Optimal block execution of trades

Petar Maymounkov

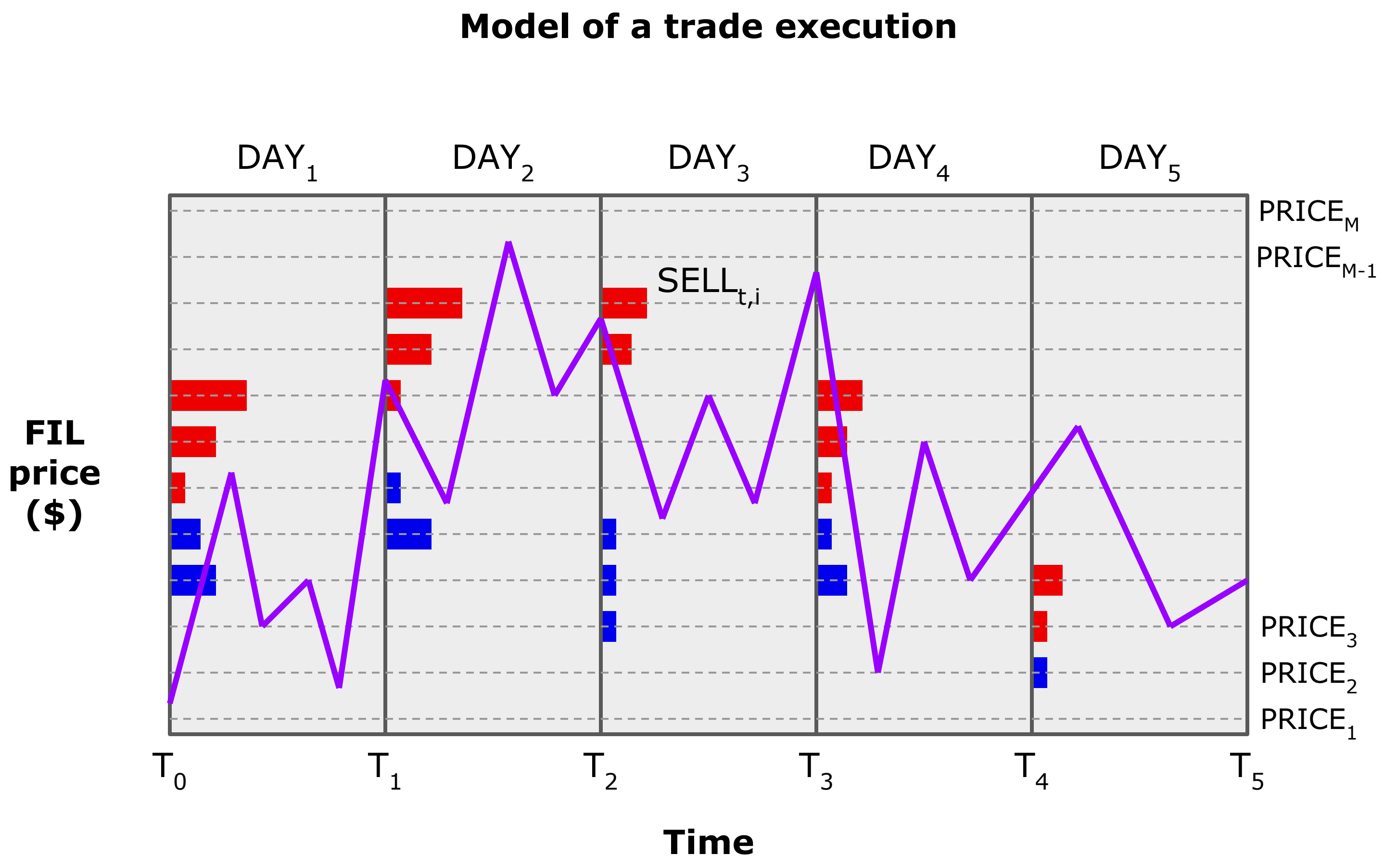
PL, July 2021

Problem statement, first stab

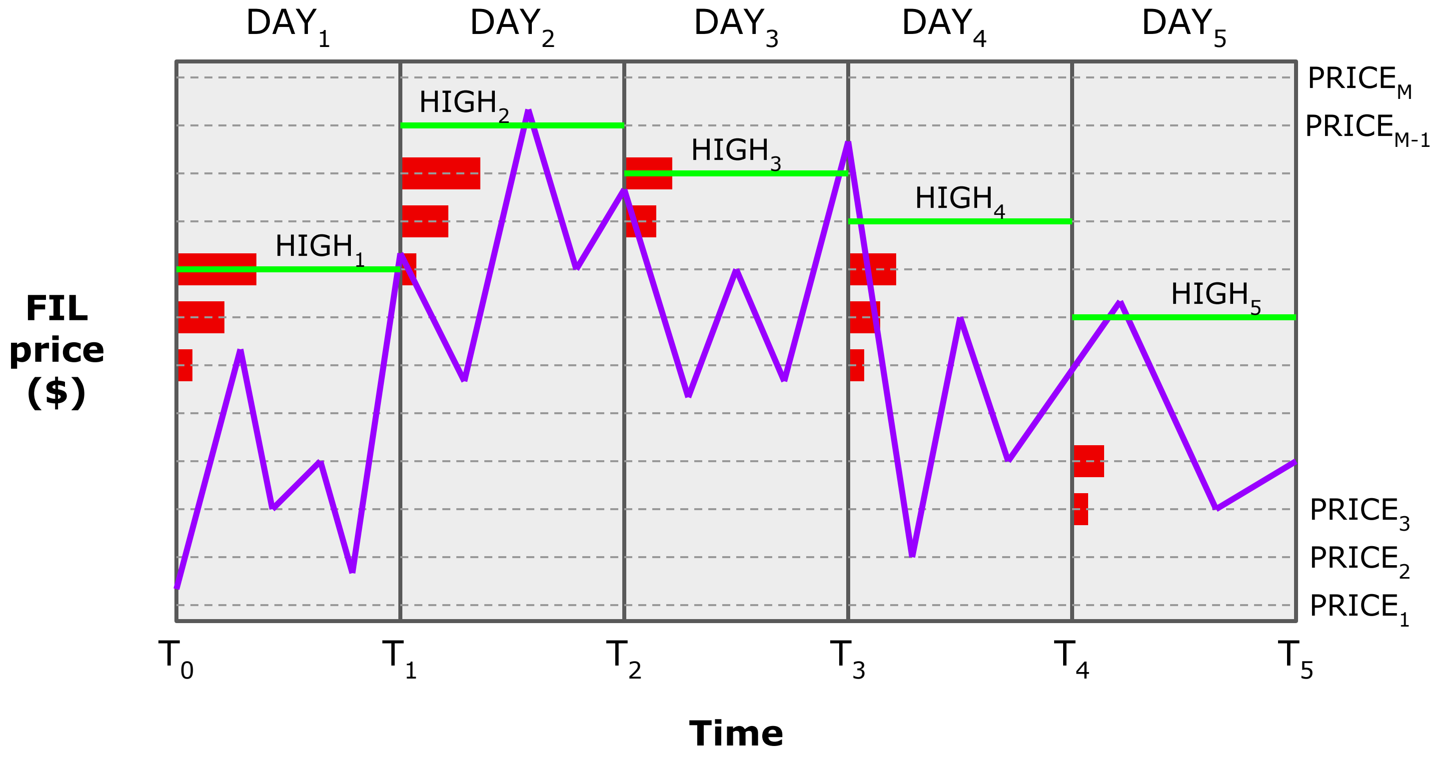
Traditionally, OTC service by brokers to institutional clients.
Electronic version, around 2005.
Now crypto OTC services.

Sell 1 FIL before a deadline for as much USD as possible, given a (probabilistic) model of FIL/USD.

(Market Making is block execution on both sides of the market.)


