# 经典策略解读

## 策略名

基于平移的高低点均值通道与K线中值突破的系统

## 策略内容

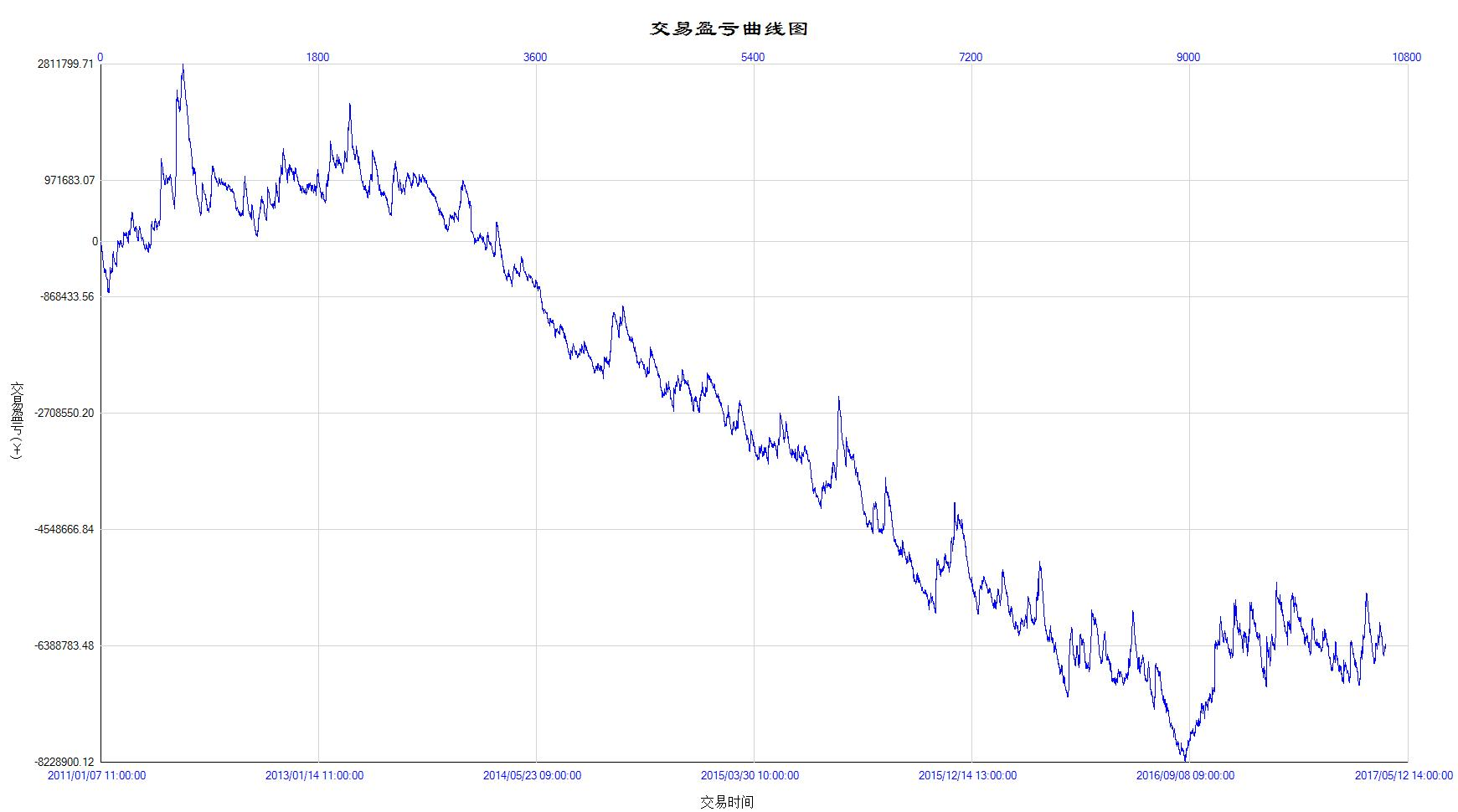
1. 计算K线的高点与低点的均线，对均线往前平移
2. 开多： 如果当根K线的中点高于上一根K线的高点，同时，上根K线的收盘价 > N周期前高点的MA，开多仓
3. 开空： 如果当根K线的中点低于上一根K线的低点，同时，上根K线的收盘价 < N 周期前低点的MA，开空仓
4. 出场条件：开仓后，5根K线内用中轨只算，5根K线后用外轨止损

