However, depending on the type of order, some constraints have to be taken into account.

In the case of market orders, the suborders must meet a certain size limit so that the impact in the market is limited, i.e.,

$$0 \leq m(t+k|t) \leq \overline{m}, \quad \forall k \in [1, N],$$

being $\overline{m} = 0.1M$ a typical value (i.e., a suborder cannot exceed a 10% of the whole order). On the other hand, in limit orders, the size limit will be the same, but only if the price forecast meets the price limit p_l. Notice that, in the case of a buy limit order, $\hat{p}(t+k|t) > p_l$ implies that $m(t+k|t) = 0$, that is, the buy suborder cannot be sent to the market. Similarly, in sell limit orders $\hat{p}(t+k|t) < p_l$ implies that $m(t+k|t) = 0$. These situations can be addressed with the following constraint

$$0 \leq m(t+k|t) \leq \overline{m}_k, \quad \forall k \in [1, N] \quad (3)$$

where

$$\overline{m}_k = \begin{cases} 0 & \text{if limit order \& } a(\hat{p}(t+k|t) - p_l) > 0 \\ \overline{m} & \text{otherwise} \end{cases} \quad (4)$$

On the other hand, the sum of all suborders should be, in principle, equal to the total number of shares to be traded (M) thus the equality constraint

$$\sum_{k=1}^{N} m(t+k|t) = M$$

should be taken into account. In the case of limit orders, the possibility of not having enough time buckets in which the price limit is met, must be taken into consideration. To illustrate this consider that if every suborder must be at most 10% of the original order and the price forecast is only under the limit in 7 time buckets, the original order can only be completed at most at a 70%, which would be equal to

[1] It is assumed that price and volume forecast are greater than zero. While this is met almost always for the price, the trade volume can be indeed zero, thus zero volume forecasts should be changed for an arbitrarily small value.

the sum of the corresponding $\overline{m}_k$ for all the time window. That is, when the order cannot be fully executed, the largest percentage of fulfilment of the order is attained forcing

$$\sum_{k=1}^{N} m(t+k|t) = \sum_{k=1}^{N} \overline{m}_k.$$

Note that if

$$\sum_{k=1}^{N} \overline{m}_k \geq M$$

then the order can be fully executed. Thus, to be able to address both situations, we include the following constraint that maximizes the degree of fulfilment of the order,

$$\sum_{k=1}^{N} m(t+k|t) = M_c, \tag{5}$$

with

$$M_c = \min\left\{M, \sum_{k=1}^{N} \overline{m}_k\right\}. \tag{6}$$

Note that this constraint will work for either market or limit orders.

Once all the necessary constraints have been formulated, the optimal sequence of suborders will be obtained by solving the optimization problem:

$$\begin{aligned} \mathbf{m}_N^*(t) = \arg\min_{\mathbf{m}_N(t)} \; & V_N(\mathbf{m}_N(t), \hat{\mathbf{p}}_N(t), \hat{\mathbf{v}}_N(t)) \\ \text{s.t.} \; & (3) \text{ and } (5). \end{aligned} \tag{7}$$

After problem (7) is solved, one could apply the entire optimal sequence $\mathbf{m}^*(t)$ executing the suborders $m^*(t+1|t), \ldots, m^*(t+N|t)$ at the corresponding time buckets. This approach suffers from two related issues: first, as the index k grows, the forecastings $\hat{p}(t+k|t)$ and $\hat{v}(t+k|t)$ have a higher prediction error. Thus the suborders $m(t+k)$ will rely on progressively more inaccurate forecastings and thus the computed optimal value will differ from the ideal optimal value that could be computed if the real values of $p(t+k)$ and $v(t+k)$ were known in advance. Second, as time advances, new real values of $p(t+k)$ and $v(t+k)$ are available, but they are not used. These new values could be used to obtain better predictions of the remaining time window, allowing to compute optimal suborders that would be closer to their ideal values, i.e., those obtained without prediction errors. Thus, in this paper it is proposed to use a feedback receding horizon optimization strategy, typical of predictive control techniques [25], in which at each time bucket t problem (7) is solved to obtain $\mathbf{m}_N^*(t)$ sending only $m^*(t+1|t)$ to the market and discarding the rest of the sequence $\mathbf{m}_N^*(t)$. To apply this receding horizon strategy it is necessary to take into account that the time window length shrinks with time, thus an initial time window length, denoted as N_p, must be considered. In this way, initially $N = N_p$, and then it will be decreasing at each time step. Furthermore, constraint (5) must reflect the fact that M would be decreasing as suborders are being sent to the market. Thus, an initial size order M_I will be considered, i.e., for the first time $M = M_I$, and then updated as necessary.

Besides the aforementioned changes, the receding horizon strategy can help in mitigating the negative effect of price prediction error in limit orders. This error can affect in two ways, one is that the price limit is met by the predicted price, whereas the real price fails to meet the limit. The other is the reverse, that is, the predicted price does not meet the limit, but the real one does. The solution to these issues is to compute the optimal sequence of suborders for the case that $\hat{p}(t+1|t)$ fails to meet p_l and for the case that it meets p_l, and then, at time $t+1$, apply the one that corresponds to the value of $p(t+1)$. For the first case, it is not really necessary to perform any computation as it is obvious that $m^*(t+1|t) = 0$. For the second case, problem (7) must be solved with the bound $\overline{m}_1 = \overline{m}$ regardless the value of $\hat{p}(t+1|t)$. So, this modification can be easily implemented by just changing (4) to

$$\overline{m}_k = \begin{cases} 0 & \text{if limit order \&} \\ & a(\hat{p}(t+k|t) - p_l) > 0 \;\; \& \;\; k > 1 \\ \overline{m} & \text{otherwise} \end{cases} \tag{8}$$

and sending to the market $m^*(t+1|t)$ only if $p(t+1)$ meets p_l. Algorithm 1 summarizes the proposed strategy.

