972 Project - GRAB Research Project

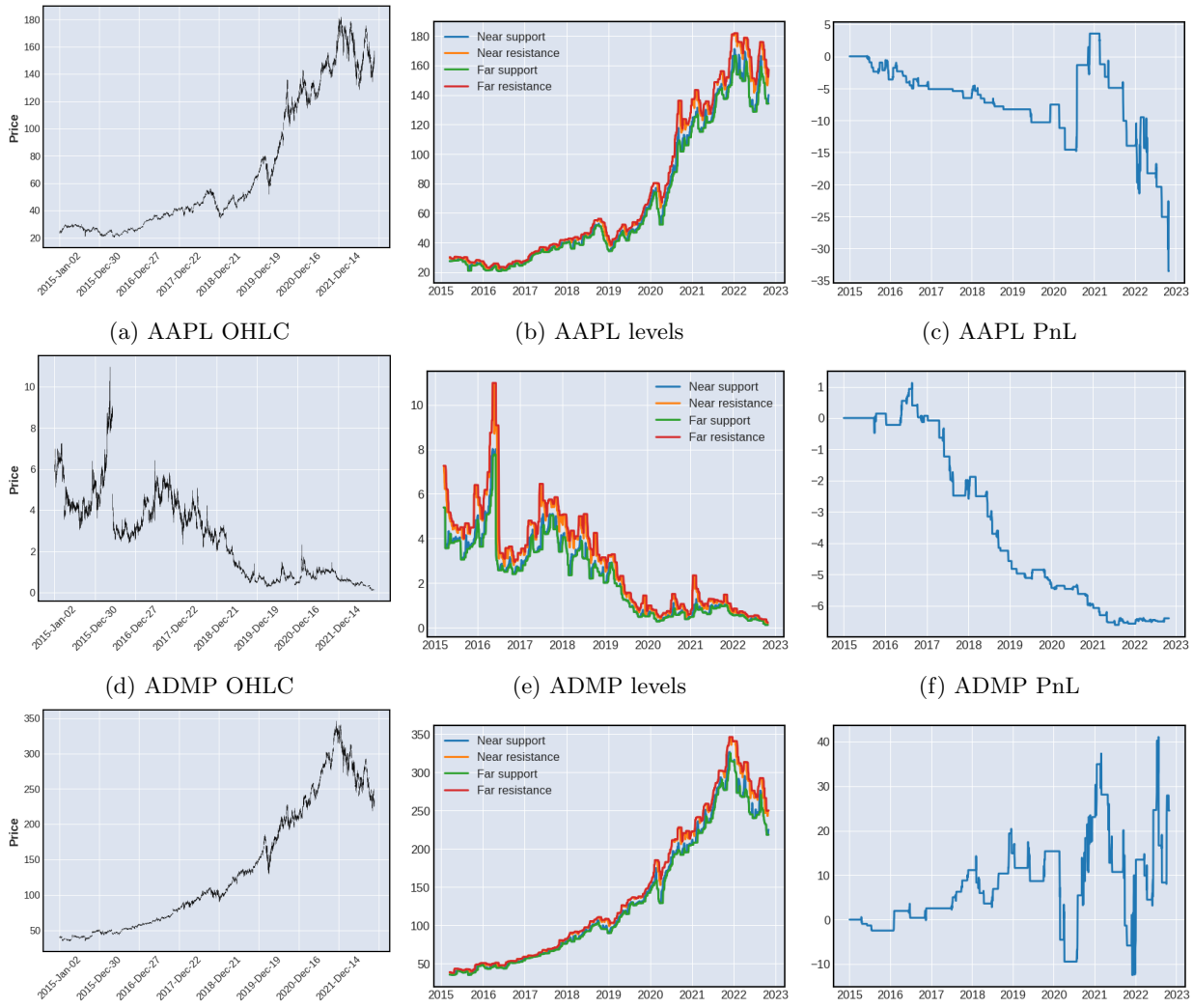
Yunfeng Liu

November 19, 2022

1 Research Project

1.1 Replication Analysis

To verify if the GRAB algorithm works, I back tested GRAB strategy with AAPL,ADMP, MSFT, TSLA and SPY from 2015-01-01 to 2022-11-01. The GRAB parameters used are $N_n = 10$, $N_f = 20$. There are some modifications on the original algorithm. In this replication, whenever a limit order is required, I used current date's closed price instead. Moreover, transaction fees and borrowing costs are also ignored.