## 策略代码一

Params  
 Numeric AvgLen(20); // 高低点均线计算周期  
 Numeric AbsDisp(5); // 高低点均线前移周期  
 Numeric ExitBar(5); // 止损周期参数，该周期以前中轨止损，以后外轨止损  
Vars  
 NumericSeries UpperAvg(0); // 通道上轨  
 NumericSeries LowerAvg(0); // 通道下轨  
 NumericSeries ExitAvg(0); // 通道中轨  
 BoolSeries RangeLeadB(False); // 是否RangeLead  
 BoolSeries RangeLeadS(False); // 是否RangeLead  
 NumericSeries MedianPrice; // K线中价  
 Numeric minpoint; // 最小变动价位  
 NumericSeries range; // 振幅  
Begin  
 // 集合竞价  
 If(BarStatus == 2 And IsCallAuctionTime) Return;  
  
 // 指标计算  
 minpoint = Minmove\*PriceScale; // 最小变动价位  
 range = High - Low;  
 UpperAvg = Average(High[AbsDisp], AvgLen); // 计算N周期前高点的MA，N=参数AbsDisp  
 LowerAvg = Average(Low[AbsDisp], AvgLen); // 计算N周期前低点的MA，N=参数AbsDisp  
 MedianPrice = (High + Low)\*0.5; // 计算K线中点  
 ExitAvg = Average(MedianPrice[AbsDisp], AvgLen); // 计算N周期前K线中点的MA，N=参数AbsDisp  
 // PlotNumeric("lineu",UpperAvg);   
 // PlotNumeric("lineM",ExitAvg);  
 // PlotNumeric("lingd",LowerAvg);  
 RangeLeadB = MedianPrice > High[1] and Range > Range[1]; // 当K线中点大于前一根K线高点并且振幅〉上一根振幅时，RangeLeadB为真  
 RangeLeadS = MedianPrice < Low[1] and Range > Range[1]; // 当K线中点小于前一根K线低点并且振幅〉上一根振幅时，RangeLeadS为真  
  
 // 系统入场  
 If(MarketPosition == 0)  
 {  
 If(RangeLeadB[1] and Close[1] > UpperAvg[1]) // 上根K线RangeLeadB为真，并且上一根收盘价大于N周期前高点的MA，当前无多仓，则开多仓  
 {  
 Buy(0,Open);  
 }  
 If(RangeLeadS[1] and Close[1] < LowerAvg[1]) // 上根K线RangeLeadS为真，并且上一根收盘价小于N周期前低点的MA，当前无空仓，则开空仓  
 {  
 SellShort(0,Open);  
 }  
 }  
  
 // 系统出场  
 If(MarketPosition == 1 and BarsSinceEntry > 0) // 开仓后N根K线内用中轨止损，N根K线后用上轨止损，N=参数ExitBar  
 {  
 If(BarsSinceEntry <= ExitBar )   
 {  
 If(Low <= ExitAvg)   
 {  
 Sell(0,Min(Open,ExitAvg));  
 }  
 }  
 Else If(BarsSinceEntry > ExitBar)  
 {  
 If(Low <= UpperAvg - minpoint)   
 {  
 Sell(0,Min(Open,UpperAvg - minpoint));  
 }  
 }  
 }  
  
 If(MarketPosition == -1 and BarsSinceEntry > 0) // 开仓后N根K线内用中轨止损，N根K线后用上轨止损，N=参数ExitBar  
 {  
 If(BarsSinceEntry <= ExitBar)   
 {  
 If(High >= ExitAvg)   
 {  
 Buytocover(0,Max(Open,ExitAvg));  
 }  
 }  
 Else If(BarsSinceEntry > ExitBar)  
 {  
 If(High >= LowerAvg + minpoint)   
 {  
 Buytocover(0,Max(Open,LowerAvg + minpoint));  
 }  
 }  
 }  
  
