

浙江工业大学

量化策略开发与程序化交易课程设计 实验报告

(2021 级)



多策略交易

学生姓名: 温家伟

学生学号: 202103151422

学科专业: 大数据分析 2101 班

所在学院: 理学院

提交日期: 2023 年 11 月 19 日

目录

1	实验策略	2
2	代码分析	2
2.1	除权换月，初始资金，手续费，滑点的设置	2
2.2	设置头寸	2
2.3	利用大周期过滤	3
2.4	小周期时的具体策略	3
2.4.1	针对趋势，利用四周加双均线策略	4
2.4.2	布林通道	5
3	程序化交易策略的实现及应用	9
3.1	策略代码	9
3.2	实盘操盘及其分析	19
3.3	稳定的回测效果	20

1 实验策略

以四周交易策略为主策略，加入双均线策略，布林通道，大周期过滤，跟踪止盈，固定止损策略。

策略的总体思路为：用大周期过滤小周期中频繁的交易，在过滤后的情况使用四周交易策略，考虑到四周使用于趋势策略又已经将震荡过滤，加入双均线策略，出现‘金叉’并突破高点，买入；出现‘死叉’并突破低点，卖出。考虑过滤的周期，较为震荡，使用平滑后的布林通道，用于震荡时刻买入和卖出的依据。

2 代码分析

2.1 除权换月，初始资金，手续费，滑点的设置

```
OnInit()
{
    Range[0:DataCount-1]
    {
        //=====除权换月相关设置=====
        AddDataFlag(Enum_Data_RolloverBackward()); //设置后复权
        AddDataFlag(Enum_Data_RolloverRealPrice()); //设置映射真实价格
        AddDataFlag(Enum_Data_AutoSwapPosition()); //设置自动换仓
        AddDataFlag(Enum_Data_IgnoreSwapSignalCalc()); //设置忽略换仓信号计算
        //=====交易相关设置=====
        SetInitCapital(1000000); //设置初始资金为100万
        SetBeginBarMaxCount(1);
        //=====交易相关设置=====
        SetInitCapital(1000000); //设置初始资金为100万
        SetCommissionRate(BitOr(Enum_Rate_FreeOfExitToday, Enum_Rate_ByFillAmount), 5); //设置手续费率为成交金额的5%，不收平今。
        SetSlippage(Enum_Rate_PointPerHand, 2); //设置滑点为2跳/手
    }
}
```

图 1: 一些设置

2.2 设置头寸

```
Range[0:0]
{
    myprice=Open; //这里使用open,更为精确的是使用委托价格
    lots=IntPart(money*10000/(myprice*contractunit*BigPointValue)); //计算开仓手数
    ma60=Average(Close[1], malength);
}
```

图 2: 设置头寸

2.3 利用大周期过滤

```
OnBar(ArrayRef<Integer> indexes)
{
    Range[0:0]
    {
        myprice=Open; //这里使用open,更为精确的是使用委托价格
        lots=IntPart(money*10000/(myprice*contractunit*BigPointValue)); //计算开仓手数
        ma60=Average(Close[1],malength);
    }
    ...
}
```

图 3: 利用大周期过滤

2.4 小周期时的具体策略

考虑到实际操作时存在订单长时间未成交的情况，引入撤单操作：

```
if(BarStatus == 2 && !openOrder)
{ //发平仓单
    //针对当前策略应用的帐户、商品发送委托单
    Bool ret = A_SendOrder(Enum_Buy, Enum_Exit, lots, Open, 0);
    if(ret)
    {
        openOrder = True;
    }
    cancelId=createTimer(10000);
}
```

图 4: 撤单

```
//定时器更新事件函数, 参数id表示定时器的编号, millsecs表示定时间的间隔毫秒值
OnTimer(Integer id,Integer intervalMillsecs)
{
    if(id == cancelId)
    {
        //撤单
        Array<Integer> orderIds;
        //获取未完成的报单数组
        Bool ret = A_GetUnFillOrderIds(orderIds, "", 0);
        Integer i;
        For i = 0 To GetArraySize(orderIds) - 1
        {
            Print("accountId:" + A_AccountID(0) + ",orderId:" + Text(orderIds[i]));
            //针对当前策略应用的帐户、商品发送撤单指令
            ret = A_DeleteOrder(orderIds[i], 0);
            Print("A_DeleteOrder:" + IIFString(ret, "True", "False"));
        }
        StopTimer(cancelId);
        openOrder = False;
    }
}
```

