

# T9 宽语言量化交易软件使用说明书

2023 年 11 月

星格量化投资技术研究所（苏州）有限公司

## 目录

第一部分 关于算法交易 .....	3
一、 算法交易是什么 .....	3
(一) 算法交易的概念 .....	3
(二) 算法交易与量化交易的关系 .....	3
(三) 滑点损失会有多大 .....	3
二、 手动下单也可以使用算法交易 .....	4
(一) 算法交易下单 .....	4
(二) 多账号算法交易下单 .....	16
第二部分 期货量化交易 .....	22
一、 关于量化交易 .....	22
(一) 量化交易是什么 .....	22
(二) 量化交易比手动交易的优势 .....	22
(三) 量化的价值不仅仅是自动下单 .....	23
二、 期货量化基本流程 .....	24
(一) 整理思路, 编写模型 .....	24
(二) 模型测试和优化 .....	30
(三) 运行模型自动交易 .....	35
三、 期货量化编写举例 .....	44
(一) 趋势策略+算法下单 .....	44
(二) 独立算法交易策略 .....	72
四、 期货量化回测详解 .....	78
(一) 单合约回测 .....	78
(二) 多合约回测 .....	105
(三) 期货主连链回测 .....	120
(四) 独立算法交易模型回测 .....	125
(五) 期货多因子分析 .....	130

五、期货量化运行详解 .....	132
（一）期货模组实现自动的头寸管理、风险控制 .....	132
（二）独立算法交易模型运行池自动交易 .....	149
第三部分 常见问题 .....	150
一、公式编写常见问题 .....	150
二、回测常见问题 .....	156
三、策略运行常见问题 .....	161
附录：回测报告的各项统计图表说明 .....	164

## 第一部分 关于算法交易

### 一、算法交易是什么

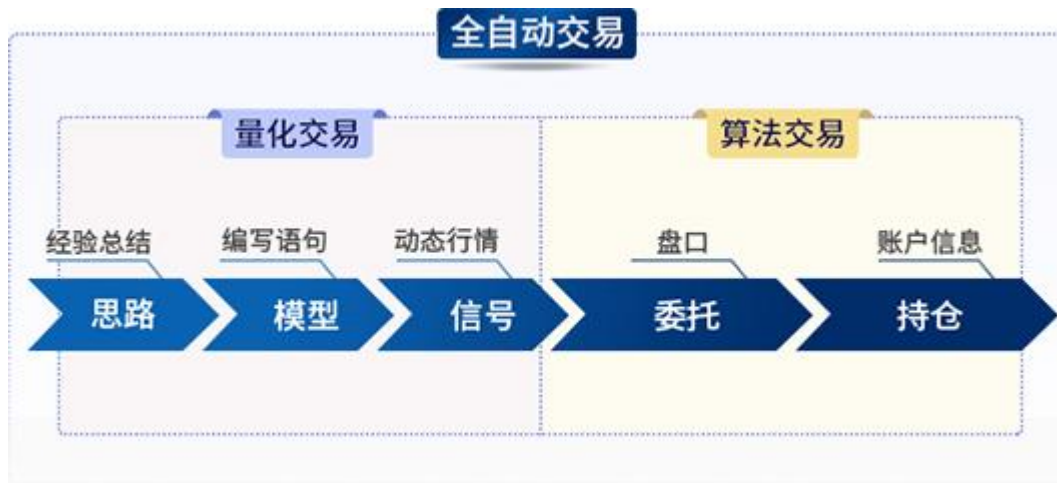
#### (一) 算法交易的概念

算法交易是将交易指令转变成账户持仓的一种优化交易方法。算法交易将计算机得天独厚的速度优势和现代化的智能算法相结合,通过读取五档盘口的委托数据和期货账户的委托状态、资金头寸等信息,自动完成分批、挂单、撤单等一系列交易步骤,最终达到提高交易效率、降低交易成本的目的。

#### (二) 算法交易与量化交易的关系

**量化交易:** 强调的是策略的研究,处于决策层面,要点是策略公式编写、回测报告。

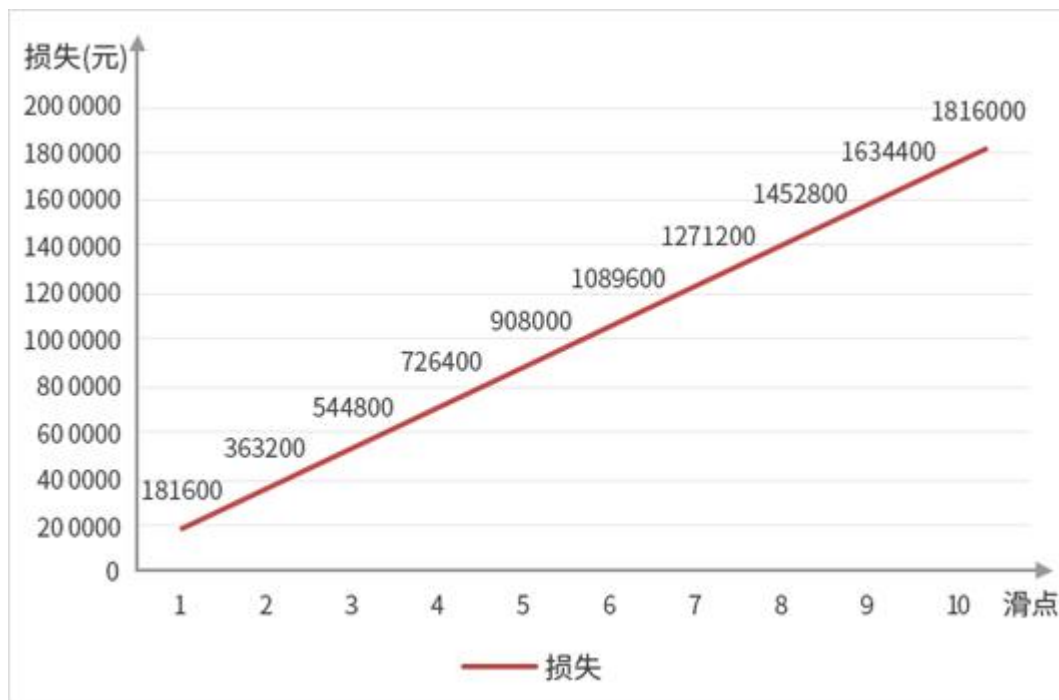
**算法交易:** 强调的是交易过程的精细控制,处于执行层面,编写算法交易公式来提高交易效率,降低交易成本。



#### (三) 滑点损失会有多大

假设我们的资金为 1000 万,每次使用 400 万资金建仓,交易 8 个品种,假设每次损失 1 个最小变动价位,交易频率为 10 天每次,那么我们 200 个交易日的滑点损失就高达 18 万元。而实际交易过程中,产生的滑点往往不止 1 个最小变动价位,而且随着资金规模的增加,滑点的损失还会持续增大。

品种	交易单位	最小变动价位	交易手数	单次交易滑点损失	每日平均交易次数	200个交易日滑点损失
豆粕	10	1	100	1000	0.1	20000
铁矿	100	0.5	50	2500	0.1	50000
螺纹	10	1	100	1000	0.1	20000
橡胶	10	5	30	1500	0.1	30000
白糖	10	1	60	600	0.1	12000
PTA	5	2	150	1500	0.1	30000
IF	300	0.2	3	180	0.1	3600
原油	1000	0.1	8	800	0.1	16000
合计				9080元		181600元