End

## 回测表现

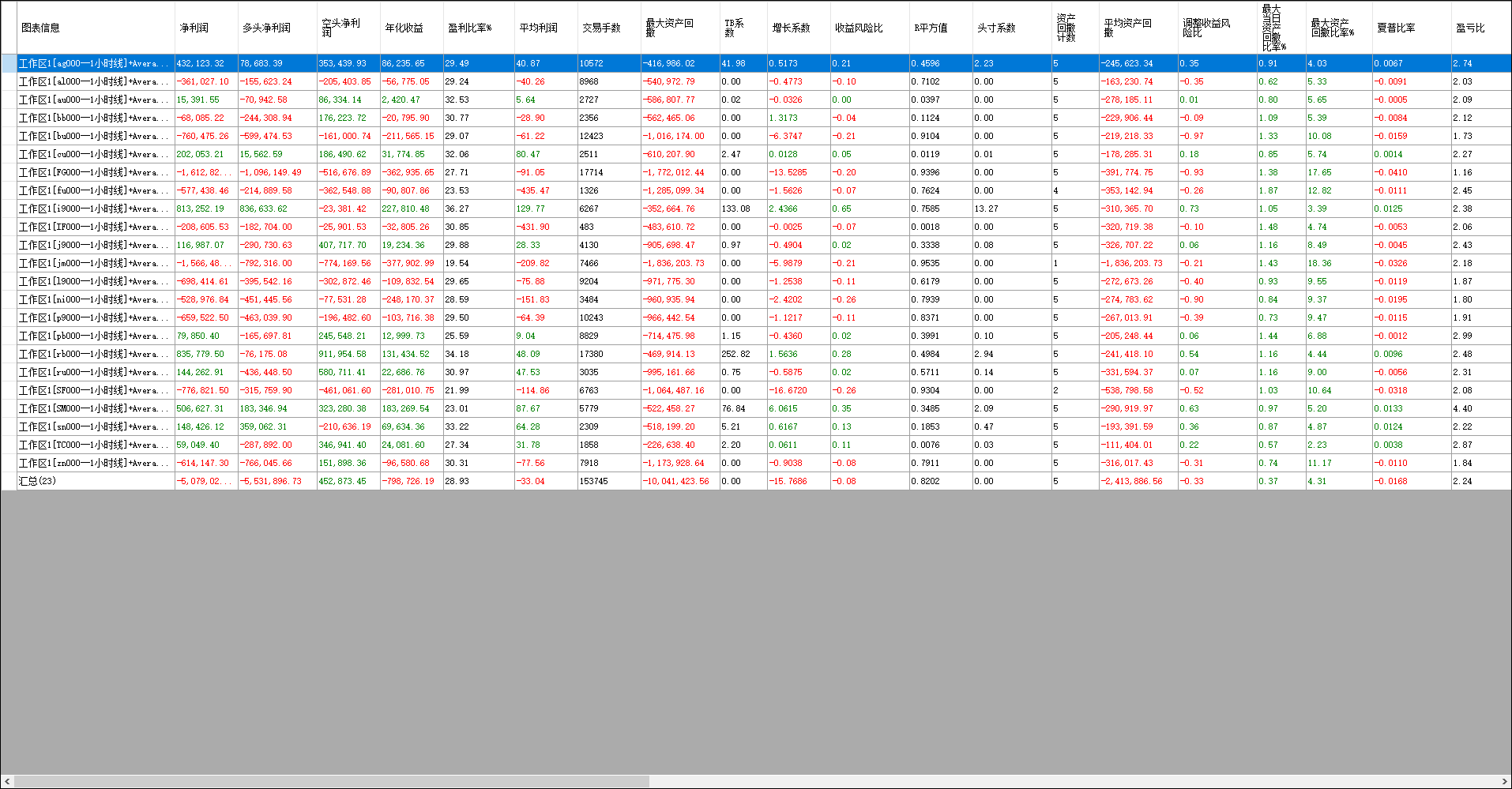
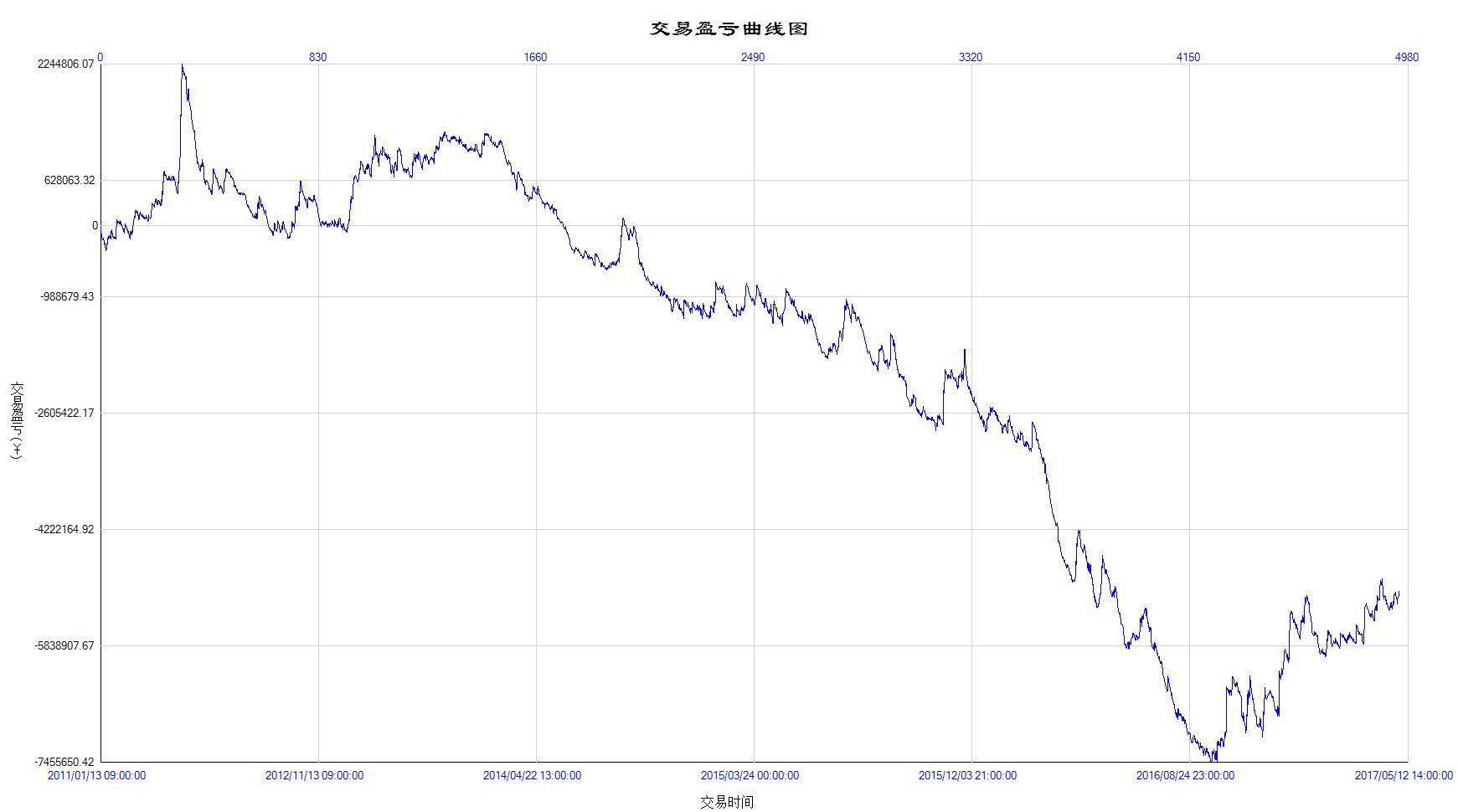
统一使用回测区间为2011年1月4日至2017年5月11日数据，手续费为 1%%，滑点为1跳，多品种回测采用固定金额， 每个品种固定金额均为 200,000

