Polyhedral Order Books

Jacob Jackson
jacob@jacobjackson.com

May 25, 2023

1 Introduction

Over the past few decades, most financial markets in the world have adopted electronic trading, using computers to match buyers with sellers. The majority of this trading uses the “central limit order book”, a technology which maintains a list of buy and sell orders for a pair of assets. I propose an alternative to the central limit order book called the “polyhedral order book” which allows each order to include any number of assets instead of being limited to two assets at a time. The polyhedral order book has the potential to improve liquidity across markets and asset classes. Spread markets, which are offered by some exchanges, are a special case of the polyhedral order book.

2 Motivation

Consider a fictional company, Acme Corp, which has a small market capitalization and trades a few hundred times per day. To facilitate these trades, the exchange maintains a central limit order book containing quotes to buy and sell Acme Corp stock. Anyone can post quotes, but in practice most quotes are orders from professional market making firms. The “best bid” is the buy order with the highest price, and the “best ask” is the sell order with the lowest price. For example, the order book could have a best bid of \$50 and a best ask of \$50.04. In this case the spread (the difference between the best bid and the best ask) is 4 cents, which means the market makers will make 2 cents per share from an uninformed trade on average. The spread corresponds roughly to the cost paid by investors to transact in Acme Corp stock, and it is beneficial to investors to decrease spreads. However, for market makers to stay profitable, the spread must be large enough to compensate them for the risk they bear by posting quotes. For example, if a large buy order arrives which moves the fair price by 10 cents, a market maker who sold at \$50.04 would take a loss of 8 cents per share. As risk increases, market makers must widen spreads to compensate.

Unfortunately, large orders in Acme Corp are not the only risk borne by market makers in Acme Corp: they are also exposed to the risk that the broader market moves against them. If Acme Corp has a correlation of 0.5 with the S&P 500, then any fluctuation in the S&P 500 could allow a high-frequency trader to make a profitable trade at the market maker's expense. This increases the cost to trade, because market makers must widen spreads to account for this risk, and increases barriers to entry by making it hard for market makers to be profitable without being high-frequency traders themselves.

This risk can be mitigated by quoting Acme Corp shares in units other than USD. For example, ACME could be quoted against SPY, an exchange-traded fund which tracks the S&P 500. The pair ACME/SPY would have lower volatility than ACME/USD, because fluctuations in the equities market would affect ACME and SPY equally, leaving the ratio of the value of ACME to the value of SPY unchanged. This would allow market makers to offer tighter spreads in ACME/SPY than ACME/USD, which would benefit investors.

But in practice, all stocks are quoted in USD. Why? Because any exchange implementing an ACME/SPY market using a central limit order book would encounter a problem: the pair ACME/SPY may have lower volatility, but narrow spreads require both low volatility and high volume. As a new market, ACME/SPY would have no volume because all orders would be routed to the existing ACME/USD market. This would force market makers to quote wide spreads in ACME/SPY, eliminating any reason for investors to switch from ACME/USD to ACME/SPY and preventing volume in ACME/SPY from improving. This is sometimes called the "cold start problem".

Polyhedral order books solve this cold start problem. In a polyhedral order book, there are no trading pairs like ACME/USD or ACME/SPY. Instead, there is a collection of assets. In this case there are three assets: ACME, SPY, and USD. With a polyhedral order book, a market maker quoting ACME against SPY would immediately begin receiving fills because the polyhedral order book would combine their quotes with other market makers' quotes for SPY against USD to create implied quotes for ACME against USD. Orders are not limited to two assets: it is valid to enter an offer to sell 100 shares of SPY and 500 shares of ACME together for a particular amount of USD. If there are other companies in the same industry, say, BACA and DADA, which have high correlation with ACME, a market maker could quote ACME against a basket of BACA and DADA, which may be less risky than quoting ACME against SPY.

One reason why exchange-traded funds (ETFs) are popular is that they allow investors to obtain exposure to a sector at a low cost because the volatility of the ETF is lower than the volatility of any individual component of the ETF. This leads to lower costs because low-volatility assets are cheaper to trade. With the polyhedral order book, market makers can quote ACME+BACA+DADA against USD directly, offering the same low trading costs without the additional complexity and management fees of an ETF. Crucially, a market maker can quote ACME+BACA+DADA even if there are no investors sending orders for this combination of assets yet, because the polyhedral order book can still match it against other incoming orders. For example, an incoming order to buy ACME could be matched against a resting offer for ACME+BACA+DADA and resting bids for BACA and DADA. And if there is an ETF, say, EVL, with ACME, BACA, and DADA as components, then market makers can offer tight spreads when quoting ACME+BACA+DADA against EVL because there is no price risk.