Algorithm 1 Optimal Execution of Large Orders

Require: $M_I, N_p, a, \alpha, \beta, \mu, \sigma, \overline{m}$.
- $M \leftarrow M_I$.
- $N \leftarrow N_p$.
- **repeat**
  - Compute the forecasts $\hat{\mathbf{p}}_N(t)$ and $\hat{\mathbf{v}}_N(t)$.
  - Compute the bounds $\overline{m}_k$ as in (8).
  - Compute M_c as in (6).
  - Solve (7) to obtain $\mathbf{m}_N^*(t)$.
  - Wait for next time bucket, that is, $t+1$.
  - Compute
    $$m_{t+1}^* = \begin{cases} 0 & \text{if limit order \& } a(p(t+1) - p_l) > 0 \\ m^*(t+1|t) & \text{otherwise} \end{cases}$$
  - **if** $m_{t+1}^* > 0$ **then**
    - Send m_{t+1}^* to the market.
    - $M \leftarrow M - m_{t+1}^*$.
  - **end if**
  - $N \leftarrow N - 1$.
- **until** $N = 0$ or $M = 0$.

Remark 1: The initial prediction horizon N_p can be quite long; thus the prediction errors can affect the performance of the proposed strategy. The receding strategy helps to mitigate such effects by using forecastings obtained with the most up to date information. Note also that the time window shrinks at each step, reducing the prediction horizon and allowing us to work with progressively better forecasts. Furthermore, the receding horizon only sends to the market the first suborder in $\mathbf{m}_N^*(t)$, that mostly depends on short term forecastings, which are the most accurate. Finally, the forecasting errors in long prediction horizons are also compensated by the second term of (1), assigning more weight to suborders that are placed far in the prediction horizon depending on the accuracy of the forecasting method.

The performance of the strategy depends, not only on the price and traded volume forecastings, but also on the choice of the performance function (1) hyperparameters. Regarding the tuning of such hyperparameters, α and β should be chosen by the user so that they reflect the impact of each suborder in the stock price. This implies a modelling task that is previous to the decision of investing in a particular stock and that it is beyond the scope of this paper (see [26]). On the other hand, σ and μ can be chosen considering the length of the prediction horizon and the long term accuracy of the forecasting method used to predict the price time series. Alternatively, the algorithm hyperparameters can be tuned by performing a search of optimal values running simulations based on historical data, choosing the values that optimize a certain performance metric, e.g., the total financial cost of executing the whole order.

Finally, note that at each time bucket, problem (7) must be solved. Because of the high frequency of nowadays trading systems, it is clear that an efficient solving method for (7) is needed. Furthermore, having an efficient optimization algorithm is important if the optimization of the hyperparameters is carried out, as this implies solving (7) many times.

III. OPTIMIZATION ALGORITHM

The nonlinear optimization problem (7) is relatively easy to solve, as the performance index and constraints are convex. Note that the equality constraint (5) is the cause of most of the computational burden of problem (7). This constraint is the one that is coupling the N decision variables of problem (7). Without this constraint, the N decision variables optimization problem could be solved as N independent problems of just one decision variable. This would be much more efficient from a computational point of view, as these N problems could even be solved in a parallel way. In order to decouple the decision variables, we resort to a dual formulation. Consider, for a given $\lambda \geq 0$, the problem:

$$\begin{aligned} J^*(\lambda) = \min_{\mathbf{m}_N(t)} \quad & V_N(\mathbf{m}_N(t), \hat{\mathbf{p}}_N(t), \hat{\mathbf{v}}_N(t)) \\ & + \lambda \left(M_c - \sum_{k=1}^{N} m_k \right) \\ \text{s.t.} \quad & 0 \leq m_k \leq \overline{m}_k, \quad \forall k \in [1, N], \end{aligned} \tag{9}$$

where, for notational convenience, m_k denotes $m(t+k|t)$. This optimization problem does not have in general the same solution as (7), but as will be shown in the following, for a certain value of λ its solution would be the same and can be obtained with a closed formula.

Taking into account the definition on V_N in (1) and that this problem is free of the constraint (5), the solution of (9) can be computed by solving for every $k \in [1, N]$ the following problem:

$$m_k^*(\lambda) = \arg\min_{m_k} \quad J_k(m_k, \lambda) \tag{10}$$
$$\text{s.t.} \quad 0 \leq m_k \leq \overline{m}_k \tag{11}$$

where $m_k^*(\lambda)$ denotes the optimal value of m_k for a given λ and

$$J_k(m_k, \lambda) = \left(a\hat{p}(t+k|t) + \alpha \left(\frac{m_k}{\hat{v}(t+k|t)} \right)^{\beta} + \mu\sigma^k - \lambda \right) m_k$$

which in turn can be rewritten as

$$J_k(m_k, \lambda) = (d_k - \lambda) m_k + e_k m_k^{\beta+1}$$

with

$$d_k = a\hat{p}(t+k|t) + \mu\sigma^k$$

and

$$e_k = \frac{\alpha}{\hat{v}(t+k|t)^{\beta}} \geq 0. \tag{12}$$

Note that each $m_k^*(\lambda)$ for $k = 1, \ldots, N$ can be computed independently, thus the N problems (10) can be solved in parallel.

To find $m_k^*(\lambda)$ consider the partial derivative of $J_k(m_k, \lambda)$:

$$\frac{\partial J_k(m_k, \lambda)}{\partial m_k} = d_k - \lambda + (\beta+1) e_k m_k^{\beta}.$$

From (11) and (12) we have that m_k and e_k are non-negative. This implies that the partial derivative is strictly positive if $d_k - \lambda > 0$. Thus, in this case, $J_k(m_k, \lambda)$ is a monotonically growing function of m_k and the minimum is attained at $m_k = 0$. That is,

$$m_k^*(\lambda) = 0 \ \text{ if } \ \lambda \leq \lambda_k^- = d_k. \tag{13}$$

On the other hand, if

$$d_k - \lambda \leq 0,$$

then the zero of the derivative, i.e. the minimizer of the unconstrained problem is attained at

$$\left(\frac{\lambda - d_k}{(\beta+1) e_k} \right)^{\frac{1}{\beta}}. \tag{14}$$

We notice that, in this case, the minimizer of the unconstrained problem coincides with the minimizer of the constrained problem only if

$$\left(\frac{\lambda - d_k}{(\beta+1) e_k} \right)^{\frac{1}{\beta}} \leq \overline{m}_k,$$