Model of a trade execution
DAY$_1$
DAY$_2$
DAY$_3$
DAY$_4$
DAY$_5$
PRICE$_M$
PRICE$_{M-1}$
SELL$_{t,i}$
FIL price ($)
PRICE$_3$
PRICE$_2$
PRICE$_1$
T$_0$
T$_1$
T$_2$
T$_3$
T$_4$
T$_5$
Time

If only selling, HIGH PRICE between syncs determines limit order matches.



Assumptions and objective

Assume given PROB_i: Probability that the next day highest price is PRICE_i.

Let USD_t = dollars earned from selling FIL 1 between T_t and T_{LAST} using strategy.

Objective:

maximize $E[USD_0]$ over the choices of $\overbrace{SELL_{i,t}}^{\text{the unknowns (the strategy)}}$

such that $SELL_{i,t} >= 0$ for all i,t
and $\sum_i SELL_{i,t} <= 1$ for all t

Solution

$$USD_{LAST} = 0$$

$$USD_t = \text{“matched orders between } T_t \text{ and } T_{t+1}\text{”} + \text{“unsold FIL”} \cdot USD_{t+1}$$

$$= \text{“orders with } SELL_{i,t} <= HIGH_t\text{”} + \text{“unsold FIL”} \cdot USD_{t+1}$$

$$= \Sigma_{p:price} \underbrace{[HIGH_t = PRICE_p]}_{\substack{\text{indicator variable:} \\ \text{1 if condition met,} \\ \text{0 otherwise}}} \cdot \Big(\underbrace{\Sigma_{i<=p}\, PRICE_i \cdot SELL_{t,i}}_{\substack{\text{proceeds from} \\ \text{matched sell} \\ \text{orders}}} + \underbrace{(1 - \Sigma_{i<=p} SELL_{t,i})}_{\substack{\text{remaining FIL} \\ \text{(not matched)}}} \cdot USD_{t+1} \Big)$$

$$E[USD_t] = \Sigma_{p:price}\, PROB_p \cdot \big(\Sigma_{i<=p}\, PRICE_i \cdot SELL_{t,i} + (1 - \Sigma_{i<=p} SELL_{t,i}) \cdot E[USD_{t+1}] \big)$$

maximize $E[USD_t]$ over $SELL_{t,i}$ using linear programming

Implementation intermission

github.com/petar/BlockExecExpMax.jl

Discussion and improvements

Caveat: Maximizing expectation, $E[USD_0]$, can produce bi-modal (all-or-nothing) outcomes.

Intuitively, we want to maximize expectation **and** minimize variance.

However, minimizing variance contradicts maximizing expectation.

So, we must embed our preferences in a single objective function.

Sharpe ratio: $E[USD_0]$ / $StdDev[USD_0]$

Weighted: $E[USD_0]$ - $3*StdDev[USD_0]$

Optimization challenges

Maximizing Sharpe (or any variance-based objectives) is **not** linear programming.

maximize_x Sharpe **and** maximize_x Sharpe^2 **find the same x**.

$$\text{Sharpe}^2 = E[USD_0]^2 \,/\, \text{StdDev}[USD_0]^2$$

$$= E[USD_0]^2 \,/\, \text{Var}[USD_0]$$

$$= E[USD_0]^2 \,/\, (\, E[USD_0{}^2] - E[USD_0]^2 \,)$$

Therefore, the Sharpe maximization is **rational** and **quadratic** (in the unknowns $SELL_{t,i}$).

Possible approaches

Linear/quadratic/conic/semidefinite programming find global optima,

but are harder to work with: objective needs to be rewritten cleverly.

Alternatively, one could find an optimum for the objective (e.g. Sharpe) using gradient methods like **(Stochastic) Gradient Descent**. Global optimum is not always found, but no need to rewrite the problem objective.

Takeaway

Every “control” problem can be somehow written as a mathematical optimization problem:

maximize f(x), subject to a(x) >= 0, b(x) >= 0, etc.

With a little effort you can find a decent solution to any optimization problem.

(Use Julia as a one-stop-shop for math.)