**Bid Price Threshold**

For a security with **Direct Market Access**, the following **Bid Price Amount Inequality** must be held for a long or short position of bid price with amount to be respectively opened or closed instantly in full amount for sure in theory:

Where is the ith currently unmatched bid price greater than or equal to with amount and is the jth currently unmatched asked price less than or equal to with amount .

Without changing the value of , as decreases, the number of currently unmatched bid prices greater than or equal to increases while the number of currently unmatched ask prices less than or equal to decreases, causing the inequality to be harder to hold.

So for a given , the minimum value of holding the inequality will be the theoretical **Bid Price Threshold(BPT)** for both instantly opening a long position and closing a short position in full amount.

Note that for **BPT** to exist, the following inequality must be held:

So the maximum value of in this inequality is the total amount of ask, which is also the **Maximum Bid Amount (MBA)**.

**Ask Price Threshold**

For a security with **Direct Market Access**, the following **Ask Price Amount Inequality** must be held for a short or long position of ask price with amount to be respectively opened or closed instantly in full amount for sure in theory:

Where is the jth currently unmatched asked price less than or equal to with amount and is the ith currently unmatched bid price greater than or equal to with amount .

Without changing the value of , as increases, the number of currently unmatched ask prices less than or equal to increases while the number of currently unmatched bid prices greater than or equal to decreases, causing the inequality to be harder to hold.

So for a given , the minimum value of holding the inequality will be the theoretical **Ask Price Threshold(APT)** for both opening a short position and closing a long position in full amount.

Note that for **APT** to exist, the following inequality must be held:

So the maximum value of in this inequality is the total amount of bid, which is also the **Maximum Ask Amount (MAA)**.

**Demonstrations**

Let’s say the current price of a security is 95 with 1 as the base unit, and it applies the following fees:

1. The actual price of a bid paid by the bidder will be increased by 1 after submission but the matched ask will still only get the submitted bid price
2. The actual price of an ask received by the asker will be decreased by 1 after submission but the matched bid will still pay the submitted ask price

With the following unmatched bid-ask table, the **BPTs** and **APTs** with different amounts can be calculated:

|  |  |  |
| --- | --- | --- |
| **Submitted Price** | **Actual Price** | **Amount** |
| 93 Ask | 92 Ask | 870 |
| 94 Ask | 93 Ask | 900 |
| 95 Ask | 94 Ask | 860 |
| 95 Bid | 96 Bid | 850 |
| 96 Bid | 97 Bid | 910 |
| 97 Bid | 98 Bid | 890 |
| 98 Bid | 99 Bid | 880 |

First, the total amount of ask is 870+900+860=2630 while that of bid is 850+910+890+880=3530, meaning that the 3530-2630=900 amount of bid having the lowest price and then the latest submission time won’t be instantly matched in theory(at least not in full amount), while the remaining 2630 amount of bid having the highest price and then the earliest submission time will theoretically be instantly matched in full.

Then, all bids with submitted price 97 and 98 should be instantly matched in full, while the earliest submitted 2630-880-890=860 amount of bids with submitted price 96 should also be instantly matched, leaving the remaining 910-860=50 latest submitted ones, along with all bids with submitted price 95 to be matched later if possible.

Assuming that a new bid can be submitted right after these submissions but right before the actual matching, then the **BPT Table** for the following amounts are:

860 or smaller amounts – 97 submitted bid price

861 to 860+890=1750 – 98 submitted bid price

1751 to 860+890+880=2630 – 99 submitted bid price

Note that it’s impossible to fully match amounts above the **MBA** (2630) instantly no matter how high the bid price is.

Assuming that a new bid can be submitted right after these submissions but right before the actual matching, then the **APT Table** for the following amounts are:

900 or smaller amounts – 97 submitted ask price

901 to 900+860=1760 – 94 submitted ask price

1761 to 900+860+900=2660 – 93 submitted ask price

2661 to 900+860+900+870=3530 – 92 submitted ask price

Note that it’s impossible to fully match amounts above the **MAA** (3530) instantly no matter how low the ask price is.

