An optimization model for multi-asset batch auctions with uniform clearing prices

Tom Walther

Gnosis / Zuse Institute Berlin
tom.walther@gnosis.pm

Abstract. Conventional asset exchange mechanisms are based on pair-wise order books and continuous order matching. Contrasting this, we introduce an alternative exchange and price-finding mechanism that simultaneously considers multiple assets in a discrete-time setting. We present a MIP formulation for this mechanism together with some computational results.

1 Introduction

In continuous-time asset exchange mechanisms, orders are typically collected in order books of two assets that are traded against each other. A trade happens whenever at least two matching orders are available, i.e., if there exists an exchange rate satisfying their specified limit prices. Most real-world asset exchanges rely on some principal asset (e.g., USD) and only offer trading pairs with this asset. Hence, arbitrary assets can not be traded directly against each other, thus limiting trading opportunities and incurring unnecessary costs for market participants. This potentially affects any market that trades financial assets, as for instance foreign exchange markets.

In our approach, we want to collect orders for a set of multiple assets in a single joint order book and compute exchange prices for all asset pairs simultaneously at the end of discrete time intervals (*multi-asset batch auction*). Trades between the same asset pairs are then all executed at the same exchange rate (*uniform clearing price*). This mechanism enables so-called *ring trades*, where orders are matched along cycles of assets, thus enhancing liquidity for traders. In order to exclude arbitrage opportunities, we require prices to be consistent along such cycles, i.e., we want to impose the constraint

$$p_{i|j} \cdot p_{j|k} = p_{i|k} \tag{1}$$

for the prices of all assets i, j, k.

Our main motivation for considering this type of asset exchange mechanism stems from the emerging blockchain technology, where no intermediary is required in order to freely exchange blockchain-based assets (so-called *tokens*) among each other. Current centralized blockchain asset exchanges do not make full use of this potential and merely try to replicate conventional stock exchanges, thus incurring the same frictions and arbitrage opportunities mentioned above.

A good overview on the theory of arbitrage from a finance perspective is given in [4]. The concept of frequent batch auctions aiming at reducing arbitrage and improving liquidity has been investigated extensively in [2]. Moreover, the advantages of uniform-price clearing have been discussed in [3]. In this document, we want to describe the problem of determining uniform clearing prices for all pairs of assets involved in a multi-asset batch auction process, and present a mixed-integer programming (MIP) solution approach.

2 Problem statement

Data Let $\mathcal{A} := \{\alpha_0 \dots \alpha_{n-1}\}$ denote the set of the n assets that are being considered. Without loss of generality, the asset α_0 shall be referred to as *reference asset*. Moreover, let there be a set $\mathcal{O} = \{\omega_1 \dots \omega_N\}$ of N limit buy orders in the batch to be processed. Every such buy order consists of a tuple $(i, j, \overline{x}, \pi)$ that is to be read as

"Buy (at most) $\overline{x}$ units of asset α_i for asset α_j if the rate $p_{i|j}$ is at most π".

Please note that limit sell orders can be modelled similarly, but will not be considered here in order to keep notation as simple as possible.

Variables The pairwise exchange rates between two assets α_i and α_j are denoted by $p_{i|j}$, meaning the price of one unit of α_i measured in units of α_j. As an example, if $p_{i|j} = 10$, one would need to pay an amount of 10 units of α_j in order to purchase a unit of α_i. In the optimization, we want to determine all pairwise exchange rates between assets as well as the fraction of every order that can be fulfilled at these rates. Obviously, this fraction may only be positive for an order if the respective exchange rate satisfies the given limit price.

Objective Our goal is to find exchange rates for all asset pairs that maximize the trade volume that is enabled, measured in units of the reference asset.

Constraints The solution we are aiming at needs to satisfy several requirements:

a. *buy limit price* – for every buy order $\omega = (i, j, \overline{x}, \pi) \in \mathcal{O}$: The order can only be executed (fully or fractionally) if the exchange rate does not exceed the stated limit price, i.e., $p_{i|j} \leq \pi$.
b. *asset balance* – for every asset $\alpha_i \in \mathcal{A}$: The amount of assets α_i bought must equal the amount of assets α_i sold across all orders.
c. *price coherence* – for all asset pairs (α_i, α_j): $p_{i|j} \cdot p_{j|i} = 1$.
d. *arbitrage-freeness* – for all asset triples $(\alpha_i, \alpha_j, \alpha_l)$: $p_{i|j} \cdot p_{j|l} = p_{i|l}$.

