Work of 200 $. (maximum budget).

Kindly only serious and experienced professional programmers in the MQL5 language.
I DO NOT ACCEPT BUDGET INCREASES DURING THE EXECUTION OF THE WORK. THANK YOU
THE MQL5 FILE WILL BE DELIVERED ONLY AFTER THE FREELANCER HAS ACCEPTED THE MILESTONE
PLEASE THE FREELANCERS BEFORE MAKING THEIR BID READ THE WORK FILE CAREFULLY
WE WILL DISCUSS ALL POINTS (INDIVIDUALLY) IN CHAT BEFORE THE START OF WORK
THE PDF FILE OF THE JOB WILL BE DELIVERED ONLY IF INTERESTED

Work overview

1) Profit calculation for closing positions
points a);...c);
2) Rebalancing positions (buy / sell)
points a);...g);
3) Start time; End time
points a);..d);
4) Total profit / total loss / trailing balance (points a; ....g;)

I'm available for any clarification and for any example.
We will analyze each point togheter. each point, will be indicated with example and indications
The programmer must deliver the “source code” (source code, of all demos and all finished work).
Before sending the “source code”, please carefully check the perfect and complete integration with existing EA.
The EA will send (for work) to the programmer, works correctly in all its parts (inputs).
To check work of the programmer, can only be done in “open markets”. To be verified, need minimum three days " open market"
The programmer must verify that the work is fully integrated with all EA parameters, before delivering the work (real - time verification).
The programmer must deliver a job with a clean code (total absence of bugs). And give assistance even after the job done.
The programmer will test and record EA video (real – time demo) in action (at all work points). Only if required
The programmer if check that there are problems, you must prove them via video recorded
EA must work in demo (real time), and in backtesting. And for real money. EA must work in forex, cfd (indices and stock) and futures (CME)
EA (file mql) will be send to the programmer only after the milestone created : And after the programmer has accepted the milestone .
After the programmer has delivered the finished work (source code), the milestone will be released.
The deadline for the programmer ’s job will be doubled . For example , the programmer indicates 5 days for the execution and delivery of the work.
I'll give it 10 days.
If the work is not delivered within 10 days, the contract expires. I have no more interest in continuing the contract. And then it is cancelled.

examples

example if there are open 20 positions 14 BUY and 6 SELL.
I want to reduce the higher positions (14 BUY) with the calculation of the percentage of 40% on the SELL (smaller positions) 6*40/100=2,4
6+2.4= 8.4 buy positions allowed (rounded to 2)
must be reduced to 8 the BUY positons
and then the system must close 6 buy (were 14)
So, after closing 6 buy, remain open: 6 SELL and 8 BUY positions.

1) PROFIT CALCULATION FOR CLOSING POSITIONS

This filter calculates the profit, between closed and still open positions.
Example, I enter:
- Start time profit calculation 1:00
- end time 14:00
- Profit calculation TRUE
- every X profit => 50€
  - closing all positions TRUE
  -

The inserted filter (above) means:
if at 4:45, with the closing of one or more positions (in the windows time) the system has arrived to make a profit of 110 € and in the portfolio, we have other positions still open, that at the same time (4:45) are accumulating a loss of – 60€.

With the positions close of the 4.45, we have:
110 € profit realised (position already closed).
-60€ accumulated loss from still open positions.
110 – 60 = 50€
Having onserted in the filter (above) “every of profit” the value of 50€

```
7964
7965       if(stopValue)
7966         {
7967          Print(StringFormat("SL pips before validation = %d", stopValue));
7968          stopValue = getValidStopPoints(stopValue);
7969          Print(StringFormat("SL pips after validation = %d", stopValue));
7970          double stopPrice =  openPrice - getSignum(positionType) * stopValue * _Point;
7971          Print(StringFormat("SL price before correction = %.8f", stopPrice));
7972
7973          Print(StringFormat("Tick size = %.5f", SymbolInfoDouble(Symbol(), SYMBOL_TRADE_TICK_SIZE)));
7974          double tickSize = SymbolInfoDouble(Symbol(), SYMBOL_TRADE_TICK_SIZE);
7975          int ratio = (int)MathFloor(stopPrice / tickSize);
7976          Print(StringFormat("Ratio = %d", ratio));
7977          Print(StringFormat("Price for comparing = %.8f", tickSize * ratio));
7978          if(MathAbs(tickSize * ratio - stopPrice) > 0.0000001)
7979            {
7980             stopPrice = ratio * tickSize;
7981            }
7982
7983          Print(StringFormat("SL price after correction = %.8f", stopPrice));
7984          Print(StringFormat("SL price after normalization = %.5f", NormalizeDouble(stopPrice, _Digits)));
7985         }
7986      }
7987
7988    Print("============================================");
7989   }
7990 //+------------------------------------------------------------------+
```

WORK TO BE DONE

Add:

3)Start time “profit calculation” **value** (if not indicated, follows the time inserted in the “trade time windows”)
3)End time “profit calculation” **value**
1) Profit calculation for closing positions **True / False** (cumulative calculation between closed and still open position)
1a) every “X” of profit => **value** (example: every 100 €) continuous calculation.
1b)Closing all positions (all buy and all sell) **True / False** (closing of all positions currently in the portfolio) and after the EA continues to work.
1c) Closing rebalancing positions (buy sell) **True / False** (equalize the buy sell positions,) **example below**

After these points (above), the programmer delivers the source code of the made points.
In this way the customer can try it and optimize the continuation of the work.