图 5: 撤单

2.4.1 针对趋势, 利用四周加双均线策略

```

.
//四周
If(MarketPosition<>1 And High>=highline And Close[1]>data0.ma60 And AvgValue1[1] > AvgValue2[1])
{
    Buy(lots,Max(Open,highline));
    if(BarStatus == 2 && !openOrder)
    { //发开仓单
        //针对当前策略应用的帐户、商品发送委托单
        Bool ret = A_SendOrder(Enum_Buy, Enum_Entry, lots, Open, 0);
        if(ret)
        {
            openOrder = True;
        }
        cancelId=createTimer(10000);
    }
}
If(MarketPosition<>-1 And Low<=lowline And Close[1]<data0.ma60 And AvgValue1[1] < AvgValue2[1])
{
    SellShort(lots,Min(Open,lowline));
    if(BarStatus == 2 && !openOrder)
    { //发开仓单
        //针对当前策略应用的帐户、商品发送委托单
        Bool ret = A_SendOrder(Enum_sell, Enum_Entry, lots, Open, 0);
        if(ret)
        {
            openOrder = True;
        }
        cancelId=createTimer(10000);
    }
}
.

```

图 6: 四周加双均线

在过滤后的情况使用四周交易策略，加入双均线策略，出现‘金叉’并突破高点，买入；出现死叉并突破地点，卖出。

2.4.2 布林通道

针对震荡，使用改进后的布林通道，下面详细介绍改进方法：

针对上轨和下轨，进行平滑处理，使其钝化，不需要频繁计算。

```
MidLine = AverageFC(Close,Length);  
Band = StandardDev(Close,Length,2);  
UpLine = MidLine + Offset * Band;  
DownLine = MidLine - Offset * Band;  
minpoint = MinMove*PriceScale;  
//平滑处理  
UpLine_ = Highest(UpLine,3);  
DownLine_ = Lowest(DownLine,3);
```

图 7: 平滑处理

买入卖出时刻，当前一次收盘价小于下轨，当前收盘价大于下轨，收盘价大于开盘价，我们认为未来趋势为上升，执行买入；当前一次收盘价高于上轨，当前收盘价低于上轨，收盘价低于开盘价，我们认为未来趋势为下降，执行卖出。

```
con1=Close[1]<DownLine_[1] and Close>DownLine_ and close>open;  
con2=Close[1]>UpLine_[1] and Close<UpLine_ and close<open;
```

图 8: 买入卖出时刻

在过滤周期时刻执行改进布林通道策略：

```
//布林
If(MarketPosition<>1 And High>=highline and Close[1]<data0.ma60 And con1[1])
{
    Buy(lots,Max(Open,highline));
    if(BarStatus == 2 && !openOrder)
    { //发开仓单
        //针对当前策略应用的帐户、商品发送委托单
        Bool ret = A_SendOrder(Enum_Buy, Enum_Entry, lots, Open, 0);
        if(ret)
        {
            openOrder = True;
        }
        cancelId=createTimer(10000);
    }
}
If(MarketPosition<>-1 And Low<=lowline And Close[1]>data0.ma60 And con2[1])
{
    SellShort(lots,Min(Open,lowline));
    if(BarStatus == 2 && !openOrder)
    { //发开仓单
        //针对当前策略应用的帐户、商品发送委托单
        Bool ret = A_SendOrder(Enum_sell, Enum_Entry, lots, Open, 0);
        if(ret)
        {
            openOrder = True;
        }
        cancelId=createTimer(10000);
    }
}
If(MarketPosition==1)
```

图 9: 布林通道

接下来考虑平仓策略：引入跟踪止盈和固定止损。


```

//跟踪止盈
If(MarketPosition==1 and Low<=buylasthigh*(1-0.01*hcrate) and buylasthigh>EntryPrice*(1+0.01*6))
{
    Sell(0,Min(Open,buylasthigh*(1-0.01*hcrate)));
    if(BarStatus == 2 && !openOrder)
    { //发平仓单
        //针对当前策略应用的帐户、商品发送委托单
        Bool ret = A_SendOrder(Enum_Sell, Enum_Exit, lots, Open, 0);
        if(ret)
        {
            openOrder = True;
        }
        cancelId=createTimer(10000);
    }
}
If(MarketPosition==-1 and High>=selllastlow*(1+0.01*hcrate) and selllastlow<EntryPrice*(1-0.01*6))
{
    BuyToCover(0,Max(Open,selllastlow*(1+0.01*hcrate)));
    if(BarStatus == 2 && !openOrder)
    { //发平仓单
        //针对当前策略应用的帐户、商品发送委托单
        Bool ret = A_SendOrder(Enum_Buy, Enum_Exit, lots, Open, 0);
        if(ret)
        {
            openOrder = True;
        }
        cancelId=createTimer(10000);
    }
}
}
}

