Nomisma *2020 May03*

Given the following orders in Nomisma matching engine find all transactions. Conditional order is a set of orders so that each of them should be filled in proportion (not necessarily all-or-nothing).

```
    order1                    buy call,   strike = 08000, price = 864, size = 40
    order2                   sell call,   strike = 08000, price = 864, size = 10
    order3                    buy call,   strike = 10000, price = 123, size = 30
    order4                   sell call,   strike = 10000, price = 123, size = 80
    order5 (conditional)      buy call,   strike = 08000, price = 864, size = 20
                             sell call,   strike = 10000, price = 123, size = 20
```

For ordinary exchange with continuous order matching engine, order placement time is critical factor when considering order-filling priority. Yet Nomisma uses frequent-batch-auction which does not consider order placement time. Objective is to maximise the number of shares traded. However if there are multiple solutions that maximise the number of shares, how can we judge which allocation (of traded size) to various investors is the best (is fair to everyone)?

case 1 : order 5 is not filled at all
Total trade volume will be 40 shares.

```
    trade1 call, strike = 08000, price = 864, size = 10, buy-side = order1, sell-side = order2
    trade2 call, strike = 10000, price = 123, size = 30, buy-side = order3, sell-side = order4
```

case 2 : order 5 is filled first
Please note as only 50% of buy-leg in order5 is filled, we can only fill exactly 50% of sell-leg in order5 as well (that's what means by in proportion). Again, total trade volume will be 40 shares.

```
    trade1 call, strike = 08000, price = 864, size = 10, buy-side = order5, sell-side = order2
    trade2 call, strike = 10000, price = 123, size = 10, buy-side = order3, sell-side = order5
    trade3 call, strike = 10000, price = 123, size = 20, buy-side = order3, sell-side = order4
```

case 3 : order 5 is filled partially
Can we do better than this? No but there are multiple optimal solutions in between case 1 and 2, instead of filling order 5 by 50%, we can fill it by 30% as shown in the following.

```
    trade1 call, strike = 08000, price = 864, size = 06, buy-side = order5, sell-side = order2
    trade2 call, strike = 10000, price = 123, size = 06, buy-side = order3, sell-side = order5
    trade3 call, strike = 08000, price = 864, size = 04, buy-side = order1, sell-side = order2
    trade4 call, strike = 10000, price = 123, size = 24, buy-side = order3, sell-side = order4
```

As there are multiple solutions, which solution (allocation) is more fair for investors? Before that, lets think about that, is a fair platform (trading enginer) necessary? The answer is yes, because :

- it is regulated, an exchange should treat all investors equal, and
- a fair platform is an incentive for investors to place order, hence earning commission

How to measure the fairness of an allocation? Let's do it with a greedy approach. In frequent-batch-auction engine, order placement time is not considered, besides in this example, all order prices are matched (864 for more ITM call and 123 for less ITM call), thus the only factor left behind is quantity :

- buy-leg of `order5` (with 20 shares, 1/3) competes with `order1` (with 40 shares, 2/3) to fill `order2`
- sell-leg of `order5` (with 20 shares, 1/5) competes with `order4` (with 80 shares, 4/5) to fill `order3`
- two legs of `order5` are bounded by proportion constraint

If we judge fairness according to race for the more ITM option, then *1/3* of *10* shares in `order2` fill buy-leg of `order5`, while the rest *2/3* of *10* shares in `order2` fill `order1`, then we have :

```
    trade1 call, strike = 08000, price = 864, size = 03, buy-side = order5, sell-side = order2
    trade2 call, strike = 10000, price = 123, size = 03, buy-side = order3, sell-side = order5
    trade3 call, strike = 08000, price = 864, size = 07, buy-side = order1, sell-side = order2
    trade4 call, strike = 10000, price = 123, size = 27, buy-side = order3, sell-side = order4
```

However if we judge fairness according to race for the less ITM option, we have different allocation :

```
trade1 call, strike = 08000, price = 864, size = 02, buy-side = order5, sell-side = order2
trade2 call, strike = 10000, price = 123, size = 02, buy-side = order3, sell-side = order5
trade3 call, strike = 08000, price = 864, size = 08, buy-side = order1, sell-side = order2
trade4 call, strike = 10000, price = 123, size = 28, buy-side = order3, sell-side = order4
```

Is there generic way for determining allocation? Construct vector in 6 dimensional space, composing of the order size (or filled size) for each order and legs of order :

```
order_size   = [40,10,30,80,20,20]
case2_ans    = [00,10,30,20,10,10]         sum = 80 = 2 * turnover
case3_ans    = [01,10,30,21,09,09]         sum = 80 = 2 * turnover
case3_ans    = [02,10,30,22,08,08]         ...
case3_ans    = [03,10,30,23,07,07]         ...
case3_ans    = [04,10,30,24,06,06]
case3_ans    = [05,10,30,25,05,05]
case3_ans    = [06,10,30,26,04,04]
case3_ans    = [07,10,30,27,03,03]
case3_ans    = [08,10,30,28,02,02]
case3_ans    = [09,10,30,29,01,01]         sum = 80 = 2 * turnover
case1_ans    = [10,10,30,30,00,00]         sum = 80 = 2 * turnover
```

My solution (not sure whether it is correct) is to find the allocation with maximum cosine similarity when comparing to `order_size`. Cosine similarity is a measurement of the angle spanned between two vectors.

How matching algorithm linked with static replication in Nomisma?
For convention exchange, there are separate order book for various instruments : forward, call with different strikes, put with different strikes, call spread and put spread, however in Nomisma, the matching engine can static replicate options, decomposing them into fewer elements such as forward and call with different strikes only (put can be synthesized by a forward minus a call option with corresponding strike, call spread is decomposed into two call options), thus introduces dependency between order books, exhibiting conditionality, which means, orders that span across more than one books, when conditional order is filled, each leg should be filled in proportion. Hence, mixed integer programming kicks in for maximising the number of traded shares. Why decomposing multiple order books into a smaller set? The answer is offer better liquidity for less liquid exotic products, increasing turnover.

Special features about Nomisma engine :
- static replication and orderbook dependency
- frequent batch auction, avoiding low latency trade

How matching algorithm linked with quantlib in Nomisma?
As Nomisma's business is to provide an exchange for trading digital asset derivatives, why does it involve option pricing and quantitative modelling stuffs? In conventional exchange like HKEX, when we trade option there is a margin account to handle counterparty risk. When we start to make loss, the centralized entity, that is the HKEX will approach us to refill the margin account, yet this is not the case in distributed exchange. How can we manage counterparty risk? This is done by margin loan, and margin loan itself is tradable. Option and margin loan are traded together as two legs of conditional order in Nomisma. Woo, a very new concept.

Nomisma prefers local volatility to stochastic volatility?
Local volatility offers perfect fitting on market data, while stochastic volatility does not. Regarding to TDMS, why do you use stochastic volatility instead of local volatility? Some articles claimed that local volatility is not good enough to model the dynamics of volatility and hence not good for path dependent exotic products. How to calculate local volatility? Here is a simple algo : given a matrix of Black Scholes volatility, interpolate a smooth surface of call option price in strike-tenor space, find the first order derivative of call option price with respect to tenor, the first and second order derivative of call option price with respect to strike, then we can obtain local volatility as :

$$\sigma_{local}(S_t,t) = 2\frac{\partial_T C(K,T) + rK\partial_K C(K,T)}{K^2\partial_{KK}C(K,T)}\bigg|_{K=S_t,T=t}$$

local volatility is indiced with S_t and t, while Black Scholes volatility is indiced with K and T