## 二、手动下单也可以使用算法交易

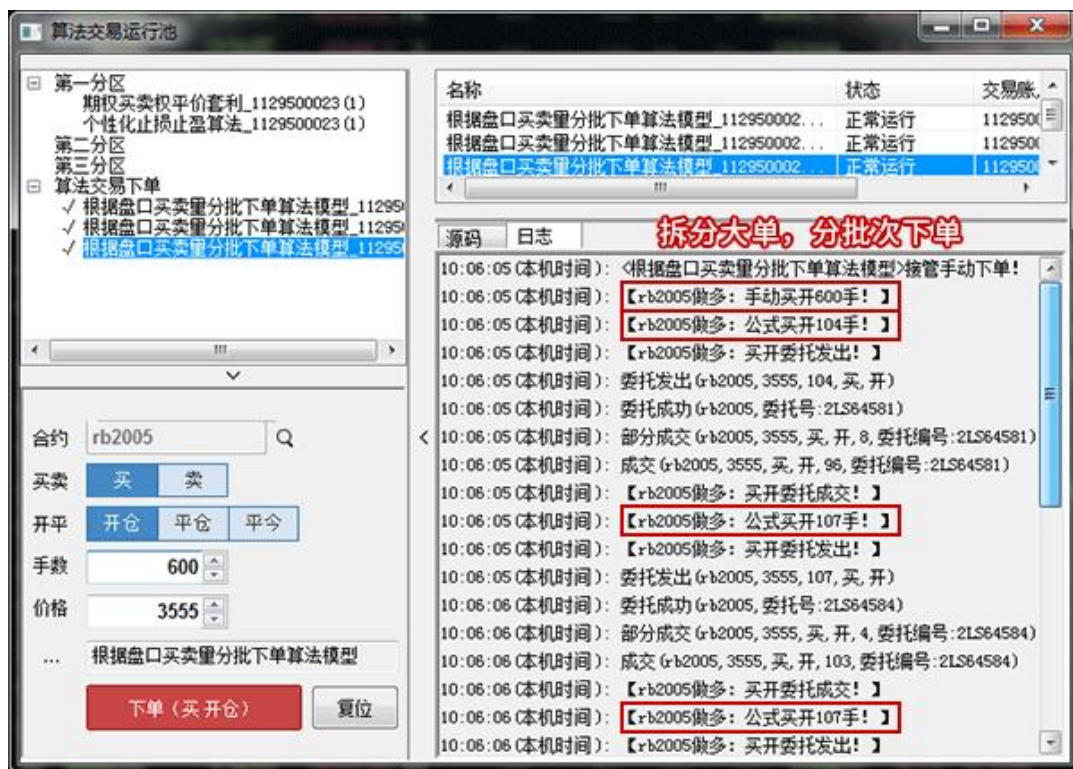
### (一) 算法交易下单

算法交易下单功能支持在手动下单时调用独立算法交易模型控制下单过程，交易员手动选择交易合约、方向和数量，由算法给出下单委托的执行方案。

### 案例：手动下单调用算法交易，自动拆分大单分批委托

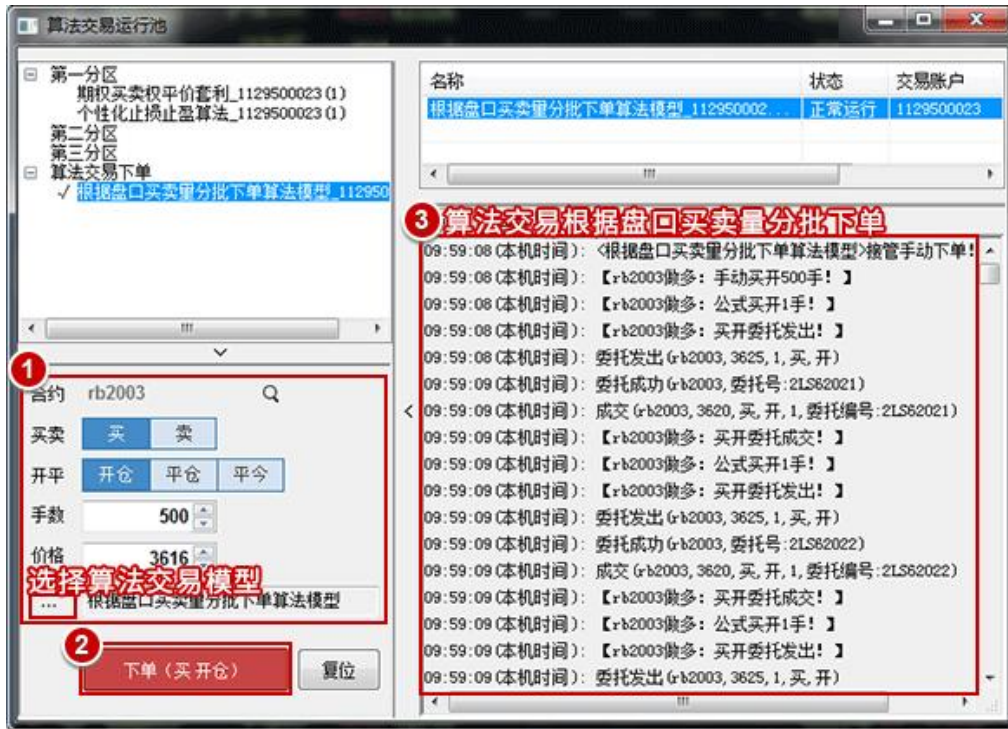
机构投资者往往有下单单量大，唯恐惊扰行情的困扰，我们可以编写一个自动拆分大单分批下单的算法程序，每次手动下单时调用此算法程序接管委托过程分批下单，严格监控下单程序并实时返回交易回报，人机结合更加灵活，配合算法交易精细化下单减少滑点等优势，可以使交易效果大幅提升。

如下图，手动下单 600 手后，由算法交易自动接管买开委托过程，将大单拆分成 N 笔小单，分批买入。将大规模的资金交易行为隐蔽起来，避免一下将市场价格打穿，从而让资金分流、稳定的介入市场。



#### (1) 手动下单调用算法交易的操作方法:

方法 1: 软件右上方菜单【量化】->【算法交易运行池】->【算法交易下单】，调出算法交易下单界面，按下图所示方法实现手动下单调用算法交易:



方法 2：在下单板选择委托价格方式为“算法”，再手动下单，可直接由算法交易池里所指定的模型接管下单，精细化处理，如下图所示。



注：

如果再次下单，会再次加载该模型，不会直接在前一个算法交易模型中继续执行。

思路解析：

/\*

1、取得手动下单的合约、手数、买卖方向、开平方向；自设委托价格。

2、如果为开仓：

(1) 如果为买入开仓：

分批买入，取盘口卖一量\*1/2 与剩余下单手数比较，取数量较小值根据盘口卖一价买入，第一批成交后，再委托第二批，直至完全成交后退出算法。

(2) 如果为卖出开仓：

分批卖出，取盘口买一量\*1/2 与剩余下单手数比较，取数量较小值根据盘口买一价卖出，第一批成交后，再委托第二批，直至完全成交后退出算法。

3、如果为平仓：

(1) 如果为买入平仓：

分批处理同买入开仓。

(2) 如果为卖出平仓：

分批处理同卖出开仓。

\*/

### 源码示例：

Vars

```
Global_String AID; //账号
Global_String COD; //合约编码
Numeric P; //数值比例
Numeric SH; //上海合约标志
Numeric BSF; //买卖方向
Numeric EEF; //开平方向
Numeric BIDP,ASKP; //买卖一价
Numeric BIDV,ASKV; //买卖一量
Numeric BRP0,BRP1,BRP; //多头可用持仓
Numeric SRP0,SRP1,SRP; //空头可用持仓
Numeric OPFLG; //模型处理标志
Global_Numeric VLM; //交易手数
Global_Numeric KPFLG; //委托处理标志
Global_Numeric BKDFLG; //买开处理标志
Global_Numeric SKDFLG; //卖开处理标志
Global_Numeric BPDFLG; //买平处理标志
Global_Numeric SPDFLG; //卖平处理标志
Global_Numeric BKID,SKID; //开仓委托
Global_Numeric BKFLG,SKFLG; //开仓标志
Global_Numeric BKM,SKM; //开仓委托手数
Global_Numeric BKP,SKP; //开仓委托价格
Global_Numeric BPID0,BPID1,BPID; //买平委托
Global_Numeric SPID0,SPID1,SPID; //卖平委托
```



Global\_Numeric BPFLG0,BPFLG1,BPFLG; //买平标志

Global\_Numeric SPFLG0,SPFLG1,SPFLG; //卖平标志

Global\_Numeric BPM0,BPM1,BPM; //买平委托手数

Global\_Numeric SPM0,SPM1,SPM; //卖平委托手数

Global\_Numeric BPP0,BPP1,BPP; //买平委托价格

Global\_Numeric SPP0,SPP1,SPP; //卖平委托价格

Begin

//-----委托获取-----//

If(1) //委托获取

{

    If(KPFLG == 0) //如果未开启委托处理

    {

        AID = M\_GetTradeAccount(0); //账号

        COD = M\_GetTradeContract(); //合约编码

        BSF = M\_GetTradeBuySell(); //买卖方向

        EEF = M\_GetTradeEntryExit(); //开平方向

        VLM = M\_GetTradeVol(); //交易手数

        If(VLM > 0 && COD.A\_IsExchangeOpen() == 1) //如果交易手数大于0，且当前状态是

开盘

        {

            If(BSF == Enum\_Buy && EEF == Enum\_Entry) //如果是买开方向

            {

                Commentary("【" + COD + "做多：手动买开" + Text(VLM) + "手！】");

                BKDFLG = 1; //开启买开处理

            }

            Else If(BSF == Enum\_Sell && EEF == Enum\_Exit) //如果是卖平方向

            {

                Commentary("【" + COD + "平多：手动卖平" + Text(VLM) + "手！】");

                SPDFLG = 1; //开启卖平处理

            }

            Else If(BSF == Enum\_Sell && EEF == Enum\_ExitToday) //如果是卖平今方向

            {

                Commentary("【" + COD + "平今多：手动卖平" + Text(VLM) + "手！】");