A mechanism providing some of the functionality of the polyhedral order book already exists in the form of spread markets on futures exchanges. For example, market makers in a "calendar spread" market might quote prices for the difference between March and April contracts in a commodity such as crude oil. A trader placing an order in the April market could either match directly with a resting order in the April market, or against an "implied order" formed by the combination of a resting order in the March market with a resting order in the March-April calendar spread. Futures exchanges such as the Chicago Mercantile Exchange already have systems in place which match and disseminate these implied orders. A polyhedral order book would implicitly provide all possible spread markets: market makers could quote whichever spreads they want with no need for the exchange to explicitly designate the set of allowed spreads. Other spread markets, such as inter-commodity spreads (for example, a market for the difference in price between crude oil and gasoline), would be handled seamlessly by the same system, again with no need for the exchange to designate which spread markets should exist.

Having outlined the benefits of the polyhedral order book, I will describe how it works.

3 Definition

In a traditional central limit order book, there are two kinds of orders: bids and asks. The book is said to be "crossed" when there is a pair of orders, a bid and an ask, where the bid price is greater than or equal to the ask price. Equivalently, the book is crossed when there are orders on the book that could be matched to make a trade. During normal operation, the book is never crossed. Instead, when an order arrives which would cross the book, it is repeatedly matched with resting orders until the book is no longer crossed.

Determining whether the book is crossed — that is, given a set of orders, determining whether it is possible to match some of them to make a trade — is the purpose of an order book. Given a solution to this problem, the order book implementation follows. I will define the polyhedral order book in terms of its solution to this problem.

3.1 The matching problem

Let N be the number of assets and M be the number of orders. For simplicity, assume assets cannot have negative value. The state of the polyhedral order book is a matrix $A \in \mathbb{R}^{N\times M}$ whose columns represent orders, such that A_{ij} is the change in exposure to asset i that would result from taking the opposite side of the j-th order. For example, if asset 1 is USD, asset 2 is SPY, and asset 3 is ACME, then an order to buy 500 shares of ACME for 25,000 USD would be represented by the vector

$$\begin{bmatrix} 25{,}000 \\ 0 \\ -500 \end{bmatrix}$$

indicating that if the order were filled, the counterparty would have its balance of USD increased by 25,000, its balance of SPY unchanged, and its balance of ACME decreased by 500.

A *match vector* is defined as a vector $x \in \mathbb{R}^M$ such that $Ax \geq 0$ and $0 \leq x \leq 1$. The book is crossed when there exists a nonzero match vector.

The match vector gives the fraction of each order that is filled. The vector Ax, which I call the *surplus*, gives the amount of each asset remaining after all orders are matched. If the surplus is nonzero, there are excess assets which must be distributed according to some rule. A reasonable rule is to assign all the surplus to the entity that placed the order which triggered the match. This mirrors the behavior of a traditional central limit order book.

The problem of determining whether there exists a nonzero match vector, that is, whether the book is crossed, can be solved by the following linear program:

$$\begin{aligned} \max \quad & \sum_i x_i \\ \text{subject to} \quad & Ax \geq 0 \\ & x \geq 0 \end{aligned}$$

Any nonzero match vector x will satisfy $\sum_i x_i > 0$, so the objective will be unbounded if a nonzero match vector exists. (To recover the match vector, add the additional constraint $x \leq 1$.) I call this linear program the *matching problem*.

3.2 The pricing problem

A theorem in linear programming states that every linear program has an equivalent form, called its "dual". The original program is called the "primal". The primal and the dual are related in the following way: the primal is infeasible if and only if the dual is unbounded, and vice versa.

The dual of the matching problem is to find a *price vector* $y \in \mathbb{R}^N, y \geq 0$ such that $y^\top A < 0$. I call this the *pricing problem*. If there exists $y \geq 0$ such that $y^\top A < 0$, then the book is not crossed.

The price vector y can be interpreted as a hypothetical market participant offering to buy or sell the i-th asset for y_i units of some currency (called the *pricing currency*). If the book is not crossed, then it is possible to find y such that, despite the participant being willing to buy or sell at the same price y_i for each asset, it is still impossible to make risk-free profit in the pricing currency by transacting with the hypothetical market participant and a resting order.

The pricing problem is important because although solving the matching problem determines whether a nonzero match vector exists, it does not find the match vector which maximizes price improvement for the incoming order. To find such a vector, we use a procedure I call the *pricing-matching algorithm*.

3.3 The pricing-matching algorithm

We assume the book is a matrix $A \in \mathbb{R}^{N \times M}$ which just received an incoming order $z \in \mathbb{R}^N$, and that z is the last column of A. We assume the book was not crossed before z arrived.

We define a *fair match* as a match vector x where there exists a price vector $y \geq 0$, $y \neq 0$, $y^\top z \geq 0$, which we call a *fair price*, satisfying the following conditions for all $i \neq M$:

1. *Representativeness:* $x_i = 0$ or $(y^\top A)_i = 0$.
2. *No trade-through:* $(y^\top A)_i \leq 0$.