3a)Start time “ Rebalance larger positions” **value** (if not indicated, follows the time inserted in the “trade time windows”)
3a)End time “ Rebalance larger positions” **value**
2) Rebalance larger positions (present in portfolio) **True / false** if there is a (not equal) difference between buy and sell. Reduces the largest **example below**
2a) Rebalance only if the total position has a loss => **value** (Rebalance only if the total position still open, have a loss generated => X)
2b) Reduce larger positions to the percentage =< **value** (applied percentage of reduction, calculated on smaller positions) and immediately after continue to work again) **example below**
2c) Reduce, only if the total number of positions still open is => **value** (number of total positions : buy and sell; or buy only; or sell only)

After these points (above), the programmer delivers the source code of the made points.
In this way the customer can try it and optimize the continuation of the work.

3b)Start time “ Profit Realised” **value** (if not indicated, follows the time inserted in the “trade time windows”)
3b)End time “ Profit Realised” **value**
2d) For each profit realised with closed position => **True / false**
2e) For each profit realised with closed position => **value**
2f) close still open position that have a loss =< **value** **example below**
2g) If there are only (still open position) all buy or all sell positions **True / false**
And the loss cumulated is => **value** (ex 50€)
close the positions of the % =< **value** (50%) Then close positions for =<25€ (always starting with the best percentage even if negative**)**
After these points (above), the programmer delivers the source code of the made points.
In this way the customer can try it and optimize the continuation of the work.

3c)Start time “ Cumulative Profit” **value** (if not indicated, follows the time inserted in the “trade time windows”)
3c)End time “ Cumulative Profit” **value**
4) If we have a cumulative profit => of X, close all positions **True / false** (cumulative profit, calculated between closed position and still open position) and does not negotiate until the next session)
4a) cumulative profit => **value**
4b) If we have a cumulative loss => of Y, close all positions **True / false** (cumulative profit, calculated between closed position and still open position) and does not negotiate until the next session)
4c) cumulative loss => of the **value**

3d)Start time “ Trailing on the cumulative profit ” **value** (if not indicated, follows the time inserted in the “trade time windows”)
3d)End time “ Trailing on the cumulative profit ” **value**
4d) Trailing on the cumulative profit **True / False**
4e) with profit realized => **value** (cumulative profit, calculated between closed position and still open position)
4f) apply trailing of the percentage (%) **value** **example below**
4g) After trailing stop, end of trading for the current session **True / False** (resumes at the next session)

After all the run points, the programmer delivers the source code . The customer to try, as indicated, requires from 3 days of open markets (minimum).

EXAMPLES

1c ex: reached the tp of 100€ (indicated above in value), realized with the positions already closed. And still remain open in portfoglio, 22 positions (7 buy and 15 sell). the system (EA) must close 8 sell, so then will remain open 7 Buy and 7 Sell (equalizing positions).
Which of the 8 sales positions should be closed ? must close the eight sell with the best performance.

2 ex: It's the same mechanism of calculation seen in the example of point 1c) with the difference:
- that in point 1) the rebalancing is done by the EA only and exclusively when a TP of X is reached (as entered in value 100€)
- instead, in point 2) the rebalancing is done by the EA always, when there is a difference between Buy and Sell still open in the portfolio.
- The points 1 c) and 2), can be inserted individually or together

2b ex: if the total open positions, in portfolio, are 15 (9 buy and 6 sell). I enter 40% value 2b): This means that the largest position (9 buy) cannot be larger than 40% compared to the smallest positions: 6*40/100= 2,4 (rounded to 2), that is, 6 (lowest position) + 2 (+40% of the smallest position) = 8 Buy position allowed.
Then, the EA must close 1 buy position in the portfolio with the best performance (must always close positions with the best percentage. Even if the best percentage is negative

2e/2f ex: I inserted in the value 2e) 10 € and in the value 2f) 5€. It means, for each => 10 € of closed position in profit, the EA closes =< 5€ of loss position (initial of the best percentage). If to reach the indicated value of =< to 5€, need 3 loss positions, the EA will close 3 positions still open (of any type, buy or sell). The criterion to follow is always the best percentage loss.

4e/4f ex: if I have inserted in value 4e) 500€ and in value 4f) 20%
means, that when it reached a cumulative profit of 500€, the trailing of 100€ starts (500*20/100 = 100€)
so when the cumulative profit drops = to 400€ closes all the positions still open