                SPDFLG = 2; //开启卖平今仓处理

            }

            Else If(BSF == Enum\_Sell && EEF == Enum\_Entry) //如果是卖开方向

            {

                Commentary("【" + COD + "做空：手动卖开" + Text(VLM) + "手！】");

                SKDFLG = 1; //开启卖开处理

            }

            Else If(BSF == Enum\_Buy && EEF == Enum\_Exit) //如果是买平方向

            {

                Commentary("【" + COD + "平空：手动买平" + Text(VLM) + "手！】");

```

        BPDFLG = 1; //开启买平处理
    }
    Else If(BSF == Enum_Buy && EEF == Enum_ExitToday) //如果是买平今方向
    {
        Commentary("【" + COD + "平今空：手动买平" + Text(VLM) + "手！】");
        BPDFLG = 2; //开启买平今仓处理
    }
    KPFLG = 1; //开启委托处理
}
}
If(KPFLG == 1 && COD.A_IsExchangeOpen() == 1) //如果已开启委托处理，且当前状态是
开盘
{
    OPFLG = 1; //开启模型处理
}
}
//-----变量赋值-----//
If(OPFLG == 1) //变量赋值
{
    P = 0.5; //数值比例
    BIDP = COD.Price("Bid1"); //买一价
    ASKP = COD.Price("Ask1"); //卖一价
    BIDV = COD.Price("BidVol1"); //买一量
    ASKV = COD.Price("AskVol1"); //卖一量
    SH = COD.A_IsSHCode(); //上海合约标志
    BRP = COD.A_BuyRemainPosition(); //多头可用持仓
    SRP = COD.A_SellRemainPosition(); //空头可用持仓
    If(SH == 1) //如果是上期所合约
    {
        BRP0 = COD.A_TodayBuyRemainPosition(); //多头今仓可用持仓
        SRP0 = COD.A_TodaySellRemainPosition(); //空头今仓可用持仓
        BRP1 = BRP - BRP0; //多头老仓可用持仓
        SRP1 = SRP - SRP0; //空头老仓可用持仓
    }
}
//-----成交判断-----//
If(OPFLG == 1) //成交判断
{
    If(BKFLG == 1) //如果有买开委托
    {
        If(F_OrderStatus(BKID) == Enum_Filled) //如果买开委托成交
        {
            Commentary("【" + COD + "做多：买开委托成交！】");
        }
    }
}

```

```
VLM = VLM - BKM; //交易手数
BKFLG = 0; //买开标志归 0
}
}
If(SH == 1) //如果是上期所合约
{
    If(SPFLG0 == 1) //如果有卖平今仓委托
    {
        If(F_OrderStatus(SPID0) == Enum_Filled) //如果卖平今仓委托成交
        {
            Commentary("【" + COD + "平多：卖平今仓委托成交！】");
            VLM = VLM - SPM0; //交易手数
            SPFLG0 = 0; //卖平今仓标志归 0
        }
    }
    If(SPFLG1 == 1) //如果有卖平老仓委托
    {
        If(F_OrderStatus(SPID1) == Enum_Filled) //如果卖平老仓委托成交
        {
            Commentary("【" + COD + "平多：卖平老仓委托成交！】");
            VLM = VLM - SPM1; //交易手数
            SPFLG1 = 0; //卖平老仓标志归 0
        }
    }
}
Else //如果非上期所合约
{
    If(SPFLG == 1) //如果有卖平委托
    {
        If(F_OrderStatus(SPID) == Enum_Filled) //如果卖平委托成交
        {
            Commentary("【" + COD + "平多：卖平委托成交！】");
            VLM = VLM - SPM; //交易手数
            SPFLG = 0; //卖平标志归 0
        }
    }
}
If(SKFLG == 1) //如果有卖开委托
{
    If(F_OrderStatus(SKID) == Enum_Filled) //如果卖开委托成交
    {
        Commentary("【" + COD + "做空：卖开委托成交！】");
        VLM = VLM - SKM; //交易手数
```

```

        SKFLG = 0; //卖开标志归 0
    }
}
If(SH == 1) //如果是上期所合约
{
    If(BPFLG0 == 1) //如果有买平今仓委托
    {
        If(F_OrderStatus(BPID0) == Enum_Filled) //如果买平今仓委托成交
        {
            Commentary("【" + COD + "平空：买平今仓委托成交！】");
            VLM = VLM - BPM0; //交易手数
            BPFLG0 = 0; //买平今仓标志归 0
        }
    }
    If(BPFLG1 == 1) //如果有买平老仓委托
    {
        If(F_OrderStatus(BPID1) == Enum_Filled) //如果买平老仓委托成交
        {
            Commentary("【" + COD + "平空：买平老仓委托成交！】");
            VLM = VLM - BPM1; //交易手数
            BPFLG1 = 0; //买平老仓标志归 0
        }
    }
}
Else //如果非上期所合约
{
    If(BPFLG == 1) //如果有买平委托
    {
        If(F_OrderStatus(BPID) == Enum_Filled) //如果买平委托成交
        {
            Commentary("【" + COD + "平空：买平委托成交！】");
            VLM = VLM - BPM; //交易手数
            BPFLG = 0; //买平标志归 0
        }
    }
}
}
//-----委托处理-----//
If(OPFLG == 1) //委托处理
{
    If(BKDFLG == 1) //如果已开启买开处理
    {
        If(BKFLG == 0) //如果没有买开委托

```

```

{
    BKM = Min(Ceiling(P * ASKV,1),VLM); //买开委托手数
    If(BKM > 0) //如果买开委托手数大于 0
    {
        BKP = ASKP; //买开委托价格
        Commentary("【" + COD + "做多： 公式买开" + Text(BKM) + "手! 】");
        Commentary("【" + COD + "做多： 买开委托发出! 】");
        BKID = COD.A_SendOrder(Enum_Buy,Enum_Entry,BKM,BKP,AID); //发出买开
委托
        BKFLG = 1; //已发出买开委托
    }
    Else If(BKM == 0) //如果买开委托手数等于 0
    {
        Commentary("【" + COD + "做多： 买开委托完成! 】");
        Exit(); //退出公式
    }
}
}
If(SPDFLG > 0) //如果已开启卖平处理
{
    If(SPFLG0 == 0 && SPFLG1 == 0 && SPFLG == 0) //如果没有卖平委托
    {
        SPM = Min(Ceiling(P * BIDV,1),VLM); //卖平委托手数
        If(BRP > 0 && SPM > 0) //如果有多头可用持仓， 且卖平委托手数大于 0
        {
            If(SH == 1) //如果是上期所合约
            {
                If(SPDFLG == 2) //如果是卖平今仓处理
                {
                    If(BRP0 > 0) //如果有多头今仓可用持仓
                    {
                        SPM0 = Min(BRP0,SPM); //卖平今仓委托手数
                        SPP0 = BIDP; //卖平今仓委托价格
                        Commentary("【" + COD + "平多： 公式卖平今仓" + Text(SPM0) + "
手! 】");
                        Commentary("【" + COD + "平多： 卖平今仓委托发出! 】");
                        SPID0
                        COD.A_SendOrder(Enum_Sell,Enum_ExitToday,SPM0,SPP0,AID); //发出卖平今仓委托
                        SPFLG0 = 1; //已发出卖平今仓委托
                    }
                    Else If(BRP0 == 0) //如果没有多头今仓可用持仓
                    {
                        Commentary("【" + COD + "平多： 卖平委托完成! 】");
                    }
                }
            }
        }
    }
}

```

```

        Exit(); //退出公式
    }
}
Else If(SPDFLG == 1) //如果是卖平处理
{
    If(BRP1 > 0) //如果有多头老仓可用持仓
    {
        SPM1 = Min(BRP1,SPM); //卖平老仓委托手数
        SPP1 = BIDP; //卖平老仓委托价格
        Commentary("【" + COD + "平多：公式卖平老仓" + Text(SPM1) + "
手!】");
        Commentary("【" + COD + "平多：卖平老仓委托发出!】");
        SPID1 = COD.A_SendOrder(Enum_Sell,Enum_Exit,SPM1,SPP1,AID);
//发出卖平老仓委托
        SPFLG1 = 1; //已发出卖平老仓委托
    }
    Else If(BRP1 == 0) //如果没有多头老仓可用持仓
    {
        Commentary("【" + COD + "平多：卖平委托完成!】");
        Exit(); //退出公式
    }
}
}
Else //如果非上期所合约
{
    SPM = Min(BRP,SPM); //卖平委托手数
    SPP = BIDP; //卖平委托价格
    Commentary("【" + COD + "平多：公式卖平" + Text(SPM) + "手!】");
    Commentary("【" + COD + "平多：卖平委托发出!】");
    SPID = COD.A_SendOrder(Enum_Sell,Enum_Exit,SPM,SPP,AID); //发出卖平
委托
    SPFLG = 1; //已发出卖平委托
}
}
Else If(BRP == 0 || SPM == 0) //如果没有多头可用持仓，或卖平委托手数等于 0
{
    Commentary("【" + COD + "平多：卖平委托完成!】");
    Exit(); //退出公式
}
}
}
If(SKDFLG == 1) //如果已开启卖开处理
{