3 Mixed-integer programming model

3.1 Model data

With the exchange rates between pairs of assets being variables, modelling arbitrage-freeness intuitively leads to a nonlinear formulation due to the multiplications that are required. This would result in tight limitations to the tractable problem size as well as numerical instability. However, it is possible to avoid nonlinearity by not considering all pairwise asset rates explicitely but representing all asset prices only w.r.t. the reference asset α_0. Let $p_i := p_{i|0}$ denote the price of asset α_i expressed in units of α_0 (hence, $p_0 = 1$). Applying the arbitrage-freeness and price coherence conditions directly, we can write the exchange rate between two assets α_i and α_j as

$$p_{i|j} = p_{i|0} \cdot p_{0|j} = \frac{p_{i|0}}{p_{j|0}} = \frac{p_i}{p_j}. \tag{2}$$

In order to be able to state a MIP formulation for our problem, we will express all data and variables in terms of matrices and vectors. First, we store all asset price variables in a vector $(p_i) =: \mathbf{p} \in \mathbb{R}^n_{\geq 0}$. It makes sense to set an explicit lower and upper bound for the price of every asset α_i (e.g., in order to avoid excessive fluctuations), so let us require $p_i \in [\underline{p}_i, \overline{p}_i]$. These bounds could for example be derived as half/double the previous asset prices, or simply be set to some low/high values that appear reasonable. In order to ensure $p_0 = 1$ for the reference asset α_0, we set $\underline{p}_0 = \overline{p}_0 = 1$. The price bounds shall be stored in vectors $\underline{\mathbf{p}}$ and $\overline{\mathbf{p}}$, respectively.
Moreover, we introduce two data matrices

$$\mathbf{T}^b \in \{0,1\}^{N \times n} \quad \text{with} \quad \mathbf{T}^b \ni t^b_{k,i} = 1 \Leftrightarrow \text{asset } \alpha_i \text{ to be bought in order } \omega_k,$$
$$\mathbf{T}^s \in \{0,1\}^{N \times n} \quad \text{with} \quad \mathbf{T}^s \ni t^s_{k,i} = 1 \Leftrightarrow \text{asset } \alpha_i \text{ to be sold in order } \omega_k.$$

Since we are only considering orders of one asset against one other, there must be exactly one entry equal to 1 per row (order) in both $\mathbf{T}^b$ and $\mathbf{T}^s$. The maximum number of units of the asset α_i to be bought in an order $\omega_k = (i, j, \overline{x}, \pi)_k$ shall be denoted by $\overline{x}_k$ and stored, for all orders, in a vector $\overline{\mathbf{x}} \in \mathbb{R}^N_{\geq 0}$. The precise buy amount of α_i, which is to be determined by the optimization procedure, will be denoted by $(x_k) =: \mathbf{x} \in \mathbb{R}^N_{\geq 0}$. Hence, $x_k \leq \overline{x}_k$ for all orders $\omega_k \in \mathcal{O}$. However, we will only determine the values in $\mathbf{x}$ indirectly. Instead, for the purpose of keeping the model linear, we define another variable vector $(v_k) =: \mathbf{v} \in \mathbb{R}^N_{\geq 0}$, where v_k represents the traded volume of assets in the order ω_k denoted in units of the reference asset α_0. The limit prices of all orders are stored as vector $(\pi_k) =: \boldsymbol{\pi} \in \mathbb{R}^N_{\geq 0}$, where $\pi_k \in \boldsymbol{\pi}$ refers to the exchange rate between the respective assets of order ω_k according to the definition above.
Besides the main problem variables $\mathbf{p} \in \mathbb{R}^n_{\geq 0}$ and $\mathbf{v} \in \mathbb{R}^N_{\geq 0}$, our MIP model requires several additional variables. Most importantly, let $\mathbf{z} \in \{0,1\}^N$ be a vector of binary variables, where $z_k \in \mathbf{z}$ indicates whether an order ω_k may be (fully or partially) executed, or not. Precisely, we require $z_k = 1$ if and only if the exchange rate between the assets of ω_k

satisfies the respective limit price π_k, otherwise $z_k = 0$. Furthermore, the feasible region for the execution volume v_k of an order ω_k depends on the value of z_k. In particular, v_k must be set to zero if $z_k = 0$ and can only be non-zero otherwise. In order to model this disjoint behaviour, we make use of a *disjunctive programming* formulation [1] for which we introduce the auxiliary price variable vectors $\mathbf{p}^{b,0}, \mathbf{p}^{b,1}, \mathbf{p}^{s,0}, \mathbf{p}^{s,1} \in \mathbb{R}^N_{\geq 0}$. The idea behind this approach will be expanded upon after the model formulation.

