Our strategy consists of 3 main parts:

1. **ETF arbitrage** (exploiting the difference between the market price of AKAV and the sum (NAV) of the market prices of APT, DLR, MKJ
2. **Inverse ETF arbitrage** (exploiting the difference between the market price of AKAV and AKIM and their inverse relationship)
3. **Market Making** using fair price calculations for APT, DLR, MKJ

**ETF arbitrage**

In a perfect market, the market price of AKAV would be the sum of market prices of APT, DLR, MKJ. When these two values are different due to inefficiencies in the market, we have an arbitrage opportunity.

Case 1: AKAV is *overpriced* (i.e. AKAV > APT + DLR + MKJ)

* We buy APT, DLR, MKJ, swap to AKAV, and sell AKAV, making a profit on the difference
* Implementation
  + We have a variable called **min\_etf\_margin**, which is the minimum profit that we expect to make from an ETF arbitrage trade in order for us to start executing the trade
  + We calculate **market\_price\_AKAV** = best bid AKAV. We calculate **market\_nav** = best ask APT + best ask DLR + best ask MKJ
  + **swap\_fee\_to\_AKAV** is provided by the competition
  + We check: **market\_price\_AKAV - market\_nav - swap\_fee\_to\_AKAV > min\_etf\_margin**
  + If so, we try to execute the arbitrage, placing orders at the current best bid/ask
  + Otherwise, we don’t do anything

Case 2: AKAV is *underpriced* (i.e. AKAV < APT + DLR + MKJ)

* We buy AKAV, swap to APT, DLR, MKJ, and sell APT, DLR, MKJ, making a profit on the difference
* Implementation
  + We have a variable called **min\_etf\_margin**, which is the minimum profit that we expect to make from an ETF arbitrage trade in order for us to start executing the trade
  + We calculate **market\_price\_AKAV** = best ask AKAV. We calculate **market\_nav** = best bid APT + best bid DLR + best bid MKJ
  + **swap\_fee\_from\_AKAV** is provided by the competition
  + We check: **market\_nav - market\_price\_AKAV - swap\_fee\_from\_AKAV > min\_etf\_margin**
  + If so, we try to execute the arbitrage, placing orders at the current best bid/ask
  + Otherwise, we don’t do anything

*ETF arbitrage potential improvements*

* We may want to implement “pennying” instead of trying to place orders at the best bid/ask prices in order to make sure our orders actually get filled
* We currently do not have any backup plan for when we are only able to complete one side of the arbitrage (e.g. we sold AKAV but then are unable to buy back APT, DLR, or MKJ due to the market moving, risk limits, or other constraints)

**Inverse ETF arbitrage**

In a perfect market, AKIM should always move inverse to AKAV. E.g., if AKAV rises 5% from its closing price yesterday, AKIM should drop by 5% from its (AKIM’s) opening price today. An example in prices: AKAV closed at $100 on Day 1 and opened again at $100 on Day 2. AKIM rebalances at the end of Day 1 and opens at $100 on Day 2. In the middle of the day, AKAV is trading at $105. AKIM should be trading at $95 to maintain the inverse relationship. However, say AKIM is actually trading at $97. We have an arbitrage opportunity on the difference between the market price of AKIM ($97) and its “fair price” ($95).

*Case 1*: AKIM is overpriced (e.g. AKIM trading at $97 but fair price is $95)

* We sell shares of AKIM in anticipation that it will revert to its true price ($95), expecting to make a $2 profit.
* We sell shares of AKAV to hedge. For example, if later AKAV drops to $103 and AKIM stays at $97, we still make money even though AKIM did not go down. In this case, we would buy back AKAV (net +2) and buy back AKIM (net 0).
* By hedging with AKAV, we ensure that we make money as long as AKAV and AKIM revert to their inverse relationship, no matter if AKIM moves up or down.
* Implementation
  + We have a variable called **inverse\_etf\_margin**, which is the minimum profit that we must expect to make from an inverse ETF arbitrage trade in order for us to start executing the trade
  + At the end of each day we receive an unstructured news event that says something like “EOD - AKIM has rebalanced”; when we get this message, we set the **prev\_day\_close\_AKAV** and **curr\_day\_open\_AKIM** variables to the midprice of AKAV and AKIM at this time respectively
  + Every time the **trade** method is run, we get the current midprice of AKAV (**current\_akav**) and calculate the fair price of AKIM as follows: **fair\_price\_AKIM = curr\_day\_open\_AKIM \* (1 - (current\_akav - prev\_day\_close\_AKAV) / prev\_day\_close\_AKAV)**
  + Basically, the above formula calculates what AKIM should be if it changed the same percentage as AKAV at this moment but in the opposite direction
  + We get the best AKIM bid (**akim\_bid**) and if **akim\_bid - fair\_price\_AKIM > inverse\_etf\_margin**, then we have an arbitrage opportunity and we place an order to sell AKIM and AKAV
  + We store this order with our own data structure
  + We continuously check if AKIM and AKAV have reverted to their correct relationship. If they do, we buy back AKIM and AKAV to complete the arbitrage.