```

```

If(SKFLG == 0) //如果没有卖开委托
{
    SKM = Min(Ceiling(P * BIDV,1),VLM); //卖开委托手数
    If(SKM > 0) //如果卖开委托手数大于 0
    {
        SKP = BIDP; //卖开委托价格
        Commentary("【" + COD + "做空： 公式卖开" + Text(SKM) + "手! 】");
        Commentary("【" + COD + "做空： 卖开委托发出! 】");
        SKID = COD.A_SendOrder(Enum_Sell,Enum_Entry,SKM,SKP,AID); //发出卖开
委托

        SKFLG = 1; //已发出卖开委托
    }
    Else If(SKM == 0) //如果卖开委托手数等于 0
    {
        Commentary("【" + COD + "做空： 卖开委托完成! 】");
        Exit(); //退出公式
    }
}
}

If(BPDFLG > 0) //如果已开启买平处理
{
    If(BPFLG0 == 0 && BPFLG1 == 0 && BPFLG == 0) //如果没有买平委托
    {
        BPM = Min(Ceiling(P * ASKV,1),VLM); //买平委托手数
        If(SRP > 0 && BPM > 0) //如果有空头可用持仓， 且买平委托手数大于 0
        {
            If(SH == 1) //如果是上期所合约
            {
                If(BPDFLG == 2) //如果是买平今仓处理
                {
                    If(SRP0 > 0) //如果有空头今仓可用持仓
                    {
                        BPM0 = Min(SRP0,BPM); //买平今仓委托手数
                        BPP0 = ASKP; //买平今仓委托价格
                        Commentary("【" + COD + "平空： 公式买平今仓" + Text(BPM0) + "
手! 】");

                        Commentary("【" + COD + "平空： 买平今仓委托发出! 】");
                        BPID0
                        =
COD.A_SendOrder(Enum_Buy,Enum_ExitToday,BPM0,BPP0,AID); //发出买平今仓委托
                        BPFLG0 = 1; //已发出买平今仓委托
                    }
                }
                Else If(SRP0 == 0) //如果没有空头今仓可用持仓
                {

```

```

        Commentary("【" + COD + "平空：买平委托完成！】");
        Exit(); //退出公式
    }
}
Else If(BPDLFG == 1) //如果是买平处理
{
    If(SRP1 > 0) //如果有空头老仓可用持仓
    {
        BPM1 = Min(SRP1,BPM); //买平老仓委托手数
        BPP1 = ASKP; //买平老仓委托价格
        Commentary("【" + COD + "平空：公式买平老仓" + Text(BPM1) + "
手！】");

        Commentary("【" + COD + "平空：买平老仓委托发出！】");
        BPID1 = COD.A_SendOrder(Enum_Buy,Enum_Exit,BPM1,BPP1,AID);
//发出买平老仓委托

        BPFLG1 = 1; //已发出买平老仓委托
    }
    Else If(SRP1 == 0) //如果没有空头老仓可用持仓
    {
        Commentary("【" + COD + "平空：买平委托完成！】");
        Exit(); //退出公式
    }
}
}
Else //如果非上期所合约
{
    BPM = Min(SRP,BPM); //买平委托手数
    BPP = ASKP; //买平委托价格
    Commentary("【" + COD + "平空：公式买平" + Text(BPM) + "手！】");
    Commentary("【" + COD + "平空：买平委托发出！】");
    BPID = COD.A_SendOrder(Enum_Buy,Enum_Exit,BPM,BPP,AID); //发出买
平委托

    BPFLG = 1; //已发出买平委托
}
}
Else If(SRP == 0 || BPM == 0) //如果没有空头可用持仓，或买平委托手数等于 0
{
    Commentary("【" + COD + "平空：买平委托完成！】");
    Exit(); //退出公式
}
}
}
}
}

```



End

## （二）多账号算法交易下单

算法交易也支持多账号智能分批，既可以对多个账号进行分组，设定不同的委托手数倍率，也可以根据对手盘情况实时计算每一批的总手数，实现多账号智能分批、统一管理。

例如：

我们三个交易账号需要同时开仓，三个交易账号的委托手数倍率分别为 1、1.5、2.5。我们可在多账号下单界面中勾选要参与交易的账号，并设定好下单倍率。

期	账号	组				
外 股 权	<input checked="" type="checkbox"/>	账号	状态	<input checked="" type="checkbox"/> 倍率	权益	可用资金
	<input checked="" type="checkbox"/>	1135040753	已登录	1.00	213038230	213000061
	<input checked="" type="checkbox"/>	1103101446	已登录	1.50	2039723276	1996580111
	<input checked="" type="checkbox"/>	1105100404	已登录	2.50	2014227939	2014227939
					4266989445	4223808111

我们可以一键对这三个账号一起进行智能分批下单。

首先，要求每批的总手数不超过盘口对手盘量的 20%。首批三个账户按照 1 : 1.5 : 2.5 的比例进行下单。

其次，任何一个账号第一批成交以后（三个账号不一定同时都成交），第二批委托数量为对手盘量的 20%乘以该账号的比例，直到全部成交。

最后，任何一个账号如 3 秒未全部成交，则执行追价策略确保成交。

这样既保证了总手数不对趋势造成影响，也同时完成了三个账号的分批建仓。

代码如下：

Vars

Global\_StringArray AID; //账号

Global\_String COD; //合约编码

Numeric X; //循环变量

Numeric T; //时间间隔

Numeric P; //数值比例

Numeric BSF; //买卖方向

Numeric EEF; //开平方向

Numeric VLM; //交易手数

```
Numeric NOW; //当前时间
Numeric ASKP; //卖一价
Numeric ASKV; //卖一量
Numeric ODFLG; //挂单判断标志
Numeric OPFLG; //模型处理标志
Global_Numeric ADN; //账号数量
Global_Numeric TP; //下单总倍数
Global_Numeric KPFLG; //委托处理标志
Global_Numeric BKDFLG; //买开处理标志
Global_NumericArray NP; //下单倍数
Global_NumericArray BKID; //买开委托
Global_NumericArray BKFLG; //买开标志
Global_NumericArray BKDEL; //买开撤单标志
Global_NumericArray BKSM; //买开总手数
Global_NumericArray BKM; //买开委托手数
Global_NumericArray BKP; //买开委托价格
Global_NumericArray BKT; //买开委托时间

Begin
//-----委托获取-----//
If(1) //委托获取
{
    If(KPFLG == 0) //如果未开启委托处理
    {
        ADN = M_GetTradeAccountNum; //账号数量
        COD = M_GetTradeContract(); //合约编码
        BSF = M_GetTradeBuySell(); //买卖方向
        EEF = M_GetTradeEntryExit(); //开平方向
        VLM = M_GetTradeVol(); //交易手数
        For X = 0 To ADN - 1 //遍历账号
        {
            AID[X] = M_GetTradeAccount(X); //账号
```

```

NP[X] = M_GetTradeOrderRatio(X); //下单倍数
BKSM[X] = NP[X] * VLM; //买开总手数
TP = TP + NP[X]; //下单总倍数
}
If(VLM > 0 && COD.A_IsExchangeOpen() == 1) //如果交易手数大于 0, 且当前状态是开盘
{
    If(BSF == Enum_Buy && EEF == Enum_Entry) //如果是买开方向
    {
        Commentary("【" + COD + "做多：手动买开" + Text(VLM) + "手! 】");
        BKDFLG = 1; //开启买开处理
    }
    KPFLG = 1; //开启委托处理
}
}
If(KPFLG == 1 && COD.A_IsExchangeOpen() == 1) //如果已开启委托处理, 且当前状态是开
盘
{
    OPFLG = 1; //开启模型处理
}
}
//-----变量赋值-----//
If(OPFLG == 1) //变量赋值
{
    T = 3; //时间间隔
    P = 0.2; //数值比例
    NOW = CurrentTime(); //当前时间
    ASKP = COD.Price("Ask1"); //卖一价
    ASKV = COD.Price("AskVol1"); //卖一量
}
//-----成交判断-----//
If(OPFLG == 1) //成交判断
{