# 基于经典策略的改进

## 加入ROC过滤

考虑到回测中看到开平仓太过频繁，是不是加入一个ROC过滤，对趋势行情加以确认和对 震荡行情加以过滤，采用 5 日 ROC与12日ROC均线系统对上述经典策略进行改进，回测 结果如下，可以看到确实显著减小了开平仓动作，使得亏损额大幅减小。

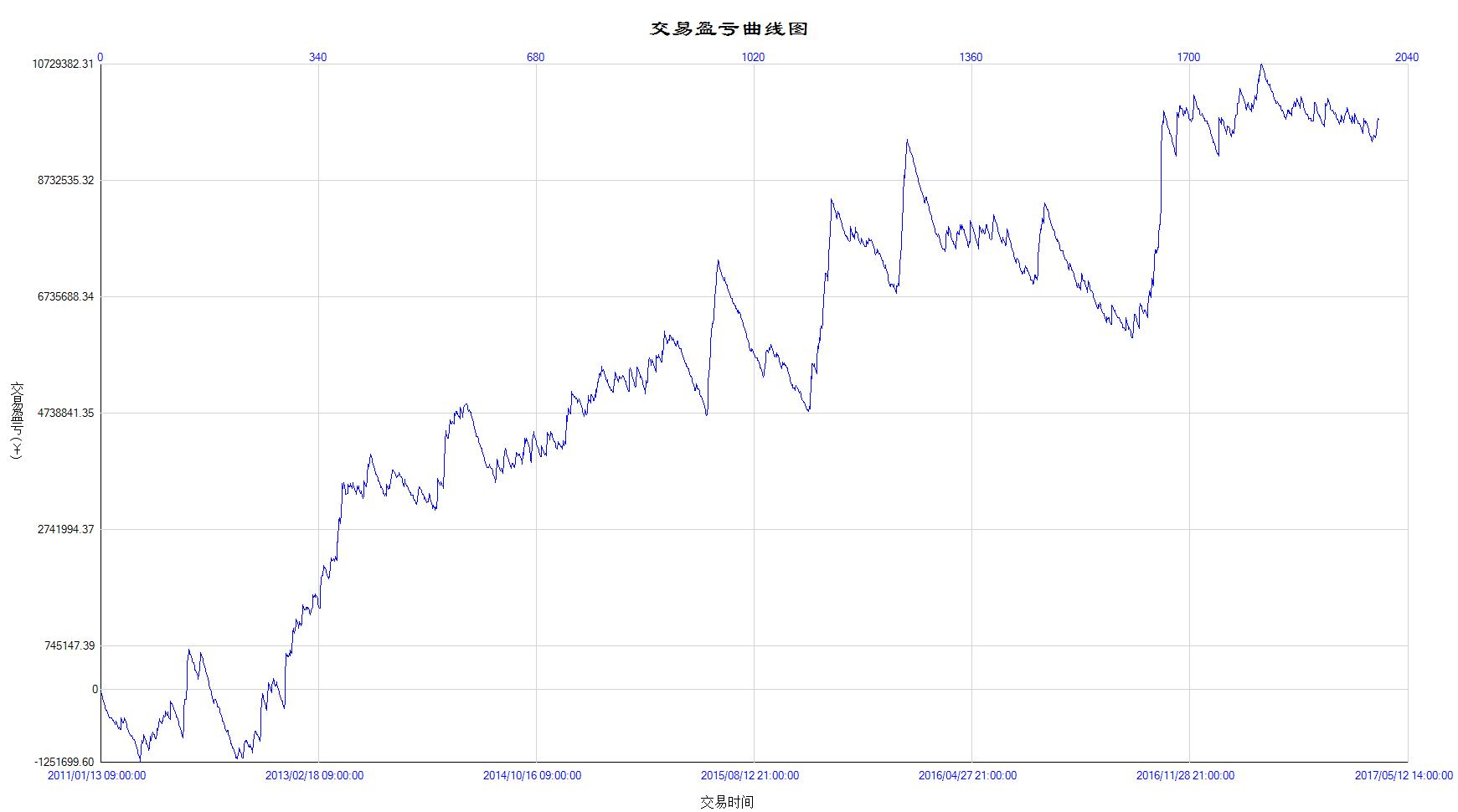
 

ROC过滤代码块如下：

...  
  
Numeric Length1(5);  
Numeric Length2(12);  
...  
  
ROCValue = (Close - Close[Length1])/Close[Length1]\*100;  
ROCValueMA = AverageFC(ROCValue, Length2);  
...  
  
RangeLeadB = CrossOver(ROCValue, ROCValueMA) && MedianPrice > High[1] and Range > Range[1]; // 当K线中点大于前一根K线高点并且振幅〉上一根振幅时，RangeLeadB为真  
 RangeLeadS = CrossUnder(ROCValue, ROCValueMA) && MedianPrice < Low[1] and Range > Range[1]; // 当K线中点小于前一根K线低点并且振幅〉上一根振幅时，RangeLeadS为真  
  
...

## 止损方式改进

从回测资产变化图，可以看到回测曲线的回撤比较大，考虑到原始策略止损系统比较简单，是不是可以加入三级止损来改进 代码效果。

止损代码块如下：

...  
 If(BarsSinceEntry >= 1){  
 HigherAfterEntry = Max(HigherAfterEntry[1], High[1]);  
 LowerAfterEntry = Min(LowerAfterEntry[1], Low[1]);  
 }  
  
...  
 If(MarketPosition == 1){  
 StopLine = EntryPrice\*(1-InitialStop/1000);  
 If(HigherAfterEntry >= EntryPrice\*(1+BreakEvenStop/1000)){  
 StopLine = EntryPrice;  
 }  
 If(StopLine < HigherAfterEntry\*(1-TrailingStop/1000)){  
 StopLine = HigherAfterEntry\*(1-TrailingStop/1000);  
 }  
 If(Low <= StopLine && BarsSinceEntry >= 1){  
 Sell(0, StopLine);  
 }  
 }  
 // 持有空仓，三级止损  
 If(MarketPosition == -1){  
 StopLine = EntryPrice\*(1+InitialStop/1000);  
 If(LowerAfterEntry <= EntryPrice\*(1-BreakEvenStop/1000)){  
 StopLine = EntryPrice;  
 }  
 If(StopLine > LowerAfterEntry\*(1+TrailingStop/1000)){  
 StopLine = LowerAfterEntry\*(1+TrailingStop/1000);  
 }  
 If(High >= StopLine && BarsSinceEntry >= 1){  
 BuyToCover(0, StopLine);  
 }  
 }