Algorithm Report for a Tender Rule for LT4

My algorithm for the LT4 case assignment is built to take inputs for the case information and either accept a tender or take no action. The tender rule is built in python using the RIT API. The program will either accept a tender offer or do nothing. It doesn't decline offers because the market may change in the 15 to 30 seconds making a bad tender a good tender.

Step 0: Get Tender, Portfolio and Book

Before starting our algorithm, we get the tender, portfolio and book. We also combine the book so that instead of separate markets, we have a list for all bids of a security and all asks for a security. Additionally, we add in the market commissions because we assume we're only using market orders (market orders are worst case scenario, so if it's profitable for the worst case scenario, it must be profitable). We also keep a copy of the book without market orders.

Step 1: Removing Portfolio Position

The first step is to consider our portfolio for the security that the tender offer is for. If we don't have a position for security, we skip this step.

Case 1: The tender would add to the quantity of our position (we have a short position and the tender is to sell shares). If the total would exceed the limits, then we don't do anything. If it wouldn't exceed the limit, then we go through the book and take out the quantity for our current portfolio (in our example, we take out the quantity of bids that match the portfolio (if we are short 10,000 shares, we take out the first 10,000 shares worth of bids from the asks book).

Case 2: The tender opposes our position (we are long and the tender is to sell shares).

Subcase 2a: The portfolio has more quantity than the tender offer. Lets say q_p is the portfolio quantity and q_t is the tender quantity. We take out q_p-q_t worth of orders from the book, because

we'd have to offload that quantity regardless. Next, we compare the Adjusted Market Price (from step 2) for offloading q_t to the tender price, and accept if the tender price is favorable.

Subcase 2b: The tender quantity is higher than the portfolio quantity. We take the portfolio quantity out of the tender quantity, then calculate the AMP for the remaining quantity and compare it to the tender price. If the tender price is favorable considering the AMP, we also know that the price will be favorable to get rid of our portfolio (compared to the market price).

Step 2: Calculating Adjusted Market Price (AMP)

Adjusted market price is our metric for what we can offload a quantity of a security for after including market commissions (the worst case possible). It accounts for market depth, order rate, and volatility.

A big factor is the fact that there is a limit to how fast orders can be put in (either rate limit of automated trading or limit to how fast the user can input trades). We calculate the underlying price by finding the bid ask spread for all anonymous users (we're looking past the fact that all traders are anonymous for round 7). We find the VWAP for the market (with commisions) if we were to offload the tender with market orders. We also find the price that we're 95% certain the underlying price will be above or below (depending on buying or selling) in the amount of time it takes us to offload the shares. If it takes us 2 seconds to place the market orders to offload our position, we are finding out the underlying price bound (with 95% certainty) in 2 seconds.

The AMP is the average between the VWAP we calculated and the worst case for the underlying price, or if the market book isn't deep enough it is just the worst case for the underlying price for when we can offload by (2 seconds in our example).

Step 3: Consider not offloading

Finally, the last thing we consider is if we didn't offload the shares. Since the expected value for the final price compared to the current underlying price is always 0, we use the same calculations for underlying price and consider not offloading the shares. We first assume we'd sell any shares that we could get above the underlying price for. Then, we consider the AMP of remaining shares and compare it to the underlying price minus the fees we'd pay if we didn't sell. The idea here that instead of being forced to sell at an unfavorable price, we could always not sell and pay the fee. If this is still profitable, we'd accept the tenders.

Conclusion:

Finally, if the tender compared to the AMP is profitable or if not selling and paying the fines is profitable, we accept the order.

Next Steps:

The big next step is to take into account liquidity in a market. Taking the average in step two is a short term solution, but ideally the algorithm takes into account how quickly orders disappear, and then how long it would take the anonymous users to add the number of orders we need to offload the shares. Then, instead of looking at how long it would take to offload the position, we'd also take into account how long it would take to have market depth and what the worst case price is with 95% at that point in the future.

I have abided by the Babson Code of Ethics in this work and pledge to be better than that which would compromise my integrity

Nividh Singh

All code (and past versions) can be found here:

https://github.com/NividhSingh/FinancialTradingandRiskManagement/tree/main/LT4