This is equivalent to

$$\lambda \leq \lambda_k^+ = \overline{m}_k^{\beta} (\beta+1) e_k + d_k. \tag{15}$$

Thus, for all $\lambda \geq \lambda_k^+$ the constraint (11) determines the minimizer, i.e., $m_k^*(\lambda) = \overline{m}_k$. Summing up, the solution of (10) is

$$m_k^*(\lambda) = \begin{cases} 0 & \text{if } \lambda \leq \lambda_k^- \\ \left(\dfrac{\lambda - d_k}{(\beta+1)e_k}\right)^{\frac{1}{\beta}} & \text{if } \lambda \in [\lambda_k^-, \lambda_k^+] \\ \overline{m}_k & \text{if } \lambda \geq \lambda_k^+, \end{cases} \tag{16}$$

where λ_k^- and λ_k^+ are given in (13) and (15).

Note that the values of λ providing values different from 0 and $\overline{m}_k$ are those in the interval $[\lambda_{\min}, \lambda_{\max}]$, where

$$\lambda_{\min} = \min_{k \in [1,N]} \lambda_k^-, \tag{17}$$

and

$$\lambda_{\max} = \max_{k \in [1,N]} \lambda_k^+. \tag{18}$$

Once the value of $m_k^*(\lambda)$ has been obtained, consider $M^*(\lambda)$ defined as

$$M^*(\lambda) = \sum_{k=1}^{N} m_k^*(\lambda). \tag{19}$$

In what follows, it will be shown that there exist $\lambda^* \in [\lambda_{\min}, \lambda_{\max}]$ such that

$$M^*(\lambda^*) = M_c, \tag{20}$$

which implies that constraint (5) holds and the combined solutions $m_k^*(\lambda^*)$ will be equal to that of (7), i.e.,

$$\mathbf{m}_N^*(t) = [m_1^*(\lambda^*), m_2^*(\lambda^*), \ldots, m_N^*(\lambda^*)]. \tag{21}$$

The following property is key to show that λ^* exists.

Property 1 (Extreme Values of m_k^* and M^*): The extreme values of $m_k^*(\lambda)$ and $M^*(\lambda)$ as functions of λ are

$$\begin{cases} m_k^*(\lambda_{\min}) = 0, & \forall k \in [1,N] \\ M^*(\lambda_{\min}) = 0, \end{cases}$$

and

$$\begin{cases} m_k^*(\lambda_{\max}) = \overline{m}_k, & \forall k \in [1,N] \\ M^*(\lambda_{\max}) = \displaystyle\sum_{k=1}^{N} \overline{m}_k \geq M_c. \end{cases}$$

Proof: These values stem directly from (16) and the definitions of $\lambda_{\min}$ and $\lambda_{\max}$. Taking into account that $\lambda_{\min} \leq \lambda_k^-$, it is clear that $m_k^*(\lambda_{\min}) = 0$, for all k. This in turn, makes that, by definition, $M^*(\lambda_{\min}) = 0$.

On the other hand, $\lambda_{\max} \geq \lambda_k^+$, which implies that $m_k^*(\lambda_{\max}) = \overline{m}_k$, for all k. Furthermore, $M^*(\lambda_{\max}) = \sum_{k=1}^{N} \overline{m}_k$, which due to (6) is greater or equal to M_c. ∎

Note that, taking into account (16) it is obvious that $m_k^*(\lambda)$ is continuous, as at $\lambda = \lambda_k^-$, $m_k^*(\lambda_k^-) = 0$ from both sides and at $\lambda = \lambda_k^+$ its value is $m_k^*(\lambda_k^+) = \overline{m}_k$.

TABLE 1. Stock dataset. The following levels of market capitalization were follow for stock classification: small (<80 bn RMB), mid (from 80 to 300 bn) and large (>300 bn).

Name	Ticker	Capitalization
Nanjing Port	002040	Small
Baotou Dongbao Biotech	300239	Small
Royal Flush Information Network	300033	Small
Advanced micro-fabrication equipment	688012	Mid
Tianqi Lithium	002466	Mid
Shangahi Pudong Development Bank	600000	Mid
Midea Group	000333	Large
Ping An Bank	000001	Large
Kweichow Moutai	600519	Large

Moreover, consider the derivative of $m_k^*(\lambda)$, which is

$$\frac{\partial m_k^*(\lambda)}{\partial \lambda} = \left(\frac{1}{\beta((\beta+1)e_k)^{\frac{1}{\beta}}}\right)(-d_k + \lambda)^{\frac{1}{\beta}-1}$$

if $\lambda \in (\lambda_k^-, \lambda_k^+)$ and 0 otherwise. Taking into account that $e_k \geq 0$ and that, by definition, $-d_k + \lambda_k^- = 0$, and $-d_k + \lambda_k^+ \geq 0$, it follows that the derivative is nonnegative, thus $m_k^*(\lambda)$ is a nondecreasing function.

Given that $m_k^*(\lambda)$ is a continuous nondecreasing function and considering (19), it follows that $M^*(\lambda)$ is also a continuous nondecreasing function. Furthermore, M_c is nonnegative and by property 1,

$$M_c \in \left[0, \sum_{k=1}^{N} \overline{m}_k\right] = [M^*(\lambda_{\min}), M^*(\lambda_{\max})].$$

This implies that a certain value $\lambda^* \in [\lambda_{\min}, \lambda_{\max}]$ exists, such that $M^*(\lambda^*) = M_c$.

The value λ^* that makes (20) and, therefore, (5) hold, is computed by performing a simple bisection search in the interval $[\lambda_{\min}, \lambda_{\max}]$. Once this λ^* is found, the optimal value of the suborders, that is $m_k^*(\lambda^*)$, will already be computed, and the solution of problem (7) will be (21).

Remark 2: The proposed strategy can be easily adapted to other forms of stock orders which require other constraints, like the so-called "max" orders in which the suborder size is also limited by a fraction of the forecasted traded volume for each time bucket. In these cases, the multiple inequality constraints should be reduced to one using a simple redundant constraint elimination.