Together these conditions ensure that the price y is representative of the price at which the match occurred (condition 1), and that there is no order on the book offering a better price than y (condition 2).

The pricing-matching algorithm finds a nonzero fair match if one exists. First we solve the following problem, which I call the *fair price problem*:

$$\begin{aligned} \text{min} \quad & z^\top y \\ \text{subject to} \quad & y \geq 0 \\ & \|y\|_1 = 1 \\ \text{for each } i \neq M\text{:} \quad & (y^\top A)_i \leq 0 \qquad (*) \end{aligned}$$

The constraint $\|y\|_1 = 1$ is included to avoid the solution $y = 0$. The problem is feasible because we assumed the book was not crossed before z arrived.

Having solved for y, we check whether $z^\top y < 0$. If so, the neighborhood of y contains a solution to the pricing problem for A, so the book is not crossed and we return $x = 0$.

Otherwise, $z^\top y \geq 0$. We will show this implies the pricing problem is infeasible. Any solution y' to the pricing problem satisfies $A^\top y' < 0$, implying $z^\top y' < 0$ because z is a column of A. Assume $\|y'\|_1 = 1$ (otherwise scale it by a $\frac{1}{\|y'\|_1}$ factor). Then y' satisfies all the constraints of the fair price problem, but $z^\top y' < 0 \leq z^\top y$, contradicting the optimality of y.

So the pricing problem is infeasible, and the book is crossed. Now, dropping slack constraints from a linear program does not change its optimal value, so the argument in the previous paragraph would continue to apply if we dropped the starred constraints that are slack. Therefore the book would remain crossed even if we were to remove all orders corresponding to the slack starred constraints. Thus, we can solve the following matching problem and be assured of a nonzero solution:

$$\begin{aligned} \max \quad & x_M \\ \text{subject to} \quad & Ax \geq 0 \\ \text{for each } i\text{:} \quad & 0 \leq x_i \leq 1 \quad \text{if } i = M \text{ or } (y^\top A)_i = 0 \\ & x_i = 0 \qquad\quad\;\; \text{otherwise} \end{aligned}$$

By construction, this is a fair match. We apply the match, removing any orders that are completely filled from A. Then we repeat the pricing-matching algorithm as long as z is not completely filled.

3.4 Pseudocode

The following pseudocode implements a polyhedral order book.

```
A := empty matrix
For each event e:
    If e is an incoming order z:
        Add z as a column to A
        While x := PricingMatching(A, z) is nonzero:
            Report a trade for each nonzero entry in x
            For each i:
                Multiply the i-th column of A by (1 - x_i)
            Remove any all-zero columns from A
    If e is an order cancellation:
        Remove the cancelled order's column from A
```

4 Remarks

1. A central limit order book is a polyhedral order book with $N = 2$.
2. A polyhedral order book is backwards compatible: it can accept orders from systems that are designed around central limit order books. Users can start receiving better fills without any need to be aware of the details of the implementation.
3. Polyhedral order books are computationally more expensive than central limit order books. Using them to handle the transaction volume of modern financial markets poses a significant engineering challenge.

4. An efficient implementation would likely maintain a price vector y, only applying the pricing-matching algorithm when the pricing problem becomes infeasible. Most orders do not change the best bid and offer; with this approach, such orders can be accepted cheaply by checking only that the dot product $y^\top z < 0$, where z is the vector representing the new order.

5. An exchange implementing a polyhedral order book would want to remain compatible with existing systems designed to consume central limit order book data. A polyhedral order book could present a view of itself as if it were a central limit order book for a given pair of assets, providing for each price level the amount of liquidity available to an incoming order seeking to trade one asset for the other.

6. Many exchanges apply different fees to resting orders ("maker fees") and incoming orders ("taker fees"). Some exchanges offer negative maker fees ("maker rebates"). For example, an exchange might have a taker fee of 0.3 cents per share and a maker rebate of 0.1 cents per share. However, this is inadvisable when using a polyhedral order book because a single match can include more than one resting order. If an incoming order for 100 shares were to match against 4 resting orders of 100 shares each, the exchange would take a loss because the total maker rebate of 0.4 cents per share would exceed the taker fee of 0.3 cents per share. To avoid this issue, the simplest solution is to set a maker fee of zero and levy all fees on the taker order. A positive maker fee also works to avoid the exchange taking losses, but may decrease liquidity as it increases the fee on a match proportionally to the number of resting orders it includes.

5 Conclusion

This document introduced polyhedral order books, which have the potential to improve liquidity in financial markets. Future work may focus on implementing polyhedral order books efficiently in practice. Adapting the concepts of price-time priority and tick sizes from central limit order books is also left for future work.