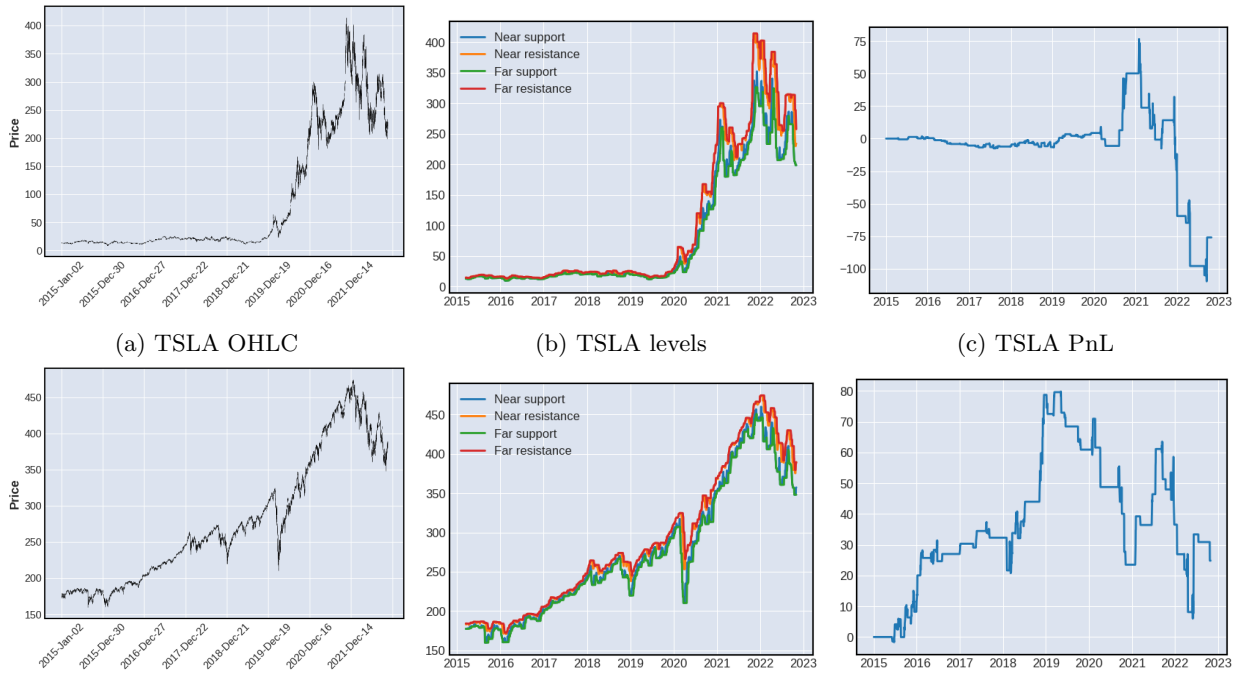


(a) AAPL OHLC (b) AAPL levels (c) AAPL PnL

(d) ADMP OHLC (e) ADMP levels (f) ADMP PnL

(g) MSFT OHLC (h) MSFT levels (i) MSFT PnL

Figure 1: AAPL, ADMP, MSFT



(a) TSLA OHLC (b) TSLA levels (c) TSLA PnL

(d) SPY OHLC (e) SPY levels (f) SPY PnL

Figure 2: TSLA, SPY

Figure 1 and 2 are the results of back testing the GRAB algorithm on different instruments. The PnL charts on the right for all instruments are the portfolio value process over time. As we noticed, with zero initial portfolio value, AAPL, ADMP, TSLA all end up with losing money and MSFT, SPY end up with profits but not significant. The volatility of the PnL varies from one instrument to another.

1.2 Strengths and Weaknesses

There are couple of weaknesses observed by the author:

1. Sometimes, the far and near term lines are the same. This happens when the N_f and N_n are close to each other. In this case, the system wants to buy and sell at the same time.