IV. CASE STUDY: CHINESE STOCK MARKET

The proposed approach has been validated using stocks from the Chinese Stock Market. Every working day has two continuous trading sessions, each one two hours long. Thus, the objective of the case study will be to split large orders over a 120 minute session, being the time bucket 1 minute long.

To validate the proposed strategy, nine stocks from the Chinese Stock Market have been chosen, splitting the choices between small, mid or large-capitalization stocks. Table 1 show the names and tickers of the nine stocks ordered according to their capitalization. The validation consists on the optimization of market orders of 2, 000, 000 shares over a

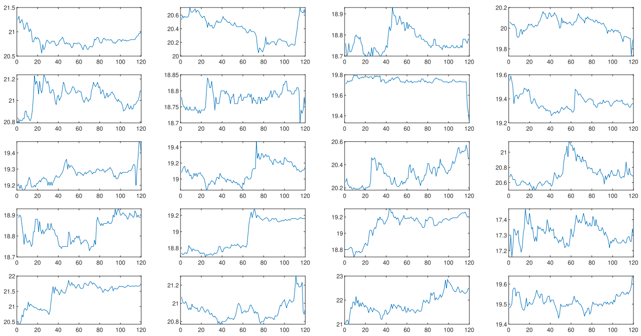


FIGURE 1. Validation data set for 002040.

FIGURE 2. Validation data set for 688012.

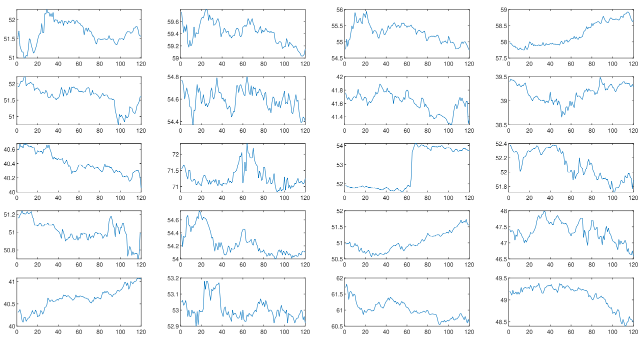


FIGURE 3. Validation data set for 000333.

validation set of 20 two hour sessions randomly chosen from the range 1/4/2017 to 12/29/2020 (except the newer stock 688012 which is 1/2/2020 to 12/29/2020). For conciseness, only three of the validation datasets are shown in figures 1 to 3. It can be seen that the sessions in each set show bearish, bullish and sideways trends.

The proposed trading strategy relies on forecasts of prices and traded volumes, but it is not linked to any particular forecasting technique. Here, neural networks and the local data-based technique proposed in [7] have been considered. In order to forecast prices using either technique, the market state, denoted as $z_p(t)$ has been described as composed by the last 100 prices and the last traded volume,[2] i.e.,

$$z_p(t) = [p(t), p(t-1), \ldots, p(t-99), v(t)].$$

[2]This paper is not focused on the forecasting task but on the optimization of the splitting of the large order. Better forecastings could be obtained by carefully testing and choosing other market state vectors, e.g., using technical indicators, but this task is outside the scope of this paper.

On the other hand, to forecast the traded volume, the market state, denoted by $z_v(t)$ is formed in this case by

$$z_v(t) = [v(t), v(t-1), \ldots, v(t-99), p(t)].$$

The components of both state vectors are lightly smoothed with an exponential moving average of 5 minutes (EMA5), that is,

$$p_{EMA}(t) = \frac{2}{6}p(t) + \left(1 - \frac{2}{6}\right) p_{EMA}(t-1),$$

but the target prices to be forecasted are unsmoothed. The number of market states in the local data set used by the technique of [7] has been chosen to be 350 and the weighting factor $\gamma = 0$, whereas in the case of the neural network predictors, a Multilayer Perceptron with one hidden layer of 40 neurons has been used. Note that in the case of the local data technique of [7] it is not necessary to train the predictor, whereas in the case of the neural network, 120 networks have been trained using the Levenberg-Marquardt rule, one for each time bucket in the time window. It is noteworthy that the prediction horizon, determined by the time window and time bucket, is quite long, being $N_p = 120$, thus it is quite challenging in relation to the prediction errors, that are bound to arise. Taking into account that both price and volume have to be forecasting using one of these techniques, four possible prediction models have been considered, one using only neural networks (denoted as NN-NN), other relying only on the local data approach of [7] (LD-LD), and two combining these two techniques, i.e., LD-NN that predicts prices using the local data approach and volumes with neural networks and the reverse case, i.e., NN-LD.

We first analyze market orders in which the objective is to split the large stock order, over each of the sessions of the validation set, optimizing the buying price. Thus the proposed strategy must solve (7) with an initial prediction horizon of $N = N_p = 120$ that will shrink down to $N = 1$ as the receding horizon optimization is applied through the entire time window. Other parameters of V_N used in this case study are, $a = 1$ (buy orders), $\alpha = 0.001$, $\beta = 1.1$, $\mu = 0.6$, $\sigma = 1.05$. The average prices obtained when using the proposed approach over the validation set of each stock with each of the four prediction models are shown in table 2. It can be seen that no model can be chosen as the best for all the stocks, even if they are close in their results (especially for models LD-LD and LD-NN). In fact, when considering the size of the orders, it is clear that even minute differences in prices count, thus, in general, for each stock it would be necessary to pick the best available model.

To validate the results, two well-known benchmark prices will be considered for comparison purposes. The first one is the VWAP price assuming knowledge of the traded volume (which has to be forecasted in practice). This price is computed as

$$\text{VWAP(t)} = \frac{\sum_{k=1}^{N_p} v(t+k)p(t+k)}{\sum_{k=1}^{N_p} v(t+k)}. \quad (22)$$

TABLE 2. Average prices in the validation set (RMB) using the proposed approach with each of the prediction models. The best value for each stock highlighted in bold.