```

```

For X = 0 To ADN - 1 //遍历账号
{
  If(BKFLG[X] == 1) //如果有买开委托
  {
    If(F_OrderStatus(BKID[X]) == Enum_Filled) //如果买开委托成交
    {
      Commentary("【" + COD + "做多：买开委托成交！" + AID[X] + "】");
      BKFLG[X] = 0; //买开标志归 0
      BKDEL[X] = 0; //买开撤单标志归 0
    }
    Else If(F_OrderStatus(BKID[X]) == Enum_Canceled) //如果买开委托已撤
    {
      Commentary("【" + COD + "做多：买开委托已撤！" + AID[X] + "】");
      If(F_OrderFilledLot(BKID[X]) > 0) //如果买开委托部分成交
      {
        BKM[X] = BKM[X] - F_OrderFilledLot(BKID[X]); //买开委托手数
      }
      If(BKM[X] > 0) //如果买开委托手数大于 0
      {
        BKP[X] = ASKP; //买开委托价格
        Commentary("【" + COD + "做多：买开委托追价！" + AID[X] + "】");
        BKID[X]
        COD.A_SendOrder(Enum_Buy,Enum_Entry,BKM[X],BKP[X],AID[X]); //发出买开委托
        BKT[X] = NOW; //买开委托时间
      }
      BKDEL[X] = 0; //买开撤单标志归 0
    }
    Else If(F_OrderStatus(BKID[X]) == Enum_Declared || F_OrderStatus(BKID[X]) ==
Enum_FillPart) //如果买开委托挂单
    {
      If(BKDEL[X] == 0) //如果未撤单
      {

```

```

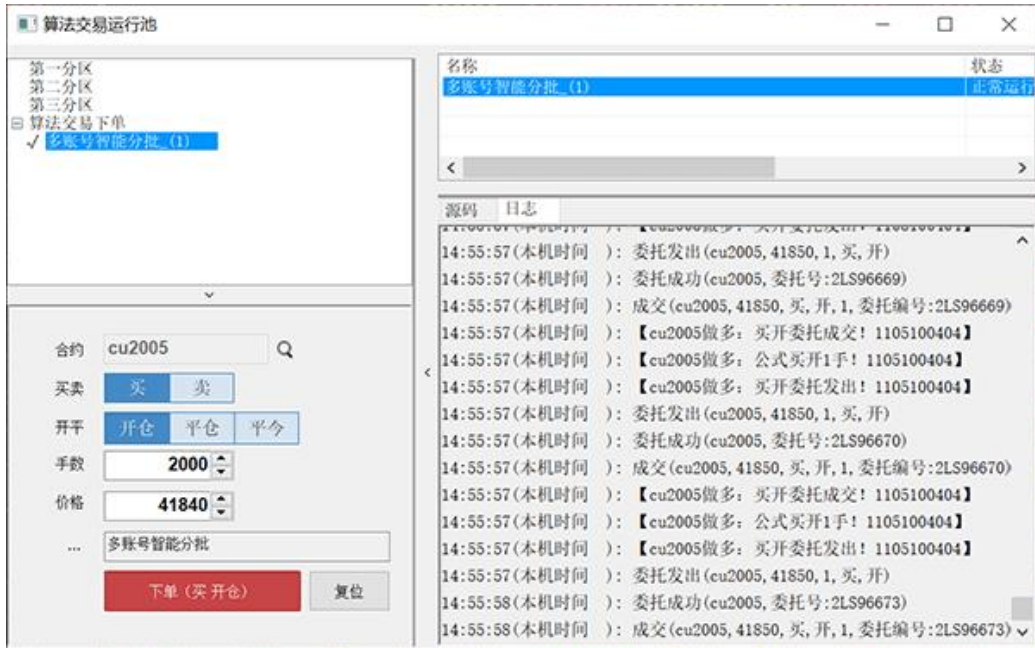
If(TimeDiff(BKT[X],NOW) >= T) //如果时间间隔 T 秒
{
    Commentary("【" + COD + "做多：买开委托撤单！" + AID[X] + "】");
    COD.F_DeleteOrder(F_OrderContractNo(BKID[X]),AID[X]); //撤掉买开委托
挂单
    BKDEL[X] = 1; //已发出撤掉买开委托挂单
}
}
}
}
}
}
}
}
ODFLG = 1; //挂单判断标志初始化
For X = 0 To ADN - 1 //遍历账号
{
    If(BKFLG[X] == 1 || BKSM[X] > 0) //如果有买开委托，或有剩余买开手数
    {
        ODFLG = 2; //存在买开委托挂单
    }
}
If(ODFLG == 1) //如果没有买开委托
{
    Commentary("【" + COD + "做多：买开委托完成！】");
    BKDFLG = 0; //买开处理标志归 0
    Exit(); //退出公式
}
}
//-----委托处理-----//
If(OPFLG == 1) //委托处理
{
    If(BKDFLG == 1) //如果已开启买开处理
    {
        For X = 0 To ADN - 1 //遍历账号

```

```

{
  If(BKFLG[X] == 0) //如果没有买开委托
  {
    If(TP > 0) //如果已取得下单总倍数
    {
      BKM[X] = Min(Floor((NP[X] / TP) * P * ASKV),BKSM[X]); //买开委托手数
    }
    If(BKM[X] > 0) //如果买开委托手数大于 0
    {
      BKP[X] = ASKP; //买开委托价格
      Commentary("【" + COD + "做多：公式买开" + Text(BKM[X]) + "手!" + AID[X]
+ "】");
      Commentary("【" + COD + "做多：买开委托发出!" + AID[X] + "】");
      BKID[X] =
COD.A_SendOrder(Enum_Buy,Enum_Entry,BKM[X],BKP[X],AID[X]); //发出买开委托
      BKSM[X] = BKSM[X] - BKM[X]; //买开总手数自减买开处理手数
      BKT[X] = NOW; //买开委托时间
      BKFLG[X] = 1; //已发出买开委托
    }
  }
}
}
}
}
}
End

```



## 第二部分 期货量化交易

### 一、关于量化交易

#### (一) 量化交易是什么

##### 量化交易的概念

量化交易是借助现代统计学和数学建模的方法，使用海量的历史数据进行回测研究，评估策略的优劣，找到盈利与风险的平衡点，并依靠计算机配置多元化的投资组合，以实现降低系统风险并克服人性弱点的作用，使投资决策更科学，更理性。

#### (二) 量化交易比手动交易的优势

##### 1、策略生产高效迭代，多维度认知策略

量化交易是一个研究的过程，投资者可借助量化平台提供的丰富数据和回测系统验证思路的可行性。以前动辄几个月甚至几年的实盘验证，在量化仿真交易系统下几天就可以轻松搞定，高效生产策略。分析报告多维度解读策略优缺点，不需要真金白银和大量的时间就可以全面考察策略的可行性，同时也避免了实盘量化测试不足可能导致的事故和风险。

## 2、资金管理，组合交易分散风险

手动交易管理一个策略、一个合约、一个账户难度不算大，但管理一组策略、一篮子合约、一组账户交叉组合运行难免分身乏术。量化多线程运算，可精确统计各个策略和账户的头寸和资金调度，根据行情智能调仓，保障策略有条不紊的运行。多模型多合约多账户组合运行，高强度分散风险，提高资金利用率，避免了手动下单的不理性交易，统筹账户进退有度。

## 3、减小市场摩擦，降低冲击成本

冲击成本是交易中不容忽视的成本项，尤其是大资金用户，大规模集中下单不仅会迅速拉升行情，产生冲击成本，而且集中产生的巨量还会暴露交易行踪，造成市场跟风。量化高频下单可极大程度减小市场摩擦，拆细大单，潜入市场，降低冲击成本，锁定既定收益。

## 4、捕捉市场机会，节约人力成本

大数据时代，数据挖掘和云计算主导着交易市场的潮流风向。相对于手动交易，量化交易的运算能力占据了得天独厚的优势，量化策略源源不断的从市场中挖掘信息，持续发现市场中的投资机会，在瞬息万变的交易市场里面，第一时间进出场，降低交易成本，让交易者积累更多的财富，节约人力成本，将交易者从电脑前解放出来。

# （三）量化的价值不仅仅是自动下单

## 量化也是研究的平台

在近几年的交易中，有这样一种现象正悄然形成，一些年轻的投资者虽然进入期货市场的时间不长，但却一样可以拥有良好的收益情况并且将回撤也控制的很到位。从前成功的交易人几乎都是交易经验丰富的老期货，为什么近几年却能够涌现出年轻的优秀交易员呢？这其中，量化扮演了重要的角色。

计算机的最大特点是高效率的数据运算和高智能的数据分析，1分钟周期一天有225根K线数据，按照每年250个交易日计算，如果想要分析出1分钟周期一年的均线走势，我们需要计算至少5.6万根K线数据，这个统计由人来完成可能需要几天，但计算机只需要几秒钟。我们可利用量化语言将想要统计的数据告诉计算机，由计算机帮我们完成计算，例如挖掘历史行情研究K线震动幅度和行情涨跌的规律、探究开盘跳空幅度和当日行情涨跌幅之间的规律等等。



当我们觉得自己似乎发现了一些规律希望验证时，量化平台自带的效果测试功能可帮助我们在历史数据上验证规律是否有效，策略是否可行。我们还可通过量化平台自带的策略优化功能对思路进行完善，大大缩短了投资者确立自己交易策略的时间。

