1. **How to fix the bug**
   1. Update *quantity\_behind* after aggressive trading instead of *quantity\_remaining*
   2. Correct the fair value function: for a buy order, the sign for schedule coefficient should be positive; for a buy order, the sign for schedule coefficient should be negative.
2. **How to force the order to complete the exact number of shares**
   1. Adjust the order type in the last-minute bar: aggressively trade all orders that haven’t been filled.
   2. Make sure that passive trading and aggressive trading do not occur in the same tick to eliminate duplicate indexing problems. A passive order might be filled as the same tick with an aggressive order. We eliminate this problem by serving passive order first and wait for the next tick to execute the aggressive order.
3. **Simplified Trading Logic**

Update target order quantity every minute

Update shares behind the target

Update new tick signal to window once a trade occurs

Calculate the tick factor

Calculate the schedule factor

Calculate fair value for every tick

sell order

buy order

Fair value last ask price

Fair value last ask price

N

Y

N

Y

Trade aggressively

Trade aggressively

Trade passively

Trade passively

Round lots of shares behind target quoted at last ask price

Round lots of shares behind target immediately traded at last bid price

Round lots of shares behind target quoted at last bid price

Round lots of shares behind target immediately traded at last ask price

Get filled by upcoming qualified trades

1. **The Factor Model**
   1. Create the VWAP profile. Select historical minute data of “AAPL” from 03/02/2020 to 03/11/2020 and filter out the non-trading days. Apply a linear regression model to fit the relationship between the minute bar and the cumulative trading volume. Save the model coefficients to simulate the target trading volume in the testing days.
   2. Create the TAQ file for the testing days. Download tick quotes and tick trades of “AAPL” on 03/11/2020, 03/12/2020 and 03/13/2020 from WRDS TAQ Database. Filter out quotes and trades that happen outside regular trading hours, and keep only the NBBO quotes.
   3. The schedule-factor-only model
      1. Determine the parameters to optimize. There are two parameters that needs to optimize: (a) schedule factor, (b) schedule coefficient. Since the schedule factor only depends on the predetermined maximum behinds of shares from target shares, we replace (a) with *max\_behinds*.
      2. Find out the best combination of *max\_behinds* and *schedule\_coefficient.* Start with the buy order side.Test different combinations of the two parameters on the three testing days. Calculate the average slippage over the three trading days. The combination with the smallest slippage is kept as the best parameter combination. Switch side from buy to sell, and repeat the preceding procedure. Save the best parameter combination for the sell side.
   4. The schedule-factor and tick-factor model
      1. Determine the parameters to optimize. For the new model including tick factor, there are three new parameters to optimize: (a) tick window (b) tick coefficient (c) schedule coefficient. For the sake of simplicity, we adopt the best *max\_behinds* from the previous schedule-factor-only model.
      2. Find out the best combination of the three parameters in 4.1 using the similar methods in 3.2.
2. **Limit order**
   1. Add additional constraints in every trade.

|  |  |  |
| --- | --- | --- |
|  | Aggressive trade | Passive trade |
| Buy order | Execute the trade only when last ask price limit bid price | Quote only when last bid price limit bid price |
| Sell order | Execute the trade only when last bid price limit ask price | Quote only when last ask price limit ask price |

Note:

For limit price orders, there’s no guarantees that all initial order quantity be filled. If the price is not favorable, a trade will not be executed.

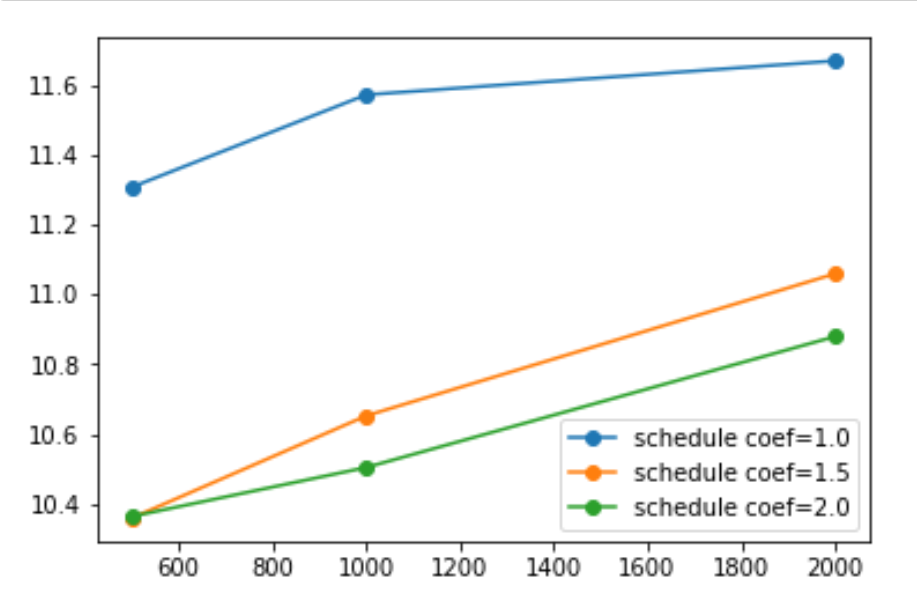
To avoid dumping too much shares in a single trade (if the stock price travels a long time until it reaches the limit price, there remains significant large number of shares to be executed), we also limit the maximum quantity of shares to be executed per trade.