Stock	NN-NN	LD-LD	LD-NN	NN-LD
002040	19.831	19.799	**19.798**	19.835
300239	5.262	5.233	**5.232**	5.261
300033	65.003	64.936	**64.913**	65.004
688012	188.706	**187.076**	187.100	188.700
002466	42.972	42.935	**42.933**	42.969
600000	12.777	12.754	**12.753**	12.776
000333	51.746	**51.703**	51.709	51.745
000001	12.169	12.130	**12.129**	12.170
600519	658.287	**656.885**	656.890	658.288

TABLE 3. Validation of the results by comparing to benchmark prices (RMB).

Stock	Proposed (historical)	Proposed (forecasted)	VWAP	TWAP
002040	19.619	19.798	19.837	19.814
300239	5.211	5.232	5.245	5.242
300033	64.243	64.913	65.013	64.950
688012	183.411	187.076	188.341	187.799
002466	42.514	42.933	42.965	42.962
600000	12.732	12.753	12.771	12.769
000333	51.421	51.703	51.806	51.774
000001	12.076	12.129	12.161	12.150
600519	653.808	656.885	657.608	657.630

The second baseline strategy is the TWAP price, which is the average price of the stock over the time window, that is,

$$\mathrm{TWAP}(t) = \sum_{k=1}^{N_p} \frac{p(t+k)}{N_p}. \tag{23}$$

Furthermore, the price attained with the proposed strategy if historical data, i.e., not forecasted, are used will be also considered. This price cannot be achieved in practice, but it is useful to find the maximum performance that could be reached in the absence of prediction errors, and therefore properly calibrate their impact on the performance of the strategy. Table 3 shows the average prices over the validation set using both historical and forecasted prices and volumes, along with the VWAP and TWAP prices.

It can be seen that the proposed strategy obtains a lower price than VWAP or TWAP in all the validation sets.[3] Clearly, the prediction error in the price and volume forecasting reduces the difference, but even with forecasted prices and volumes the proposed strategy obtains lower prices than the baseline approaches. Regarding the capitalization of the stocks, it appears that the proposed strategy gets worse results in the case of small-cap stocks, but this could be related to the differences in the price per share, getting better results in stocks with a higher price per share. On the other hand, it is quite remarkable that for some of the stocks, even the prices obtained with historical data are very close to VWAP or TWAP, e.g. stocks with ticker 300239 and 600000. In these cases, there is clearly a very small margin of improvement, meaning that in these cases it proved very difficult to beat the market on the validation set. There are other stocks, like 688012, in which the hypothetical benefit that could be obtained is greatly wasted by the prediction errors. Regarding this, it is noteworthy that the prediction horizon is quite large in this case study, forcing the forecast up to 120 steps ahead. This is much longer than usual, given the fact that most of the stock forecasting applications focus on one step ahead predictions. On the other hand, higher prediction errors do not always imply a worse result as long as the price trend is correctly predicted. The reason for this is that in the proposed strategy price forecasting accuracy is not what really allows better results, but the ability to forecast the position of the lower prices in the time window. Finally, even if the price difference attained in practice seems small, given the high number of shares traded in this type of orders, the overall benefit can be significant. To illustrate this, consider table 4 in which the savings of the proposed strategy over VWAP and TWAP are shown for the case of using the real or forecast prices and volumes.

[3]Note that, in the simulation, the impact of each suborder in the market has not been taken into account, as this impact is very difficult to model (the term in the performance function (1) acts as a de-tuning factor and does not constitute an accurate model of market impact). Thus, in practice, the results could be even better provided that the parameters α and β are properly chosen and adapted to the market conditions.

TABLE 4. Savings on the validation set over VWAP and TWAP when using historical or forecasted data (RMB).

	Savings over VWAP		Savings over TWAP	
Stock	Historical	Forecasted	Historical	Forecasted
002040	8,720,000	1,560,000	7,800,000	640,000
300239	1,360,000	520,000	1,240,000	400,000
300033	30,800,000	4,000,000	28,280,000	1,480,000
688012	197,200,000	50,600,000	175,520,000	28,920,000
002466	18,040,000	1,280,000	17,920,000	1,160,000
600000	1,560,000	720,000	1,480,000	640,000
000333	15,400,000	4,120,000	14,120,000	2,840,000
000001	3,400,000	1,280,000	2,960,000	840,000
600519	152,000,000	28,920,000	152,880,000	29,800,000

It can be seen that the savings are substantial in most cases and very substantial in some cases like the stocks with tickers 688012 and 600519. Furthermore, considering the savings that can be obtained over the many large orders that are traded over a year, it is evident that even a modest improvement on the price can justify the use of strategies like the proposed in this paper.

The proposed approach has also been validated when using limit orders with the case study. Table 5 shows the average prices and the percentage of order execution in the validation set for all the stocks. Different price limits have been set for each session in the validation set according to the price levels so that the price limit is not very stringent but not trivial to meet (which would result, in practice, in a *de facto* market order). Regarding the benchmark price comparison, it is noteworthy that in the case of limit orders the TWAP and VWAP prices cannot be considered, as they do not consider the limit in the price. Instead, TWAP limit and VWAP limit orders have been used as benchmarks. These orders work in the same way as conventional TWAP and VWAP orders except they are forced to meet the price limit. Thus, a TWAP or VWAP limit order cannot be started until the price limit is met, and they have to be stopped whenever the current price does not meet the limit. If the limit is met

TABLE 5. Average prices and percentage of execution in the case of limit orders.

	Average prices over the validation set					Average execution over the validation set (%)			
Stock	Limit	Historical	Forecasted	TWAP	VWAP	Historical	Forecasted	TWAP	VWAP
002040	19.779	19.608	19.691	19.687	19.683	96.000	86.706	64.439	60.621
300239	5.239	5.203	5.216	5.210	5.211	100.000	96.447	62.255	56.667
300033	64.770	64.230	64.461	64.459	64.452	100.000	87.960	77.006	74.983
688012	187.815	183.408	185.597	185.824	185.782	100.000	97.615	80.882	79.417
002466	42.995	42.490	42.733	42.728	42.729	100.000	96.500	81.296	80.390
600000	12.789	12.721	12.735	12.734	12.734	100.000	94.759	56.250	55.353
000333	51.597	51.380	51.482	51.467	51.466	98.000	92.550	72.187	70.989
000001	12.145	12.069	12.105	12.094	12.095	100.000	95.582	68.606	66.519
600519	658.275	653.808	655.556	655.861	655.989	100.000	94.544	74.467	74.571

TABLE 6. Limit orders, stock 002040.