```

图 10: 跟踪止盈

固定止损：根据损失跳数作为平仓依据。

```

//固定止损
If(MarketPosition==1 and Low <= EntryPrice-sp*MinMove*PriceScale)
{
    Sell(0,Max(open,EntryPrice-sp*MinMove*PriceScale));
    if(BarStatus == 2 && !openOrder)
    {//发平仓单
        //针对当前策略应用的帐户、商品发送委托单
        Bool ret = A_SendOrder(Enum_Sell, Enum_Exit, lots, Open, 0);
        if(ret)
        {
            openOrder = True;
        }
        cancelId=createTimer(10000);
    }
}
If(MarketPosition==-1 and High >= EntryPrice+sp*MinMove*PriceScale)
{
    BuyToCover(0,Min(open,EntryPrice+sp*MinMove*PriceScale));
    if(BarStatus == 2 && !openOrder)
    {//发平仓单
        //针对当前策略应用的帐户、商品发送委托单
        Bool ret = A_SendOrder(Enum_Buy, Enum_Exit, lots, Open, 0);
        if(ret)
        {
            openOrder = True;
        }
        cancelId=createTimer(10000);
    }
}
//四周

```

图 11: 固定止损

3 程序化交易策略的实现及应用

3.1 策略代码

Params

```

Numeric money(100);
    //固定市值开仓：单位万
Numeric length(20,10,60,2);    //四周周期参数
Numeric hcrate(8,1,2,1);        //价格回撤幅度
Numeric malength(60,20,120,10); //均线周期参数

```

```
Numeric FastLength(5);           //
    短期指数平均线参数
Numeric SlowLength(20);          //
    长期指数平均线参数
Numeric millsecs(1000);
Numeric Offset(2);               //标准差倍数
Numeric sp(20);                  //止损

Vars
Numeric highline;                //高点连线
Numeric lowline;                 //低点连线
Numeric myprice;                 //委托价格
Numeric lots;                    //委托数量
Series<Numeric> buylasthigh(0,2);
    //买持仓价格峰值
Series<Numeric> selllastlow(0,2);
    //卖持仓价格低谷
Numeric ma60;                    //均线
Numeric StopPrice;
Series<Numeric> AvgValue1;
Series<Numeric> AvgValue2;
Global Integer cancelId;
Global Bool openOrder(False);
Series<Numeric> UpLine;           //上轨
Series<Numeric> DownLine;         //下轨
Series<Numeric> MidLine;          //中间线
Series<Numeric> UpLine_;          //上轨
Series<Numeric> DownLine_;        //下轨
Series<Numeric> Band;
Series<Numeric> minpoint;
Series<Bool> con1;
Series<Bool> con2;

Events
OnInit()
```

```
{
    Range[0:DataCount-1]
    {
        //=====除权换月相关设置=====
        AddDataFlag(Enum_Data_RolloverBackWard());
        //设置后复权
        AddDataFlag(Enum_Data_RolloverRealPrice());
        //设置映射真实价格
        AddDataFlag(Enum_Data_AutoSwapPosition());
        //设置自动换仓
        AddDataFlag(Enum_Data_IgnoreSwapSignalCalc());
        //设置忽略换仓信号计算
        //=====交易相关设置=====
        SetInitCapital(1000000);
        //设置初始资金为100万
        SetBeginBarMaxCount(1);
        //=====交易相关设置=====
        SetInitCapital(1000000);
        //设置初始资金为100万
        SetCommissionRate(BitOr(Enum_Rate_FreeOfExitToday,Enum_Rate_ByFillAmount),5);
        //设置手续费率为成交金额的5%，不收平今。
        SetSlippage(Enum_Rate_PointPerHand,2);
        //设置滑点为2跳/手
    }
}

OnBar(ArrayRef<Integer> indexes)
{
    Range[0:0]
    {
        myprice=Open;//这里使用open,更为精确的是使用委托价格
        lots=IntPart(money*10000/(myprice*contractunit*BigPointValue));
        //计算开仓手数
        ma60=Average(Close[1],malength);
    }
}
```

```

}
Range[1:1]
{
    myprice=Open;//这里使用open,更为精确的是使用委托价格
    lots=IntPart(money*10000/(myprice*contractunit*BigPointValue));
    //计算开仓手数
    MidLine = AverageFC(Close,Length);
    Band = StandardDev(Close,Length,2);
    UpLine = MidLine + Offset * Band;
    DownLine = MidLine - Offset * Band;
    minpoint = MinMove*PriceScale;
    //平滑处理
    UpLine_ = Highest(UpLine,3);
    DownLine_ = Lowest(DownLine,3);
    con1=Close[1]<DownLine_[1] and
        Close>DownLine_ and close>open;
    con2=Close[1]>UpLine_[1] and Close<UpLine_
        and close<open;
    AvgValue1 = AverageFC(Close,FastLength);
    AvgValue2 = AverageFC(Close,SlowLength);
    highline=Highest(High[1],length);
    lowline=Lowest(Low[1],length);
    PlotNumeric("highline",highline);
    PlotNumeric("lowline",lowline);
    PlotNumeric("maline",data0.ma60);
    //跟踪止盈
    If(MarketPosition==1 and
        Low<=buylasthigh*(1-0.01*hcrate) and
        buylasthigh>EntryPrice*(1+0.01*6))
    {
        Sell(0,Min(Open,buylasthigh*(1-0.01*hcrate)));
        if(BarStatus == 2 && !openOrder)
        {//发平仓单