1. Conclusion

From the schedule-factor-only buy side model, we find that the best combination of parameter is: max behind=500, schedule coefficient=1.5

average slippage of schedule-factor-only buy-side model

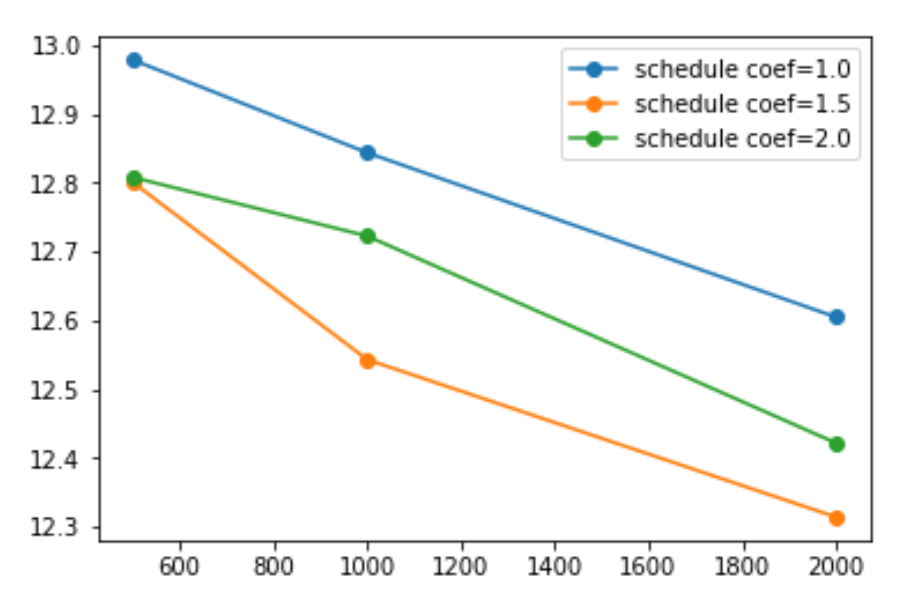
| schedule coef  max behind | 1.0 | 1.5 | 2.0 |
| --- | --- | --- | --- |
| 500 | 11.306471 | 10.359066 | 10.363807 |
| 1000 | 11.571015 | 10.652066 | 10.504120 |
| 2000 | 11.668206 | 11.058983 | 10.879131 |



From the schedule-factor-only sell-side model, we find that the best combination of parameter is: max behind=2000, schedule coefficient=1.5

average slippage of schedule-factor-only sell-side model

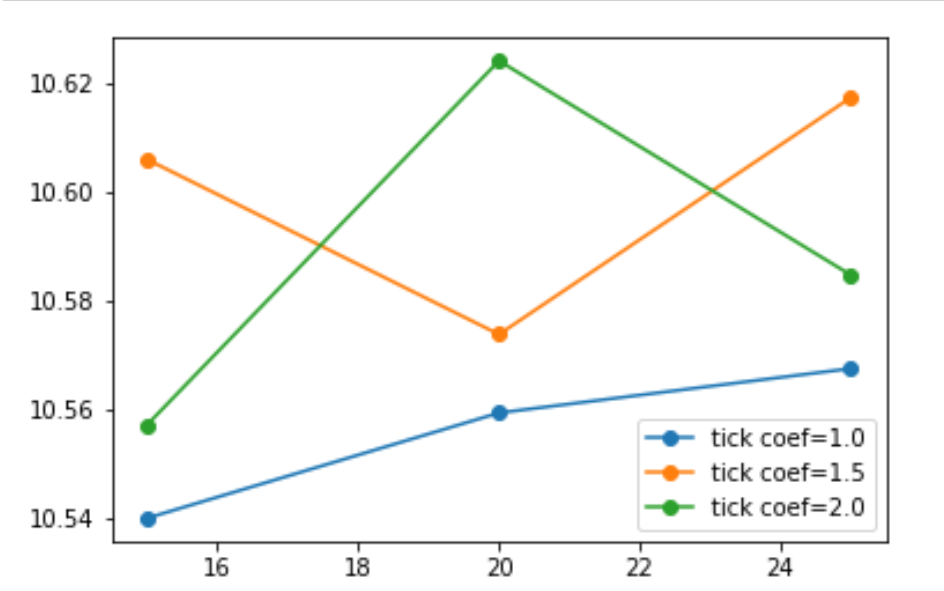
| schedule coef  max behind | 1.0 | 1.5 | 2.0 |
| --- | --- | --- | --- |
| 500 | 12.978204 | 12.800418 | 12.807640 |
| 1000 | 12.843301 | 12.542888 | 12.722310 |
| 2000 | 12.604459 | 12.314120 | 12.421625 |



With the optimal schedule parameters, we can determine the best combination of tick parameters. From the schedule-tick-factor buy-side model, the best combination of parameter is: tick window=15, tick coefficient=1.0

average slippage of schedule-tick-factor buy-side model

| tick coef  tick window | 1.0 | 1.5 | 2.0 |
| --- | --- | --- | --- |
| 15 | 10.540067 | 10.605855 | 10.557200 |
| 20 | 10.559378 | 10.573772 | 10.623949 |
| 25 | 10.567505 | 10.617215 | 10.584666 |



From the schedule-tick-factor sell-side model, the best combination of parameter is: tick window=500, tick coefficient=1.5

average slippage of schedule-tick-factor sell-side model

| tick coef  tick window | 1.0 | 1.5 | 2.0 |
| --- | --- | --- | --- |
| 15 | 12.050607 | 12.559373 | 12.631577 |
| 20 | 12.024125 | 12.605964 | 12.597415 |
| 25 | 12.029545 | 12.010353 | 12.566619 |