	Prices per session					Percentage of order executed			
Session	Limit	Historical	Forecasted	TWAP	VWAP	Historical	Forecasted	TWAP	VWAP
1	20.800	20.648	20.715	20.731	20.721	100.000	100.000	58.240	56.055
2	20.300	20.109	20.153	20.182	20.174	100.000	100.000	81.120	76.765
3	18.750	18.705	18.719	18.731	18.732	100.000	100.000	72.780	49.605
4	19.900	19.866	19.866	19.818	19.812	70.000	70.000	29.090	26.055
5	21.000	20.820	20.966	20.937	20.927	100.000	95.585	85.625	82.520
6	18.780	18.741	18.758	18.743	18.739	100.000	100.000	91.590	73.225
7	19.750	19.665	19.739	19.711	19.714	100.000	100.000	100.000	100.000
8	19.350	19.284	19.315	19.318	19.313	100.000	95.150	67.050	64.645
9	19.300	19.187	19.201	19.251	19.253	100.000	100.000	89.750	89.240
10	19.100	18.910	19.008	19.014	19.003	100.000	100.000	69.840	69.645
11	20.400	20.192	20.207	20.283	20.292	100.000	100.000	81.440	81.050
12	20.750	20.540	20.634	20.658	20.661	100.000	100.000	100.000	100.000
13	18.800	18.756	18.776	18.764	18.764	100.000	100.000	46.320	35.520
14	19.100	18.719	18.744	18.779	18.779	100.000	60.535	53.950	50.785
15	19.000	18.764	18.796	18.811	18.811	100.000	90.000	19.950	16.655
16	17.300	17.221	17.261	17.259	17.261	100.000	100.000	94.600	92.990
17	21.700	20.650	21.592	21.393	21.333	100.000	100.000	97.460	99.310
18	20.800	20.791	20.780	20.778	20.778	100.000	10.000	5.745	5.060
19	21.200	21.116	21.105	21.095	21.121	50.000	20.000	3.320	7.510
20	19.500	19.471	19.481	19.475	19.477	100.000	92.845	40.915	35.785

again, a new TWAP or VWAP limit order is started for the remaining quantity to be executed and set to be executed in the remaining time window. As a result, TWAP and VWAP limit orders have no guarantee of being executed completely and their success can be judged by the percentage of the original order that it is executed. Thus, when comparing the proposed strategy to these benchmarks, the price should be taken into account only if the percentage of execution is the same. Note, however, that the attained price must always meet the price limit or the order will be considered failed (and the brokerage firm forced to pay the excess over the limit). The results in table 5 show that the proposed strategy manages to reach a higher percentage of order execution than TWAP or VWAP limit orders while meeting the price limit. It is clear that the forecasting errors induce negative effects in both the price and execution percentage attained, but even using forecasted data the proposed strategy beats both TWAP and VWAP limit orders while meeting the price limit.

Limit orders are more complex than market orders, thus it is interesting to examine in more detail the results shown in table 5. For conciseness, the results of only three stocks are shown in tables 6 to 8 (these are the same stocks whose dataset is shown figures 1 to 3). It can be seen that the proposed approach usually achieves higher percentages of execution, although prediction errors can have a great impact on this percentage, e.g., session 18 in stock 002040 has 100% execution when using historical data but only 10% when using forecasted data. Even though, the latter result nearly doubles that of TWAP and VWAP limit orders. Another fact that can be observed is that there are sessions in which all the approaches are able to complete the orders. In these situations, the proposed approach using historical data achieves a better price, but when using forecasted data the price can be worse (e.g., session 7 in stock 002040 or session 6 in stock 688012 in which the results are better than the TWAP limit but worse than the VWAP limit). This is to be expected, as no approach can be the best in all situations in a task as difficult as trying to beat the market. Nevertheless, the proposed approach show on average better results (as illustrated in tables 5 to 8). Furthermore, there are sessions in which the proposed approach, even with forecasted data, obtains a better price and a better percentage of execution (e.g., session 9 in stock 002040 or session 17 in stock 688012). On the other hand, the price limit can be so stringent that even with historical data is not possible to execute completely the order. This can be seen in sessions 4 and 19 of stock 002040 or in session 3 of stock 000333. Note that in those cases the proposed approach reached higher execution percentages than the benchmarks.

Finally, regarding the computational burden of the optimization algorithm, the average time to solve the optimization problem in a serial Matlab implementation on a budget mobile processor Core i5-8265U clocked at 1.6 GHz is just 1.2 milliseconds. This is much lower than the time bucket but

TABLE 7. Limit orders, stock 688012.

	Prices per session					Percentage of order executed			
Session	Limit	Historical	Forecasted	TWAP	VWAP	Historical	Forecasted	TWAP	VWAP
1	168.000	160.609	166.788	166.039	165.637	100.000	100.000	63.365	77.300
2	149.500	147.929	148.404	148.524	148.513	100.000	100.000	100.000	100.000
3	152.500	151.523	151.871	151.944	152.001	100.000	100.000	46.200	37.060
4	161.500	160.307	160.947	160.837	160.849	100.000	100.000	85.590	74.865
5	227.000	217.287	225.592	224.088	223.536	100.000	90.000	59.025	57.660
6	157.800	156.068	157.081	157.110	157.021	100.000	100.000	100.000	100.000
7	160.000	155.438	156.087	156.997	156.836	100.000	100.000	48.970	50.590
8	212.000	211.776	211.803	211.774	211.756	100.000	100.000	100.000	95.820
9	262.000	258.173	258.577	260.337	260.362	100.000	100.000	100.000	100.000
10	230.000	225.664	227.864	228.214	228.259	100.000	100.000	100.000	100.000
11	158.000	148.735	152.784	153.446	152.712	100.000	99.510	75.330	71.310
12	173.000	172.347	172.589	172.636	172.672	100.000	100.000	93.050	85.820
13	168.000	166.838	166.919	167.037	167.002	100.000	100.000	100.000	100.000
14	194.000	191.795	192.823	192.666	192.678	100.000	100.000	94.465	93.495
15	185.000	175.753	178.912	180.190	180.037	100.000	100.000	75.595	79.345
16	154.000	153.086	153.460	153.503	153.536	100.000	100.000	68.470	66.405
17	221.000	215.824	217.787	219.085	218.919	100.000	100.000	71.635	79.915
18	203.000	199.054	200.579	202.048	202.036	100.000	100.000	100.000	100.000
19	240.000	229.513	237.794	235.303	236.170	100.000	62.795	56.260	40.695
20	180.000	170.440	173.286	174.707	175.115	100.000	100.000	79.680	78.070

TABLE 8. Limit orders, stock 000333.