```

```
//针对当前策略应用的帐户、商品发送委托单
Bool ret = A_SendOrder(Enum_Sell,
    Enum_Exit, lots, Open, 0);
if(ret)
{
    openOrder = True;
}
cancelId=createTimer(10000);
}
}
If(MarketPosition== -1 and
    High>=selllastlow*(1+0.01*hcrate) and
    selllastlow<EntryPrice*(1-0.01*6))
{
    BuyToCover(0,Max(Open,selllastlow*(1+0.01*hcrate)));
    if(BarStatus == 2 && !openOrder)
    { //发平仓单
        //针对当前策略应用的帐户、商品发送委托单
        Bool ret = A_SendOrder(Enum_Buy,
            Enum_Exit, lots, Open, 0);
        if(ret)
        {
            openOrder = True;
        }
        cancelId=createTimer(10000);
    }
}
//固定止损
If(MarketPosition==1 and Low <=
    EntryPrice-sp*MinMove*PriceScale)
{
    Sell(0,Max(open,EntryPrice-sp*MinMove*PriceScale));
    if(BarStatus == 2 && !openOrder)
```

```
{//发平仓单
    //针对当前策略应用的帐户、商品发送委托单
    Bool ret = A_SendOrder(Enum_Sell,
        Enum_Exit, lots, Open, 0);
    if(ret)
    {
        openOrder = True;
    }
    cancelId=createTimer(10000);
}
}
If(MarketPosition== -1 and High >=
    EntryPrice+sp*MinMove*PriceScale)
{
    BuyToCover(0,Min(open,EntryPrice+sp*MinMove*PriceScale));
    if(BarStatus == 2 && !openOrder)
    {//发平仓单
        //针对当前策略应用的帐户、商品发送委托单
        Bool ret = A_SendOrder(Enum_Buy,
            Enum_Exit, lots, Open, 0);
        if(ret)
        {
            openOrder = True;
        }
        cancelId=createTimer(10000);
    }
}
//四周
If(MarketPosition<>1 And High>=highline And
    Close[1]>data0.ma60 And AvgValue1[1] >
    AvgValue2[1])
{
    Buy(lots,Max(Open,highline));
}
```

```
    if(BarStatus == 2 && !openOrder)
    { //发开仓单
        //针对当前策略应用的帐户、商品发送委托单
        Bool ret = A_SendOrder(Enum_Buy,
            Enum_Entry, lots, Open, 0);
        if(ret)
        {
            openOrder = True;
        }
        cancelId=createTimer(10000);
    }
}
If(MarketPosition<>-1 And Low<=lowline And
    Close[1]<data0.ma60 And AvgValue1[1] <
    AvgValue2[1])
{
    SellShort(lots,Min(Open,lowline));
    if(BarStatus == 2 && !openOrder)
    { //发开仓单
        //针对当前策略应用的帐户、商品发送委托单
        Bool ret = A_SendOrder(Enum_sell,
            Enum_Entry, lots, Open, 0);
        if(ret)
        {
            openOrder = True;
        }
        cancelId=createTimer(10000);
    }
}
//布林
If(MarketPosition<>1 And High>=highline and
    Close[1]<data0.ma60 And con1[1])
{
```



```
Buy(lots,Max(Open,highline));
if(BarStatus == 2 && !openOrder)
{
    //发开仓单
    //针对当前策略应用的帐户、商品发送委托单
    Bool ret = A_SendOrder(Enum_Buy,
        Enum_Entry, lots, Open, 0);
    if(ret)
    {
        openOrder = True;
    }
    cancelId=createTimer(10000);
}
}
If(MarketPosition<>-1 And Low<=lowline And
    Close[1]>data0.ma60 And con2[1])
{
    SellShort(lots,Min(Open,lowline));
    if(BarStatus == 2 && !openOrder)
    {
        //发开仓单
        //针对当前策略应用的帐户、商品发送委托单
        Bool ret = A_SendOrder(Enum_sell,
            Enum_Entry, lots, Open, 0);
        if(ret)
        {
            openOrder = True;
        }
        cancelId=createTimer(10000);
    }
}
}
If(MarketPosition==1)
{
    If(BarsSinceEntry==0)
        buylasthigh=EntryPrice;
```

```

        Else
            buylasthigh=Max(High,buylasthigh);
    }
    If(MarketPosition== -1)
    {
        If(BarsSinceEntry==0)
            selllastlow=EntryPrice;
        Else
            selllastlow=Min(Low,selllastlow);
    }
}

//定时器更新事件函数，参数id表示定时器的编号，millsecs表示定时间的间隔毫秒值
OnTimer(Integer id,Integer intervalMillsecs)
{
    if(id == cancelId)
    {
        //撤单
        Array<Integer> orderIds;
        //获取未完成的报单数组
        Bool ret = A_GetUnFillOrderIDs(orderIds, "",
            0);
        Integer i;
        For i = 0 To GetArraySize(orderIds) - 1
        {
            Print("accountId:" + A_AccountID(0) +
                ",orderId:" + Text(orderIds[i]));
            //针对当前策略应用的帐户、商品发送撤单指令
            ret = A_DeleteOrder(orderIds[i], 0);
            Print("A_DeleteOrder:" + IIFString(ret,
                "True", "False"));
        }
        StopTimer(cancelId);
        openOrder = False;
    }
}