## 二、期货量化基本流程

提示：应期货监管部门要求，交易者必须确认上一交易日结算单后，才能进行当日交易。所以对于全自动量化交易者必须到 T9 软件交易界面->参数设置->交易安全->勾选“自动确认结算单”，否则将影响你的交易无法下单。手动交易者登录交易界面后可手动每日确认结算单，无需勾选“自动确认结算单”项。

### （一）整理思路，编写模型

#### 1、宽语言模型构建方法

T9 软件使用宽语言编程，类 C 语言语法，支持各类复杂策略的开发，支持个性化的信号管理和精细化下单方案。公式属于编译型，编译后直接运行可执行文件，大大提高了公式的执行效率。

##### （1）基于 K 线数据的公式结构

“基于 K 线数据”类型，指根据 K 线图表数据构建趋势策略，写入算法交易接管信号对下单精细化管理的模型。

公式由定义数据区、定义环境设置、设置跨合约跨周期引用关键字、定义参数、定义变量和主程序构成，如下图所示。除主程序外，其余部分可针对具体思路缺省。



### (2) 独立算法交易模型的公式结构

“独立的算法交易模型”类型，指不调用 K 线图表数据，仅以盘口和账户数据构建的算法模型。

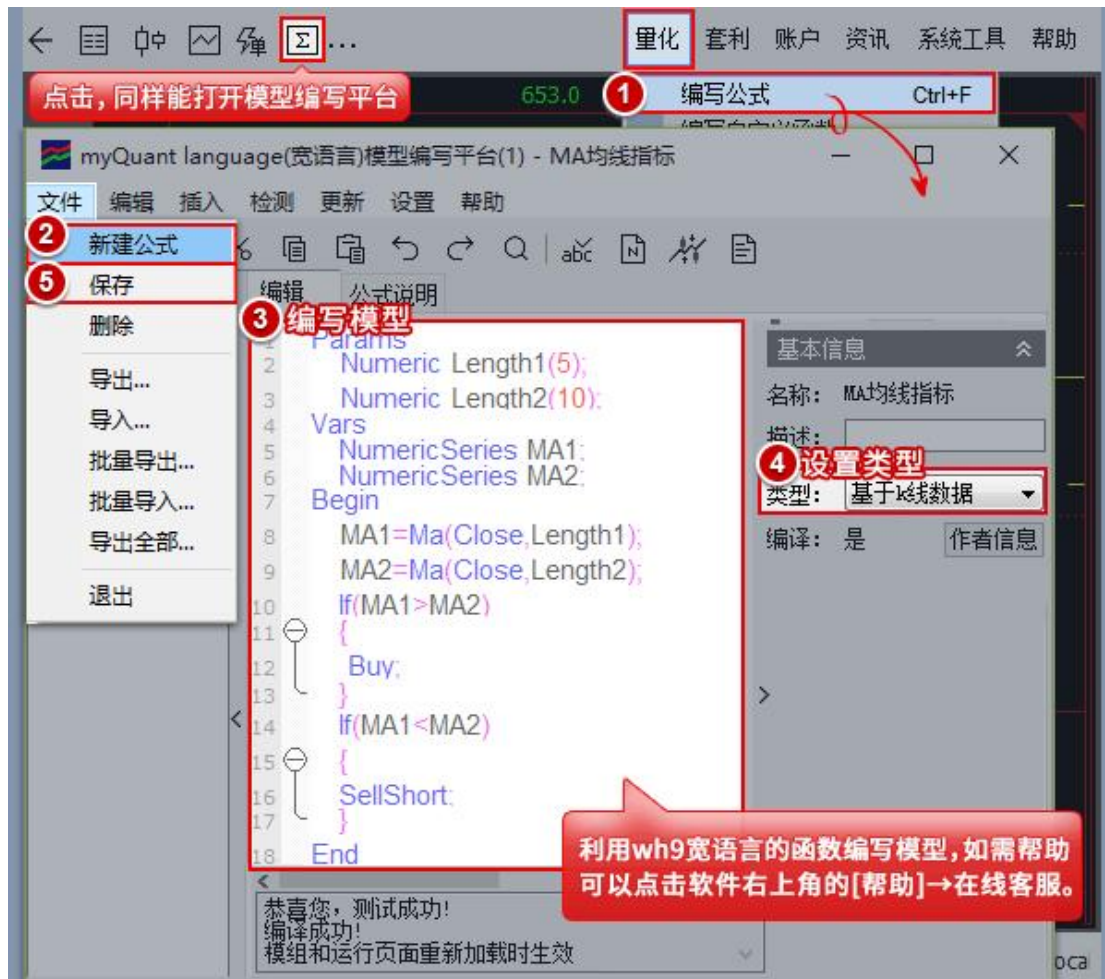
公式由调用数据区 Data、定义变量 Vars 和主函数 Begin—End 三部分组成，如下图所示：

```

1  Data
2  data1:"m1805";           //定义合约           定义数据区
3
4  Vars
5  Numeric Lots;           //定义数值型变量
6  Var_TickData data2;     //定义数据区变量
7  Global_Numeric STR0;    //定义全局变量           定义变量
8  Global_Numeric STR1;    //定义全局变量
9  Global_Numeric BKID;    //定义委托索引
10 .....
11
12 Begin
13 data2 = Def_TickData("m1805",1,4); // 保存最近N笔的tick数据
14 MinP = data1.Price("MinPrice");//获取合约的最小变动价位
15 .....           主函数
16
17 End
    
```

### (3) 编写公式：

如下图，点击软件右上方角菜单【量化】->【编写公式】进行编写。



#### (4) 编写规则:

- ① 宽语言编程语法: [http://www.wenhua.com.cn/guide/myQuant\\_zjyfs2.htm](http://www.wenhua.com.cn/guide/myQuant_zjyfs2.htm)
- ② 常见编写和机制说明: [https://www.wenhua.com.cn/guide/myquant\\_zjyfs6.htm](https://www.wenhua.com.cn/guide/myquant_zjyfs6.htm)
- ③ 账户函数取交易账户的数据，不需要在 Data 区定义；
- ④ 独立的算法交易模型不支持定义参数。