Our model allows for setting a minimum fraction of execution for every order, when its limit price is satisfied, as a parameter. In this paper, we will use a global value $\underline{r} \in [0, 1]$ for all orders.

3.2 Model formulation

With all these definitions, we can finally state the MIP model for the problem of multi-asset batch auctions with uniform clearing prices as follows:

$$\max_{\mathbf{v},\mathbf{p}} \quad \sum_{k=1}^{N} v_k \tag{3a}$$

$$\text{s.t.} \quad \sum_{k=1}^{N} t^b_{k,i}\, v_k = \sum_{k=1}^{N} t^s_{k,i}\, v_k \qquad \forall\, i = 0 \dots n{-}1 \tag{3b}$$

$$\sum_{i=1}^{n-1} t^b_{k,i}\, \underline{p}_i (1 - z_k) \leq p^{b,0}_k \leq \sum_{i=1}^{n-1} t^b_{k,i}\, \overline{p}_i (1 - z_k) \quad \forall\, k = 1 \dots N \tag{3c}$$

$$\sum_{i=1}^{n-1} t^s_{k,i}\, \underline{p}_i (1 - z_k) \leq p^{s,0}_k \leq \sum_{i=1}^{n-1} t^s_{k,i}\, \overline{p}_i (1 - z_k) \quad \forall\, k = 1 \dots N \tag{3d}$$

$$\sum_{i=1}^{n-1} t^b_{k,i}\, \underline{p}_i z_k \leq p^{b,1}_k \leq \sum_{i=1}^{n-1} t^b_{k,i}\, \overline{p}_i z_k \qquad \forall\, k = 1 \dots N \tag{3e}$$

$$\sum_{i=1}^{n-1} t^s_{k,i}\, \underline{p}_i z_k \leq p^{s,1}_k \leq \sum_{i=1}^{n-1} t^s_{k,i}\, \overline{p}_i z_k \qquad \forall\, k = 1 \dots N \tag{3f}$$

$$\sum_{i=1}^{n-1} t^b_{k,i}\, p_i = p^{b,0}_k + p^{b,1}_k \qquad \forall\, k = 1 \dots N \tag{3g}$$

$$\sum_{i=1}^{n-1} t^s_{k,i}\, p_i = p^{s,0}_k + p^{s,1}_k \qquad \forall\, k = 1 \dots N \tag{3h}$$

$$p^{b,0}_k \geq \pi_k\, p^{s,0}_k \qquad \forall\, k = 1 \dots N \tag{3i}$$

$$p^{b,1}_k \leq \pi_k\, p^{s,1}_k \qquad \forall\, k = 1 \dots N \tag{3j}$$

$$v_k \geq \underline{r}\, \overline{x}_k\, p^{b,1}_k \qquad \forall\, k = 1 \dots N \tag{3k}$$

$$v_k \leq \overline{x}_k\, p^{b,1}_k \qquad \forall\, k = 1 \dots N \tag{3l}$$

$$\mathbf{z} \in \{0,1\}^N$$

$$\mathbf{v},\, \mathbf{p}^{b,0},\, \mathbf{p}^{b,1},\, \mathbf{p}^{s,0},\, \mathbf{p}^{s,1} \in \mathbb{R}^N_{\geq 0}$$