**MBA/MAA K-chart/MA**

Just like the current price, a K-chart of **MBA**/**MAA** with a given time frame can be drawn, because as long as there’s at least 1 bid/ask, there will be opening, closing, highest and lowest values.

In case where there’s no bid/ask in that time frame, no K-chart will be drawn.

Since K-charts can be drawn for **MBA**/**MAA**, so do their respective Moving Averages, with the catch that invalid **MBA**/**MAA**s should be skipped when trying to collect the lastest set amount of closing **MBA**/**MAA**s.

In case where there are lots of consecutively invalid **MBA**/**MAA**s, **further researches** need to be done on how to exactly calculate their respective MA, since skipping them all will lead to much older data being taken into account, while not skipping them will cause a MA supposedly counting many **MBA**/**MAA**s to actually count a lot less.

It’s probable that some carefully designed adaptive algorithms will be needed to maximize the accuracies of such MA.

This can be useful for an investor without a predetermined amount of opening/closing a long/short position to decide a suitable amount and time range to open/close such positions.

**BPT/APT K-chart/MA**

Just like the current price, a K-chart of **BPT**/**APT** with a given amount and time frame can be drawn, because as long as there’s at least 1 valid **BPT**/**APT** value, there will be opening, closing, highest and lowest values.

In case where there’s no valid **BPT**/**APT** in that time frame, no K-chart will be drawn.

Since K-charts can be drawn for **BPT**/**APT**, so do their respective Moving Averages, with the catch that invalid **BPT**/**APT**s should be skipped when trying to collect the lastest set amount of closing **BPT**/**APT**s.

In case where there are lots of consecutively invalid **BPT**/**APT**s, **further researches** need to be done on how to exactly calculate their respective MA, since skipping them all will lead to much older data being taken into account, while not skipping them will cause a MA supposedly counting many **BPT**/**APT**s to actually count a lot less.

It’s probable that some carefully designed adaptive algorithms will be needed to maximize the accuracies of such MA.

This can be useful for an investor with a predetermined amount for opening/closing a long/short position to decide the precise timing to open/close it with the bid/ask price fine-tuned according to the current **BPT**/**APT** trends and values.

**Real-Time BPT/APT Tables**

With always instantly updated **BPT/APT Tables**, it’d be constantly clear on the **MBA**/**MAA** as well as the **BPT**/**APT** for any given amount, helping investors to decide whether it’d be suitable to open/close a long/short position of what bid/ask price with how many amount.

While larger amounts often means higher **BPT** and lower **APT**, such **BPT/APT Tables** can still be useful for an investor to weight against different amount-**BPT**/**APT** combos and the pros/cons of different timings more effectively and efficiently.

**BAPT Ordering Theorem**

In **Direct Market Access**, **BPT** is always greater than **APT** for any same or different amount of bid/ask at the same time.

Proof:

Recall that **BPT** is the minimum value of holding the **Bid Price Amount Inequality**:

While APT is the maximum value of holding the **Ask Price Amount Inequality**:

If , adding these 2 inequalities will result to this new inequality:

Contradicting with the fact that both and needs to be positive for their respective bid/ask to be matched.

On the other hand, would mean:

These imply that:

Adding these 2 inequalities will result to this new inequality:

Contradicting with the fact that both and needs to be positive for their respective bid/ask to be matched.

**Relationships Between BPT And APT**

Showing the **BPT**, **APT** and the current price together can reveal their relationships and how the market judges the trend.

Let , and be the **BPT**, **APT** and the current price respectively. Because of **BAPT Ordering Theorem**, there are only following 5 possible cases regarding the ordering among , and , and each case reveal different market emotions:

It probably means almost every bidder and asker think that the price’s going to rise or keep rising significantly for quite some time, so many bidders are willing to match lots of those ask prices being higher than the current price.