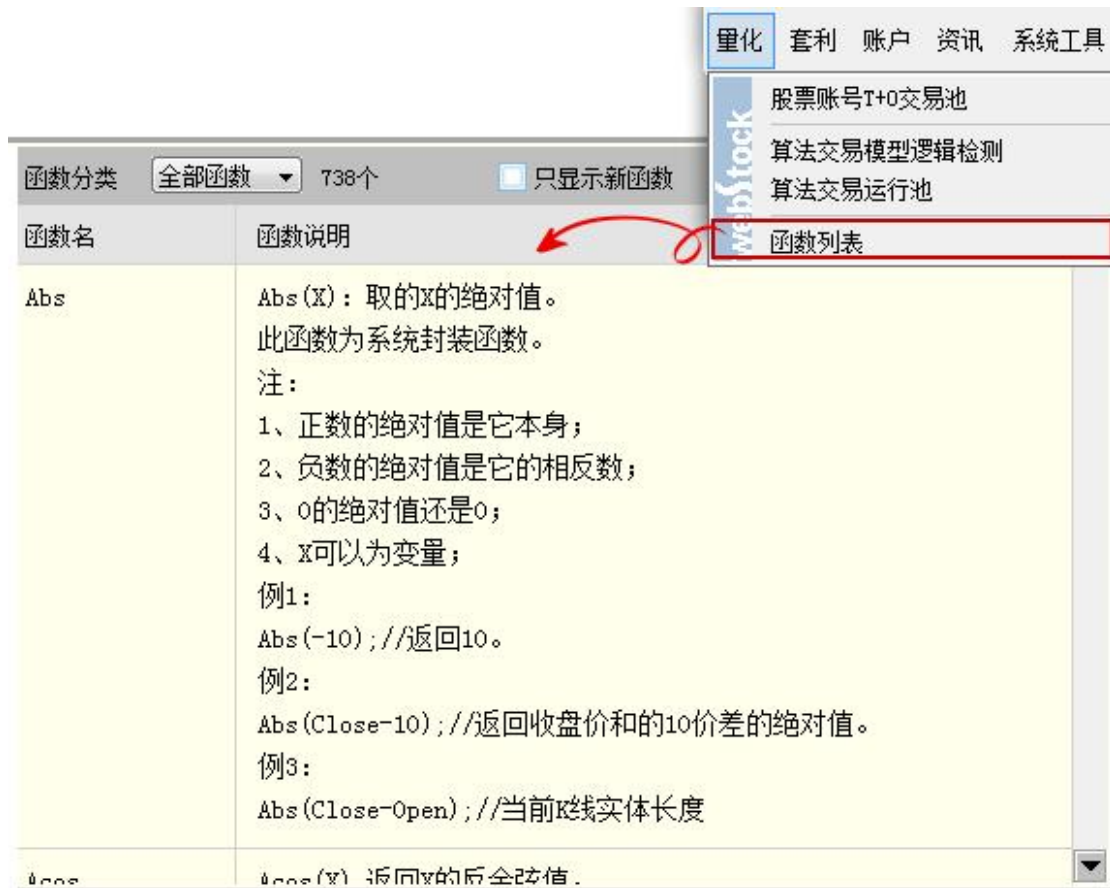
## 2、查找函数

### (1) 系统函数

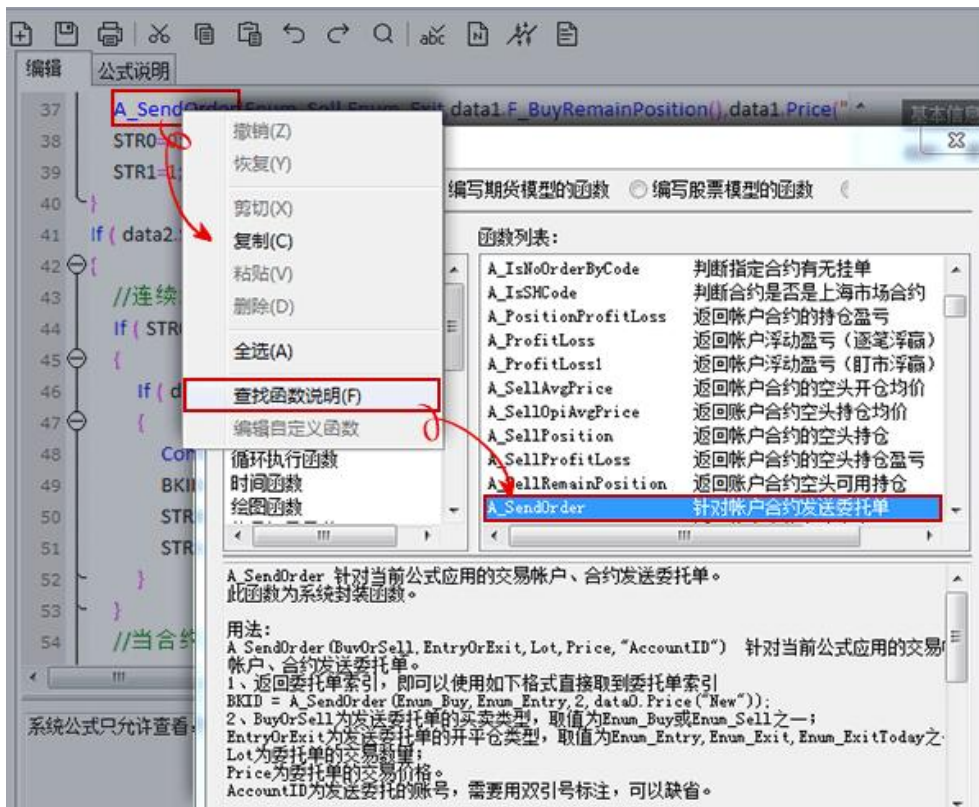
T9 支持丰富的 WFC 开源函数类库，提供近 200 个开源函数，共 700 余个系统函数，支持金融统计类函数、循环执行函数、基本面函数、头寸类函数、手动下单辅助函数、期权类

函数等交易建模所需的各类函数，支持用户自由实现各类量化交易思路。

如下图所示，在量化菜单下点击【函数列表】，可查看函数的具体分类和使用说明。



编写时，在编写平台上双击蓝色的系统函数，单击鼠标右键【查找函数说明】可查看函数具体使用方法。



## (2) 自定义函数

如果要在模型中调用比较复杂的计算，而现有的函数又无法满足需要，则可以通过创建自定义函数来实现。熟练的使用自定义函数可以简化代码，减少出错的可能，提高编写效率。

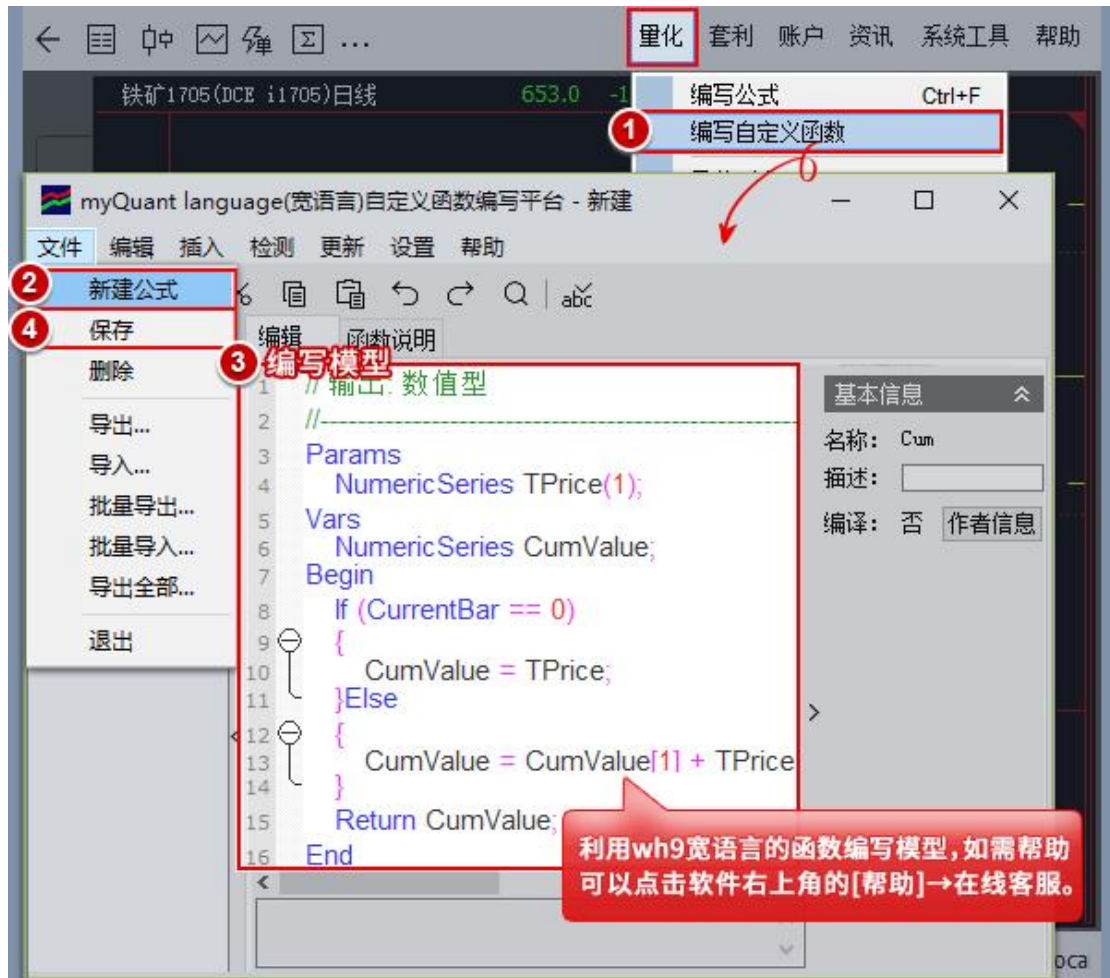
### ①自定义函数的结构：

自定义函数由参数定义、变量定义、主程序三部分构成，如下图所示：



### ②编写自定义函数：

如下图所示，是如何编写用户自定义函数。



- 自定义函数语法介绍: [https://www.wenhua.com.cn/guide/myQuant\\_zjyfh2.htm](https://www.wenhua.com.cn/guide/myQuant_zjyfh2.htm)
- 自定义函数不能直接加载在图表上, 只能在其他自定义函数或公式应用中调用;
- 未经过编译保存的自定义函数不能在其他公式中被引用;

### 3、信号指令和下单委托函数

#### (1) 六种信号指令: BK、SK、SP、BP、BPK、SPK

- 信号指令根据 K 线图表数据计算信号, 可显示信号图标并对信号做历史回测, 不支持下单委托;
- 可用于基于图表数据计算的 K 线图公式, 不可用于独立的算法模型;
- 常见思路: 通过函数获取信号的价位、距离、信号到目前的高低点等;
- 信号指令如下表, 具体使用方法请在“公式编写平台”->【帮助】->“指令说明”中查看:

指令	指令图标	指令函数
建立多头持仓		BK(N,Price,IsCancel);
建立空头持仓		SK(N,Price,IsCancel);
平多头持仓		SP(N,Price,IsCancel);
平空头持仓		BP(N,Price,IsCancel);
买平后买开新仓		BPK(N,Price,IsCancel);
卖平后卖开新仓		SPK(N,Price,IsCancel);