	Prices per session					Percentage of order executed			
Session	Limit	Historical	Forecasted	TWAP	VWAP	Historical	Foreacasted	TWAP	VWAP
1	51.500	51.110	51.321	51.385	51.372	100.000	100.000	56.345	52.220
2	59.500	59.192	59.306	59.314	59.307	100.000	100.000	100.000	100.000
3	54.800	54.778	54.778	54.772	54.775	60.000	40.000	32.070	39.765
4	58.000	57.774	57.920	57.890	57.883	100.000	81.220	43.575	42.910
5	51.400	50.933	51.128	51.075	51.079	100.000	100.000	88.435	86.865
6	54.500	54.448	54.446	54.436	54.434	100.000	100.000	92.835	78.000
7	41.400	41.327	41.333	41.338	41.338	100.000	84.930	61.460	72.495
8	39.150	38.803	39.069	38.984	38.996	100.000	100.000	52.685	63.950
9	40.300	40.195	40.271	40.230	40.221	100.000	100.000	100.000	96.780
10	71.100	70.945	70.981	71.000	70.990	100.000	100.000	79.055	76.065
11	51.800	51.640	51.743	51.718	51.728	100.000	100.000	36.585	28.290
12	51.800	51.769	51.768	51.766	51.765	100.000	94.860	46.230	36.660
13	51.000	50.782	50.838	50.898	50.894	100.000	100.000	100.000	100.000
14	54.100	54.039	54.050	54.057	54.054	100.000	100.000	86.835	81.845
15	50.800	50.623	50.692	50.681	50.681	100.000	100.000	28.375	25.195
16	47.200	47.004	47.096	46.946	46.932	100.000	100.000	100.000	100.000
17	40.600	40.169	40.498	40.385	40.385	100.000	50.000	39.260	38.745
18	53.000	52.946	52.984	52.961	52.956	100.000	100.000	100.000	100.000
19	60.900	60.635	60.657	60.735	60.736	100.000	100.000	100.000	100.000
20	49.100	48.482	48.766	48.777	48.793	100.000	100.000	100.000	100.000

could be improved with a more powerful computer or with a more efficient implementation, like a parallel C++ one using the NVIDIA CUDA GPU computing platform. Note that the algorithm relies only on simple operations that can be easily coded in a low-level programming language.

V. CONCLUSION

In this paper, a trading strategy has been proposed for large orders that have to be split to minimize the impact on the market. The strategy is based on the well-known receding horizon optimization scheme used in predictive controllers. The proposed algorithm can handle market and limit orders and, although relies on forecasting the future prices and traded volumes, is independent of the forecasting method used. The strategy has been validated with an assorted set of Chinese stocks of different capitalization levels. The results have been very positive despite the fact that the prediction horizon is quite large. The amount of potential savings over well-known trading strategies is quite relevant when the size of orders is taken into account. Moreover, from a risk management point of view, the savings are related to improved execution risk, and does not have any impact on the investment risk. This is due to the fact that savings are carried out after the investment decision is made, therefore there are no effects on investment risk.

The proposed strategy will certainly benefit from future research on how to reduce the impact of prediction errors. In this regard, most forecasting techniques are focused on one-step price prediction, whereas the needs of the proposed strategy are quite different. The topic of research would be to develop methods to forecast the position of most advantageous time moments, that is, those with the lowest prices (or greatest in sell orders), on a given time window, rather than predicting the prices themselves. Also, it would be interesting to modify the strategy to be able to use prediction horizons shorter than the time window without affecting the performance of the strategy.

REFERENCES

[1] *Marching up the Learning Curve: The Second Buy-Side Algorithmic Trading Survey*, Bank of America, Charlotte, NC, USA, 2007.

[2] H. White, "Economic prediction using neural networks: The case of IBM daily stock returns," in *Proc. ICNN*, vol. 2, 1988, pp. 451–458.

[3] J. Wang, T. Sun, B. Liu, Y. Cao, and D. Wang, "Financial markets prediction with deep learning," in *Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2018, pp. 97–104.

[4] D. Hasibuan, I. K. Jaya, B. Rumahorbo, J. Naibaho, J. Napitupulu, and E. Rajagukguk, "Time series financial market forecasting based on support vector regression algorithm," in *Proc. Int. Conf. Comput. Sci. Inf. Technol. (ICoSNIKOM)*, Nov. 2019, pp. 1–4.

[5] T. B. Trafalis and H. Ince, "Support vector machine for regression and applications to financial forecasting," in *Proc. IEEE-INNS-ENNS Int. Joint Conf. Neural Netw. Neural Comput.*, 2000, pp. 348–353.

[6] E. El, "A new particle swarm optimization based stock market prediction technique," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 4, p. 4, 2016.

[7] G. Alfonso, A. D. Carnerero, D. R. Ramirez, and T. Alamo, "Stock forecasting using local data," *IEEE Access*, vol. 9, pp. 9334–9344, 2021.

[8] W. Cui, A. Brabazon, and M. O'Neill, "Efficient trade execution using a genetic algorithm in an order book based artificial stock market," in *Proc. 11th Annu. Conf. Companion Genetic Evol. Comput. Conf.O*, 2009, pp. 2023–2028.

[9] M. Lim and R. J. Coggins, "Optimal trade execution: An evolutionary approach," in *Proc. IEEE Congr. Evol. Comput.*, vol. 2, Sep. 2005, pp. 1045–1052.