$$\mathbf{p} \in \mathbb{R}^n_{\geq 0}$$

The objective function (3a) maximizes the total trading volume in terms of units of the reference asset α_0 that is processed with all orders. Constraint (3b) secures that the total buy and sell volumes across all orders must be equal for every asset. With uniform clearing prices, volume balance implies asset balance. Please note that the summation in this constraint (as well as in the following ones) is needed to select the correct volume variables for the respective asset under consideration. The auxiliary variables in $\mathbf{p}^{b,0}$ and $\mathbf{p}^{s,0}$ refer to the prices of the buy- and sell-asset of every order ω_k if that order is not to be executed ($z_k = 0$). In that case, their values must lie within the respective bounds provided by $\underline{\mathbf{p}}$ and $\overline{\mathbf{p}}$, and otherwise shall be set to zero. This requirement is ensured by the constraints (3c) and (3d). Similarly as above, the constraints (3e) and (3f) control the auxiliary variables in $\mathbf{p}^{b,1}$ and $\mathbf{p}^{s,1}$ for the case that an order ω_k is executable ($z_k = 1$). The relation between the auxiliary price variables and the actual asset prices is established by the constraints (3g) and (3h). For all orders, the disjunctive behaviour is modelled by the constraints (3i)–(3l). The idea is as follows: If some order ω_k is not to be executed ($z_k = 0$), then the prices $p_k^{b,0}$ and $p_k^{s,0}$ must both lie within the bounds given by (3c) and (3d) as well as fulfill (3i) (i.e., not satisfy the limit price). At the same time, $p_k^{b,1}$ and $p_k^{s,1}$ are set to zero by (3e) and (3f), thus trivially satisfying (3j). This then also implies the volume v_k to be set to zero by (3k) and (3l). Conversely, if ω_k is to be executed ($z_k = 1$), $p_k^{b,0}$ and $p_k^{s,0}$ are set to zero by (3c) and (3d), while $p_k^{b,1}$ and $p_k^{s,1}$ lie within the bounds provided by (3e) and (3f) as well as fulfill (3j) (i.e., satisfy the limit price). This finally requires the execution volume v_k to be within the specified fractions via the constraints (3l) and (3l).

3.3 Computational results & Outlook

In order to investigate the performance of our model, we have conducted computational experiments for different numbers of assets ($n \in \{5, 10, 20, 50\}$) and orders ($N \in \{100, 200, 500\}$). For every combination of n and N, we have generated 20 random instances, whereby the randomness reflects our expectation of somewhat realistic situations. In particular, we expect not all assets to be equally important in terms of trading volume and, hence, to have varying numbers of orders on different asset pairs. We used Gurobi 8.0.0 as MIP solver on an Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz machine with 16Gb RAM and using 4 threads, and a timelimit set to 1800 seconds for every instance.
The results of our computational experiments as in Table 1 show that runtimes of the MIP formulation sharply increase both with the number of assets and the number of orders that are being considered. Several instances with more than 20 assets and more than 200 orders could not be solved to optimality within the timelimit. The largest instances with 50 assets and 500 orders even left very substantial gaps in most cases. If such larger problems are to be considered, we can think of the following heuristics/approximations:

- Aggregate orders with similar limit prices on every asset pair
- Optimize over subsets of assets separately and fix prices in overall problem

# orders		# assets			
		5	10	20	50
100	∅ runtime	0.32	0.32	0.90	5.54
	# timeouts	0	0	0	0
	– ∅ gap	-	-	-	-
200	∅ runtime	1.31	2.49	26.25	571.10
	# timeouts	0	0	0	12
	– ∅ gap	-	-	-	23.46%
500	∅ runtime	15.67	82.47	522.72	-
	# timeouts	0	0	11	20
	– ∅ gap	-	-	21.59%	74.82%

Table 1. Computational results of instances of the MIP model (3). The (geometric) means of the runtimes have been computed only with respect to the instances that could be solved to optimality before the timelimit. Conversely, the average optimality gap does not take solved instances into account.

Finally, a possible direction for future research besides finding more efficient solution methods for the problem at hand would be the consideration of more complex order types. For example, instead of trading one single asset against one other, traders might want to trade a set of multiple assets against multiple others (*basket orders*), without being interested in individual exchange rates. Another possibility could be to allow the order amounts to be flexible with the determined prices, i.e., traders might want to buy/sell more of some asset if its price is favorable.

References

1. Balas, E.: Disjunctive Programming. Annals of Discrete Mathematics 5, 3–51 (1979). doi:10.1016/S0167-5060(08)70342-X
2. Budish, E., Cramton, P., Shim, J.: The High-Frequency Trading Arms Race: Frequent Batch Auctions as a Market Design Response. The Quarterly Journal of Economics 130(4), 1547–1621 (2015). doi:10.2139/ssrn.2388265
3. Engelbrecht-Wiggans, R., Kahn, C.: Multi-unit auctions with uniform prices Economic Theory 12, 227–258 (1998), Springer-Verlag. doi:10.1007/s001990050220
4. Ross, S.A.: Neoclassical Finance, Princeton Lectures in Finance, Chapter 1 (2009), Princeton University Press.