```

```
    }  
}
```

3.2 实盘操盘及其分析

2023 年 11 月 8 日的交易图中可以看出，总体上是盈利的，而且当日收益为正。

	商品名称	买卖	开平	状态	委托/成交时间	委托价格	成交价	委托/成交数量	对手价	理论价	滑点	滑差%%
委托成交	豆二2312	卖出	开仓	全部成交	2023-11-08 22:58:06	4679	4682	1	4681	4682	0.00	
	豆二2312	买入	开仓	全部成交	2023-11-08 22:56:11	4682	4680	1	4680	4680	0.00	
	豆二2312	卖出	开仓	全部成交	2023-11-08 22:42:14	4682	4684	1	4684	4685	-1.00	
模式	豆二2312	卖出	开仓	全部成交	2023-11-08 22:42:09	4683	4685	1	4685	4685	0.00	
	豆二2312	卖出	开仓	全部成交	2023-11-08 22:42:02	4684	4686	1	4686	4686	0.00	
	豆二2312	卖出	开仓	全部成交	2023-11-08 22:35:29	4682	4684	1	4684	4685	-1.00	
代理	豆二2312	卖出	开仓	全部成交	2023-11-08 22:34:50	4681	4683	1	4683	4684	-1.00	
	豆二2312	买入	开仓	全部成交	2023-11-08 22:33:07	4683	4681	1	4681	4680	-1.00	
	豆二2312	买入	开仓	全部成交	2023-11-08 22:31:38	4685	4683	1	4683	4683	0.00	
	豆二2312	买入	开仓	全部成交	2023-11-08 22:31:29	4685	4683	1	4683	4683	0.00	
	豆二2312	卖出	开仓	全部成交	2023-11-08 22:30:26	4685	4688	1	4687	4688	0.00	
	豆二2312	卖出	开仓	全部成交	2023-11-08 22:30:14	4685	4687	1	4687	4687	0.00	
	豆二2312	买入	开仓	全部成交	2023-11-08 22:29:29	4687	4685	1	4685	4685	0.00	
	豆二2312	买入	开仓	全部成交	2023-11-08 22:29:14	4688	4686	1	4686	4685	-1.00	
	豆二2312	卖出	开仓	全部成交	2023-11-08 22:23:32	4682	4685	1	4684	4685	0.00	
	豆二2312	买入	开仓	全部成交	2023-11-08 22:22:02	4686	4683	1	4684	4684	1.00	