Askers having large amounts of long positions need time to fully close them, and if they think the price’s going to drop drastically in the long term or if they need to quickly release the involved financial liquidity (or if they’re involved in pump and dump), they may be willing to earn noticeably less to ensure their long positions can be fully closed shortly.

Unless this upward trend stems from exceptional and long-lasting fundamental improvements, many of the bidders might be irrationally optimistic about the increasing price, so bidding under this possibly baseless surge can become deeply stuck in when this zeal dies off, especially when it’s caused by nothing but good news coming out of thin air.

It probably means the majority of the bidders think that the price’s going to rise or keep rising significantly for quite some time while those of the askers doubt that this strong upward trend isn’t going to last much longer, so the more conservative ones are likely to close long positions while the more aggressive ones are likely to open short positions.

As the bidders notice just how close the **APT** is to the current price, many of them will also gradually lower their bidding prices to further maximize their potential profits later on, so the **BPT** should also slowly approach to the current price. Unless one can be sure that the securities are still undervalued, one should wait and look for signs of reversal first.

It probably means the bidders and askers have obviously opposite predictions on the upcoming price trend, maybe because the current price’s still under corrections, or because some important signs of reversal have already surfaced.

It’s basically similar to the case, except that the price trend directions and judgments are reversed.

It’s basically similar to the case, except that the price trend directions and judgments are reversed.

**Possible Implementations**

Assuming that the unmatched bid and ask data are stored this way(using JavaScript to show the idea):

const UNMATCHED\_BIDS = {

id1: { userId: userId1, price: price1, amount: amount1 },

id2: { userId: userId2, price: price2, amount: amount2 },

id3: { userId: userId3, price: price3, amount: amount3 },

idI: { userId: userIdI, price: priceI, amount: amountI },

idN: { userId: userIdN, price: priceN, amount: amountN }

}; // UNMATCHED\_ASKS have the same structure

Then the corresponding **BPT/APT Tables** can be extracted this way:

const baptTable = (unmatchedBidAsks, priceSortFunc) => {

const table = Object.values(unmatchedBidAsks).reduce((t, { price, amount }) => {

t[price] = (table[price] || 0) + amount; // Maps the prices with their respective total amounts

return t;

}, {}), prices = Object.keys(table).sort(priceSortFunc);

// Maps the prices with their respective cumulative amounts from high to low bid/low to high ask price

return Object.entries(prices.reduce((t, price, i) => {

t[price] += table[prices[i - 1]] || 0;

return t;

}, {})).reverse();

//

};

const bptTable = baptTable.bind(null, UNMATCHED\_BIDS, (a, b) => b - a); // Descending sort

const aptTable = baptTable.bind(null, UNMATCHED\_ASKS, (a, b) => a - b); // Ascending sort

And **MBA**/**MAA** and **BPT/APT** can be found by this:

const mbaa = baptTableFunc => baptTableFunc()[0][1]; // The total amount of bid/ask

const mba = mbaa.bind(null, aptTable), maa = mbaa.bind(null, bptTable);

const bapt = (sameSideTableFunc, oppositeSideTableFunc, tickSize, amount) => {

const table = sameSideTableFunc(), maxAmount = oppositeSideTableFunc()[0][1];

if (amount > maxAmount) throw new RangeError(`The largest matchable amount is ${maxAmount}`);

const tableSize = table.length;

for (let i = 0; i < tableSize; i++) {

// The lowest bid/highest ask price with the specified amount that can be instantly matched in full

if (maxAmount – table[i][1] >= amount) return table[i][0];

//

}

// Highest bid/lowest ask price +- tick size = lowest/highest price to be the new highest bid/lowest ask price

return table[tableSize - 1][0] + tickSize;

//

};

const bpt = bapt.bind(null, bptTable, aptTable, TICK\_SIZE);

const apt = bapt.bind(null, aptTable, bptTable, -TICK\_SIZE);