Algorithmic Trading

-

This problem is aimed at giving a flavor of various events and controls involved in real world trading. With the goal to design an optimal trading strategy for fulfilling client orders, while satisfying the constraints posed by marketplace.

Devise a trading algorithm to get the **Aggregate-best-trade-executions** (cheapest buys and highest sells) to fulfill the given client orders under certain constraints. Real time market data will be provided for the products to be traded. A Client comes in with the objective of buying / selling certain quantity on a basket of stocks [probably multiple requests during the trading time window]. Your goal is to help them get the best possible trades based on the market data available to you.

The scenario is as follows:

- You are given 3 different tradable products P1, P2, P3; such that P1 can be traded on exchange E1; P2 can be traded on exchanges E1 and E2; and P3 can be traded on exchanges E1, E2, E3, E4.

- Real time bid and ask prices and corresponding bid and ask quantities are available for these products from different exchanges as applicable. This information is available at an interval of one second. There may be intervals where this data is missing for one or more products.

- The client orders are given in the form of product to be traded, quantity to buy or sell, and time window during which the order has to be completed. You may split the original client order into chunks of orders for trading and may trade on any of the available exchanges.
  Note: If a client order is to buy a product, you can only buy that product till the order is completed, and not sell in between. Similarly if a client order is to sell a product, you cannot buy that product till the order is complete.

- You have to write a program for trading the given products on behalf of the client with the objective of getting cheapest buys and highest sells in such a way that the constraints given below are met.

Constraints

- ***Final Position:*** Net positions in all products should be equal to the net client orders at the end of the trading window, that is, all client orders should get completely filled.
- ***Order size limit:*** At any time interval you cannot sell more than the bid quantity, and you cannot buy more than the ask quantity; this holds for all exchanges and products.
- ***Trade size limit:*** At any interval, the number of stocks bought or sold on an exchange cannot exceed 1,000,000 for P1; 20,000 for P2; 20,000 for P3.
- ***Exchange position limit:*** The absolute number of stocks of a product cannot exceed 70,000 for P2; 140,000 for P3 (No such limit for P1).
- ***Restricted product:*** Typically, right after an important public announcement related to a company, its stock prices tend to move very rapidly. In order to avoid a possibility of undue advantage to someone closely related to the announcement, a list of products is maintained that cannot be traded for a specified time window. P1 is such a product that becomes restricted for certain time during the trading window (details given in **Input** section). P1 cannot be traded (bought or sold) during the time that it remains in the restricted trading list (RTL).

Evaluation

Your output should be submitted as a csv file which will be used to check for all the applicable constraints and calculate the total execution price for each buy and sell client order. The trade orders in your output should satisfy all the given constraints and fulfill the client orders in the specified time duration. The evaluation will be based on the approach and total execution price achieved for each order.
Note: For making a trade decision at a timestamp T, your solution can use any market information till timestamp T

only; and no market information after timestamp T. Any forward-looking solutions will be disqualified.

Dataset

- At the start of the competition, you will be given a sample dataset with market data for a duration of 120 minutes (t = 0 to t = 120 minutes for each interval of one second), and client order to be fulfilled in each interval of 30 minutes (starting from t = 30 minutes). You are free to use this data to set up you trading strategy/algorithm and write/test your code.
- Your code then needs to be run for five different datasets that will be provided 3 hours before the end of the competition. Each of these datasets will contain market data for a duration of 120 minutes (t = 0 to t = 120 minutes), and client orders to be fulfilled in each interval of 30 minutes (starting from t = 30 minutes). **Each of the five datasets and the initial sample dataset are independent from each other.**

Input

Filename	Fields*	Available from
Market_Data_Sample.csv	Timestamp, Exchange, Product, Bid Price, Bid Quantity, Ask Price, Ask Quantity	9pm, Sep 28, 2013
Client_Orders_Sample.csv	Start_time, End_time, Product, Quantity	9pm, Sep 28, 2013
Market_Data_A.csv	Timestamp, Exchange, Product, Bid Price, Bid Quantity, Ask Price, Ask Quantity	6pm, Sep 30, 2013
Market_Data_B.csv		
Market_Data_C.csv		
Market_Data_D.csv		
Market_Data_E.csv		
Client_Orders_A.csv	Start_time, End_time, Product, Quantity	6pm, Sep 30, 2013
Client_Orders_B.csv		
Client_Orders_C.csv		
Client_Orders_D.csv		
Client_Orders_E.csv		

Market Data

- Timestamp is given as Date hh:mm:ss.000 am/pm
- Bid and Ask Prices are given as decimal numbers rounded to 4 decimal places
- Bid and Ask quantities are integers
- *Market data will occasionally have RTL lines specified as below:
  - ***Timestamp1, RTL_begin, P1, 0, 0, 0, 0***
  - ***Timestamp2, RTL_end, P1, 0, 0, 0, 0***
  - From Timestamp1 to Timestamp2, P1 remains under RTL and thus cannot be traded

Client Orders

- Start_time and End_time are given as hh:mm; the client order has to be completed within this time window
- Quantity is positive for a buy order and negative for a sell order

Output

Zipped file to be submitted by 9pm, Sep 30, 2013 with the following contents:

Filename	**Fields**
ModelDocumentation	-
SourceCode	(zipped folder with all relevant files)
output_sample.csv	Timestamp, Exchange, Product, Quantity
output _A.csv	
output _B.csv	
output _C.csv	
output _D.csv	
output _E.csv	

- At each interval of one second during the trading time period, your code should give a trade order in the format: ***Timestamp, Exchange, Product, Quantity***

- Quantity is positive for a buy order and negative for a sell order. For example: a line from your output {*Timestamp3, E2, P3, -250}* will be interpreted as: Sell 250 stocks of product P3 on exchange E2 at Timestamp3 at the last available price; bid for sell order and ask for buy order.
- **Note:** If at an interval you do not wish to trade, you should still specify the timestamp in the output but with other fields as empty NULL)

- Model Documentation should contain following parts:
    - Description of the approach
    - Algorithm or pseudo code
    - Assumptions (if any)