[10] M. Dixon, "Sequence classification of the limit order book using recurrent neural networks," *J. Comput. Sci.*, vol. 24, pp. 277–286, Jan. 2018.

[11] J. Białkowski, S. Darolles, and G. Le Fol, "Improving VWAP strategies: A dynamic vol. approach," *J. Banking Finance*, vol. 32, no. 9, pp. 1709–1722, 2008.

[12] H. Konishi, "Optimal slice of a VWAP trade," *J. Financial Markets*, vol. 5, no. 2, pp. 197–221, Apr. 2002.

[13] B. Crawford, R. Soto, M. A. San Martín, H. de la Fuente-Mella, C. Castro, and F. Paredes, "Automatic high-frequency trading: An application to emerging Chilean stock market," *Scientific Program.*, vol. 2018, pp. 1–12, Sep. 2018.

[14] Y. Ding, L. Cheng, W. Pedrycz, and K. Hao, "Global nonlinear kernel prediction for large data set with a particle swarm-optimized interval support vector regression," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2521–2534, Oct. 2015.

[15] M. Pemy, "Optimal algorithms for trading large positions," *Automatica*, vol. 48, no. 7, pp. 1353–1358, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109812001690

[16] A. W. Stace, "A moment matching approach to the valuation of a vol. weighted, average price option," *Int. J. Theor. Appl. Finance*, vol. 10, no. 01, pp. 95–110, 2007, doi: 10.1142/S0219024907004068.

[17] D. Bertsimas and A. W. Lo, "Optimal control of execution costs," *J. Financial Markets*, vol. 1, no. 1, pp. 1–50, Apr. 1998.

[18] P. A. Forsyth, "A Hamilton–Jacobi–Bellman approach to optimal trade execution," *Appl. Numer. Math.*, vol. 61, no. 2, pp. 241–265, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168927410001789

[19] P. Forsyth, J. Kennedy, S. Tse, and H. Windcliff, "Optimal trade execution: A mean quadratic variation approach," *J. Econ. Dyn. Control*, vol. 36, no. 12, pp. 1971–1991, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165188912001236

[20] D. Mitchell, J. BiaÅkowski, and S. Tompaidis, "Volume-weighted average price tracking: A theoretical and empirical study," *IISE Trans.*, vol. 52, no. 8, pp. 864–889, 2020, doi: 10.1080/24725854.2019.1688896.

[21] X. Ye, R. Yan, and H. Li, "Forecasting trading volume in the chinese stock market based on the dynamic VWAP," *Stud. Nonlinear Dyn. Econometrics*, vol. 18, no. 2, pp. 125–144, Jan. 2014.

[22] X. Liu and K. K. Lai, "Intraday volume percentages forecasting using a dynamic SVM-based approach," *J. Syst. Sci. Complex.*, vol. 30, no. 2, pp. 421–433, Apr. 2017.

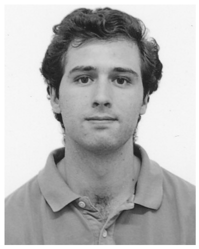
[23] H. Li and X. Ye, "A dynamic, volume-weighted average price approach based on the fast Fourier transform algorithm," *Asia–Pacific J. Financial Stud.*, vol. 42, no. 6, pp. 969–991, Dec. 2013.

[24] J. H. Song, M. L. de Prado, H. D. Simon, and K. Wu, "Exploring irregular time series through non-uniform fast Fourier transform," in *Proc. 7th Workshop High Perform. Comput. Finance*, Oct. 2014, pp. 37–44.

[25] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, vol. 2. Madison, WI, USA: Nob-Hill, 2017.

[26] R. F. Almgren, "Optimal execution with nonlinear impact functions and trading-enhanced risk," *Appl. Math. Finance*, vol. 10, no. 1, pp. 1–18, Jan. 2003.

GERARDO ALFONSO received the B.Eng. degree in electrical and electronics engineering from the University of Leicester, U.K., the M.B.A. degree from the University of Cambridge, U.K., and the Ph.D. degree in finance from the University Complutense de Madrid, Spain. He was a Postdoctoral Researcher with the University of Cambridge. He has more than a decade of experience in trading in capital markets in USA, Europe, and Asia. He is currently an Executive Director with Shenwan and Hongyuan Securities Company Ltd., Shanghai.



A. DANIEL CARNERERO received the M.Eng. degree in industrial engineering from the University of Seville, in 2019. He is currently pursuing the Ph.D. degree with the Department of Systems Engineering and Automation, University of Seville. His current research interests include GPU computing, metaheuristic optimization, model predictive control, randomized algorithms, and data-driven methods.

DANIEL R. RAMIREZ was born in Spain, in 1972. He received the M.Eng. and Ph.D. degrees in computer engineering from the University of Seville, Spain, in 1996 and 2002, respectively. From 1997 to 1999 he was a Research Assistant with the Department of System Engineering and Automatic Control. Since 1999, he has been an Assistant Professor and an Associate Professor with the Department of System Engineering and Automatic Control. He has authored or coauthored more than 70 technical articles in international journals and conference proceedings. His current research interests include randomized algorithms, model predictive control, data-based forecasting, and soft computing techniques.

TEODORO ALAMO was born in Spain, in 1968. He received the M.Eng. degree in telecommunications engineering from the Polytechnic University of Madrid, Madrid, Spain, in 1993, and the Ph.D. degree in telecommunications engineering from the University of Seville, Seville, Spain, in 1998. Part of his Ph.D. work was done at RWTH Aachen, Aachen, Germany, from June to September 1995. From September 1991 to May 1993, he was with the Ecole Nationale Superieure des Télécommunications (Télécom Paris), Paris, France. From 1993 to 2000, he was an Assistant Professor with the Department of System Engineering and Automatic Control, University of Seville, where he was an Associate Professor, from 2001 to 2010, and has been a Full Professor, since March 2010. He has Co-Founded the spin-off company Optimal Performance, University of Seville. He is the author or coauthor of more than 200 publications, including books, book chapters, journal articles, conference proceedings, and educational books. His current research interests include decision-making, model predictive control, data-driven methods, randomized algorithms, and optimization strategies.