*Case 2:* AKIM is underpriced (e.g. AKIM trading at $93 but fair price is $95)

* We buy shares of AKIM and hedge by buying shares of AKAV
* Everything is basically the same as Case 1 (instead of selling we buy AKIM and AKAV)

*Inverse ETF Potential Improvements*

* Have to check if the monitoring function that checks if AKIM and AKAV have reverted to their correct relationship is working correctly

**Market Making**

With individual stocks APT, DLR, MKJ, we employ a market making strategy, continuously posting competitive bids and asks around a calculated fair price.

For market making, we need to have a “spread” which we will represent as [bid, ask]. In our strategy, our spread looks as follows:

[fair\_price + fade\_adj - edge, fair\_price + fade\_adj + edge]

*Fair Prices*

APT: based on news: **earnings \* P/E ratio**

DLR: based on probability that DLR will receive 100,000 signatures

* We know lognormal distribution, so just do run simulation 10,000 times and take success/10,000 as the probability that DLR will receive 100,000 signatures
* **fair\_price\_DLR = probability \* 100**

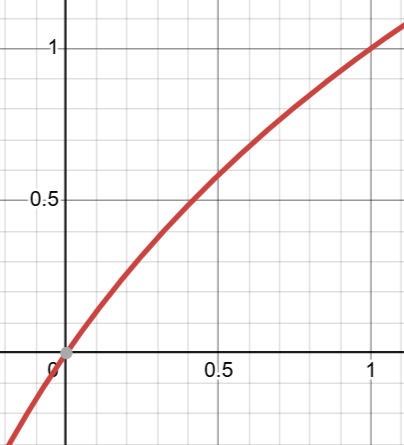
MKJ: not implemented yet

*Fade Adjustment*

Though we have our own calculated fair price, what we see on the exchange from other bots may change our confidence in our fair price. For example, say we think the fair price of MKJ is 100, but someone just bought 200 shares of MKJ from us at 105, i.e. we are short 200 shares of MKJ that we sold at 105. This should make us think that the fair price of MKJ is actually closer to 105, and we should adjust accordingly. This is where fade adjustment comes in. We add **fade\_adj** to our **fair\_price** to account for our current position and how other bots are trading on the exchange. We will keep a variable **fade** that will be the maximum amount **fade\_adj** can take. The closer our position is in absolute value to MAX\_POSITION risk limit, the higher we make our fade adjustment. The formula is below:

**fade\_adj = -fade \* sign(position) \* log2( |position| / MAX\_POSITION )**

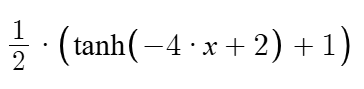
Below is a graph of log2(x+1)

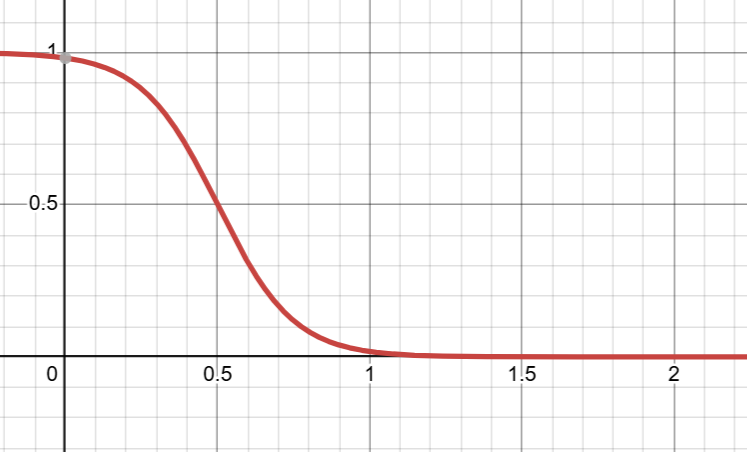


*Edge*

Our interval [fair\_price + fade\_adj - edge, fair\_price + fade\_adj + edge] is centered at **fair\_price + fade\_adj** and **2 \* edge** wide. Thus, **edge** determines how wide our spread is. We should have a certain **min\_margin** that our edge should never go below. Also, our edge should adjust dynamically based on exchange conditions. For instance, if there is a lot of trading activity, we should tighten our edge to be more competitive. If there is little activity, we can widen our edge. We will have a variable **slack** that will be the maximum amount we can increase our edge to (so our edge will be somewhere between **min\_margin** and **min\_margin + slack**. To dynamically adjust our edge, we will use **market\_activity**, which is the amount of bids or asks (we specify one) occurring within **min\_margin + slack** of the midprice. We also add an **edge\_sensitivity** variable to adjust how much we want market activity to affect our edge (0 = market activity has no effect, 1 = market activity has max effect). The formula is below:

**edge = min\_margin + slack \* (0.5 \* tanh(-4 \* market\_activity \* edge\_sensitivity + 2) + 1)**

Below is the graph of . We use the tanh function so that our edge will decrease aggressively when there is high competition in the market.



*Spreads*

Finally, we incorporate multiple different spreads to capture the orders of “dumber” bots. We have a list of spread increments (**spreads**). For example, if **spreads = [5, 10, 15]**, in addition to our competitive bids/asks described above, we also place orders at

[fair\_price + fade\_adj - edge - 5, fair\_price + fade\_adj + edge + 5]

[fair\_price + fade\_adj - edge - 10, fair\_price + fade\_adj + edge + 10]

[fair\_price + fade\_adj - edge - 15, fair\_price + fade\_adj + edge + 15]

If we are lucky and catch some market orders of weaker bots, we make a lot more from these spreads.