图 12：交易明细图

<input checked="" type="checkbox"/>	资金账户		最新权益	当日盈亏C	净杠杆	总杠杆	风险度	可
<input checked="" type="checkbox"/>	tbf_ahwei		20,164,152	266	-0.005	0.005	0.17%	20,13

图 13：当日收益

3.3 稳定的回测效果

首先选择策略研究，在叠加功能下选择 **wjw** 公式并应用。

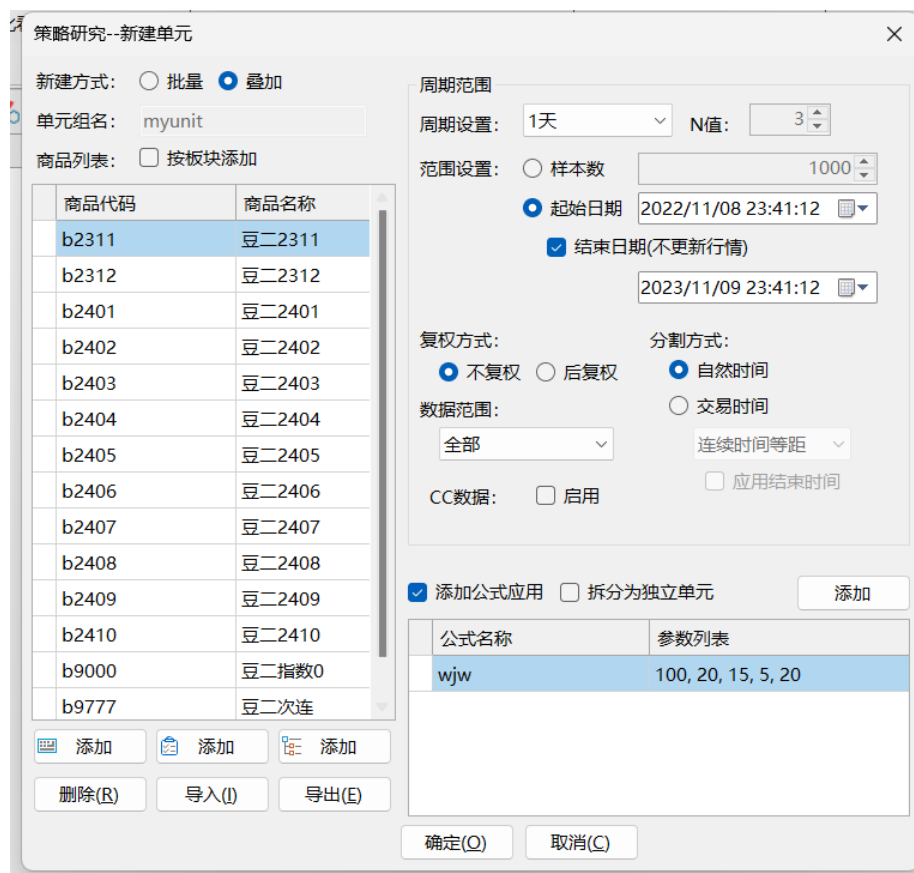


图 14：策略研究——应用 **wjw** 公式

多任务组合报告全为红色，通过对历史数据加载本策略，进行回测效果极佳，收益较为不错。



图 15：策略报告图

从净值曲线图和交易盈亏曲线图可以看出该投资组合的不错的盈利能力。



图 16：净值曲线图



图 17：交易盈亏曲线图