   In my opinion, such cases can be prevented. If the trend flipped and the far and near support levels are equal, the GRAB system just has to sell the share if the trader is holding any. If the far and near resistance levels are equal, the trader just has to buy back a share if there's a shorted position open.

For example, if the trend flipped from up to down and the far support coincide with near support, then the system should sell the holding share if the trader is holding one. And when trend flipping did happened, GRAB should skip swing trade until the next tick price is updated.

2. A losing trade does not accompany every major trend reversal, as not all trend reversal trades enter. And not all trades within the major trend are profitable. Sometimes the exit level (near resistance level) moves below the entry level during the life time. This is more likely to occur when the far and near parameters N_f, N_n are further apart.

   What the author means is that not all losing trade happen at the trend reversal times, but also during major trend period from the swing trades. This happens if the near term band moves up and down within a much broader far term band and cause the sell price lower than the entry price. However, such issue can be minimized by choosing the N_n closer to N_f thus narrowing the gaps between far and near term levels. However, such solution needs a more scientific method to further quantify it.

I do not see any strength of the GRAB algorithm, but I do observed other weaknesses in GRAB:

1. Exit trades are sent via limit orders, which means there's risk that the limit order never get fulfilled. For example, when the trend is flipping from up to down, the system wants to exit the holding instrument at near resistance level, which may not be executed for a long period of time.

2. The exit procedures described by the author is ill-designed. For example, because $K_r^f \geq K_r^n > K_s^n$, when a trend is flipped from "down" to "up", the spot price is greater or equal to K_r^f which means buy limit at K_s^n is unlikely to be executed and setting another buy stop at K_r^f will invalidate the previously set buy limit at K_s^n. Same issue also exists in the other direction of trend flipping.

3. The market is dynamic. Major events and section trends could potentially altered the price trend patterns. Fixing N_f, N_n parameters could cause entering and exiting a trade to be out of sync with market timing.

4. There's no scientific reasoning and assumption made on why the algorithm choose look-back maximum

and minimum levels as major trends indicator. Therefore, we should be questioning the effectiveness using such measures.

2 Improvement

2.1 Optimal parameters selection

The trade process $G(t; N_f, N_n, T)$ is a function of two parameters N_f and N_n, and instead of fixing these two parameters during the entire trading period, one can use dynamic parameters at each time t using the optimal N_f^* and N_n^*. The optimal N_f^* and N_n^* are defined as follows:

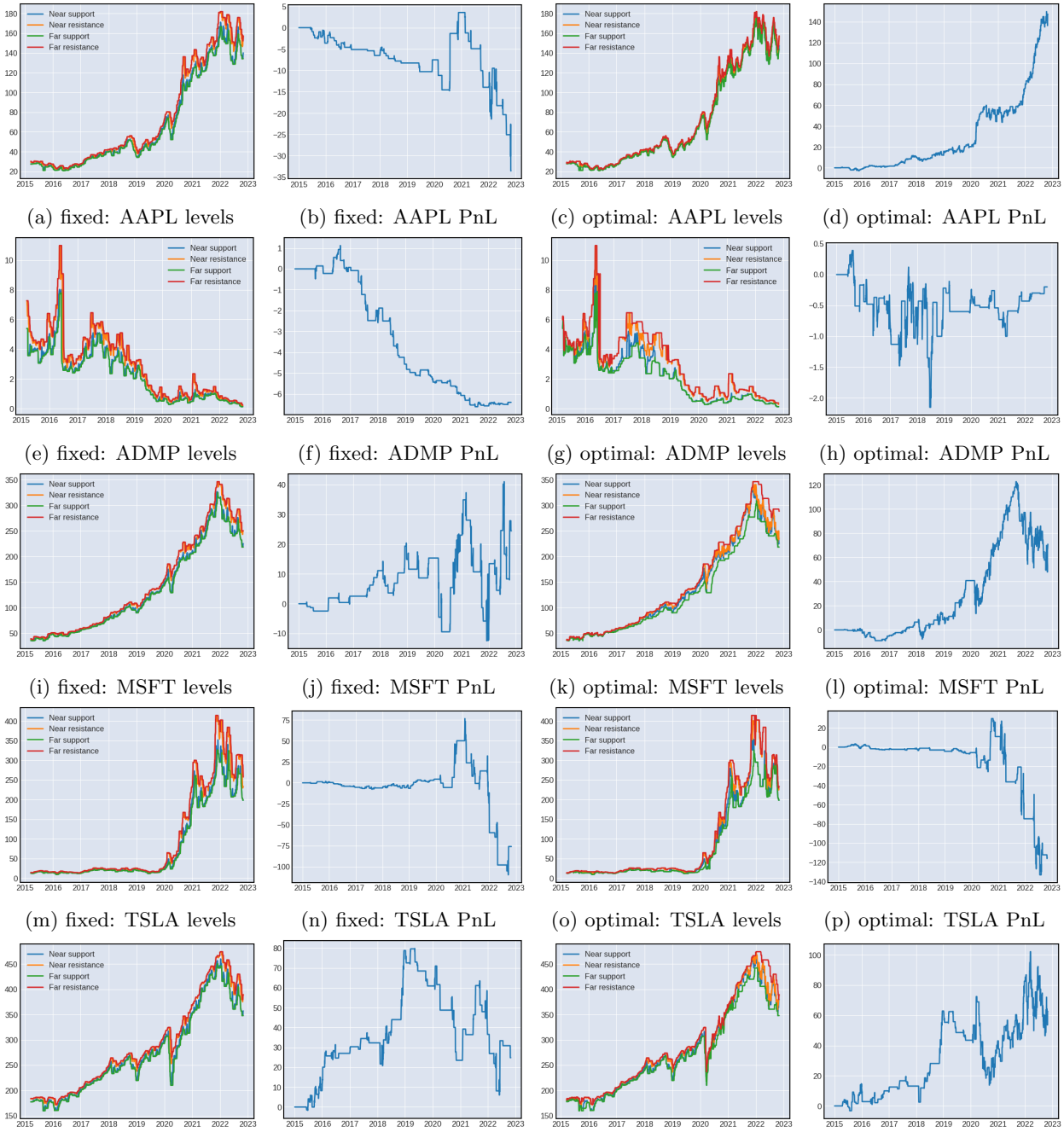
$$N_f^*, N_n^* = \underset{0<N_n<N_f<M}{\operatorname{argmax}} G(t; N_f, N_n, T) \tag{1}$$

where M is the the maximum bound allowed.

The idea is that we N_n and N_f such that the next trade its historical performance outperforms other N_n, N_f combinations up to time t. I tested this optimal strategy with the same instruments and compared them with the fixed $N_n = 10$, $N_f = 20$ strategy. In each row of the charts in Figure 3, first two charts are the fixed parameter strategy and the next two are optimal parameter strategy. (The source code can be found at https://github.com/YunfengLiu/ACTSC-972-project)

We observed that AAPL that is a losing trade using fixed parameters has become a profitable trade with the optimal parameter method. ADMP has reduces the lost of the PnL by choosing the optimal parameter method, however the volatility of the new portfolio increased dramatically in this case. The support and resistance levels are narrower than the fixed parameter case. In fact, most the optimal parameters produced on the 5 instruments are less than or equal to 10, which means the best GRAB parameters are those that look back only few days in history and older market quotes are irrelevant in most cases.

Although the optimal parameter method looks promising, we can only conclude that the optimal parameter method can improve the GRAB's PnL in some cases. Further research on improving the optimal parameter method can be conducted in the future.



(a) fixed: AAPL levels (b) fixed: AAPL PnL (c) optimal: AAPL levels (d) optimal: AAPL PnL

(e) fixed: ADMP levels (f) fixed: ADMP PnL (g) optimal: ADMP levels (h) optimal: ADMP PnL

(i) fixed: MSFT levels (j) fixed: MSFT PnL (k) optimal: MSFT levels (l) optimal: MSFT PnL

(m) fixed: TSLA levels (n) fixed: TSLA PnL (o) optimal: TSLA levels (p) optimal: TSLA PnL

(q) fixed: SPY levels (r) fixed: SPY PnL (s) optimal: SPY levels (t) optimal: SPY PnL

Figure 3: AAPL, ADMP, MSFT, TSLA, SPY