## (2) 下单委托函数: A\_SendOrder

- . 不计算信号，根据条件直接发委托单；
- . 可用于基于 K 线图的公式/独立的算法交易模型：趋势模型中，不能用于历史测试，只能用于实时行情交易；独立的算法交易模型中，可以用于盘口算法模型回测；
- . 常见思路：获取账户信息、盘口高频数据等实现精细化下单；
- . 函数用法如下，更多使用方法请在软件右上方菜单【量化】->“函数列表”中查看：

**A\_SendOrder (BuyOrSell, EntryOrExit, Lot, Price, "AccountID")**

**买卖            开平            数量   价格   交易账号**

注：

- 1、基于 K 线数据的公式中可以同时使用上述两种指令；
- 2、交易需写入下单委托函数 A\_SendOrder，信号指令只在图表上显示信号标识，不能下单委托。

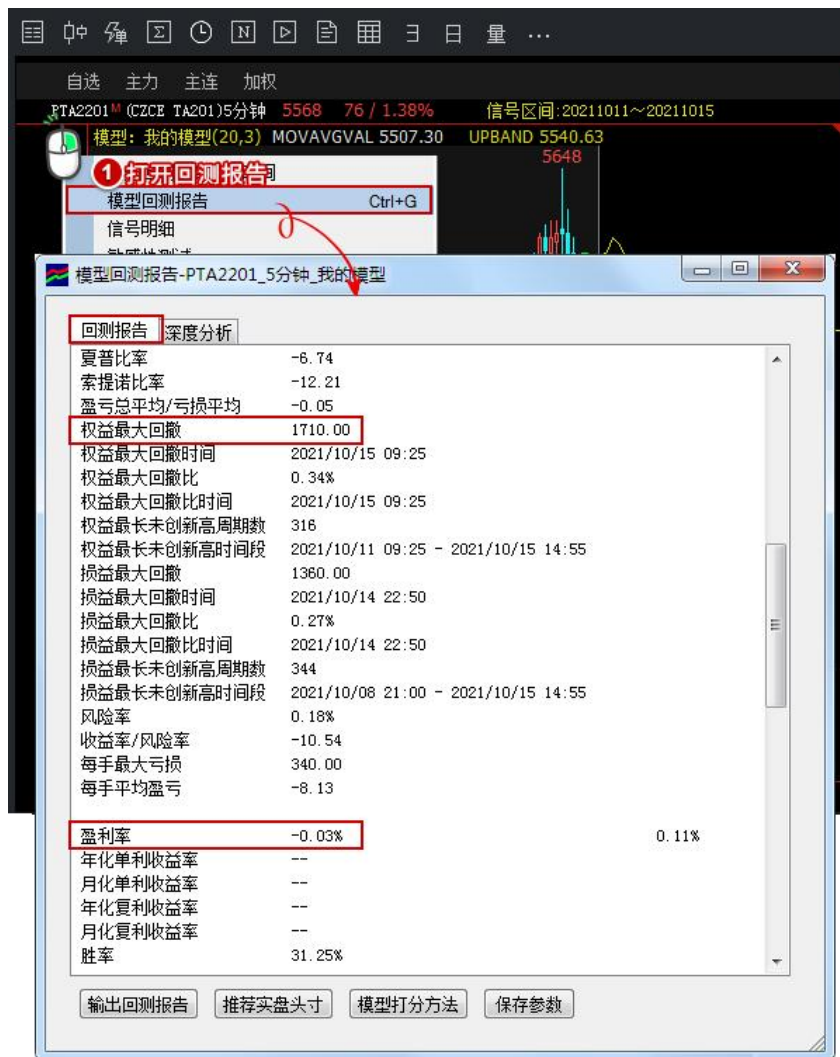
## (二) 模型测试和优化

### 1、测试模型在历史 K 线上的效果（查看信号回测报告）

当有了模型后，我们通常是不敢马上进行实盘交易的，因为我们不了解模型，不知道它与我们的交易思路是否相符，盈利率是多少、胜率是多少、多久会出一次交易信号等等。只

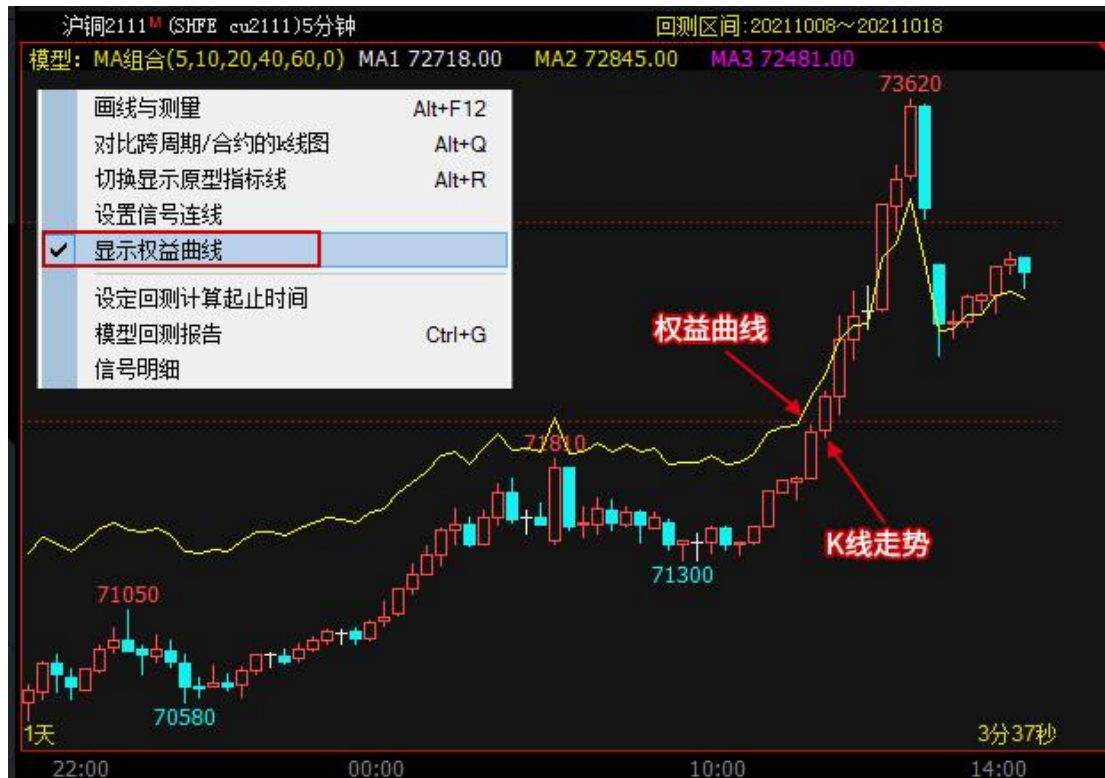
有了了解模型才能信任模型并放心的在实盘中应用它，所以在实盘交易前需要先检验模型在历史k线上的效果，了解模型的运行效果和局限性。

如下图，将模型加载到K线图，在K线图上点击鼠标右键->【模型回测报告】，即可查看详细的分析报告。通过回测分析报告中提供的各个参考指标，投资者可360度的检验模型好坏。



在K线上点击鼠标右键，勾选【显示权益曲线】，可直观的看到资金曲线的变化情况，检查模型的效果是否符合我们的预期要求。





## 2、了解模型（查看模型信号明细）

资金最大回撤发生在哪一根 K 线？出现资金最大回撤时的几笔交易的盈亏都是多少？模型测试的每笔交易的时间和价格具体是多少？这一系列问题都可在“信号明细”中找到答案。

如下图，加载模型回测后，在 K 线图上点击鼠标右键->【信号明细】，可查看详细的信号成交明细。



### 3、测试模型敏感度

模型对于参数变动的敏感程度，反映了它对市场变化的适应能力，参数稍有风吹草动，收益率就大幅变动的模型显然是不稳定的。敏感性测试图以滑点、手续费等作为横坐标，以平均盈利/平均亏损、收益率等参数为纵坐标，用线型形态直观展现出参数变化对平均盈亏、胜率等指标的影响，测试模型的敏感程度，帮助我们找到更好的模型。

如下图，加载模型回测后，在K线图右键菜单【敏感性测试】，选好横坐标和纵坐标要考量的变量后，点击计算，查看敏感性测试图。



#### 4、优化模型参数

我们会发现在一段时间内表现很好的模型，过了一段时间就好像失效了一样，这种情况可能是由于模型参数不再适应当前市场行情引起的，我们需要统计历史数据寻找新的最优参数，但单凭人工计算几乎是不可能的。利用“参数优化”功能，可在指定的范围内让计算机筛选出最适合当前行情的参数。

如下图，加载模型回测后，在K线图右键菜单选择【参数优化】，可通过①【枚举】按钮，进行大范围海选抽样；通过②【遗传】按钮，在已经筛选的优质参数附近寻找更优的参

数。参数优化原理参考“[参数优化让模型达到最优](#)”。



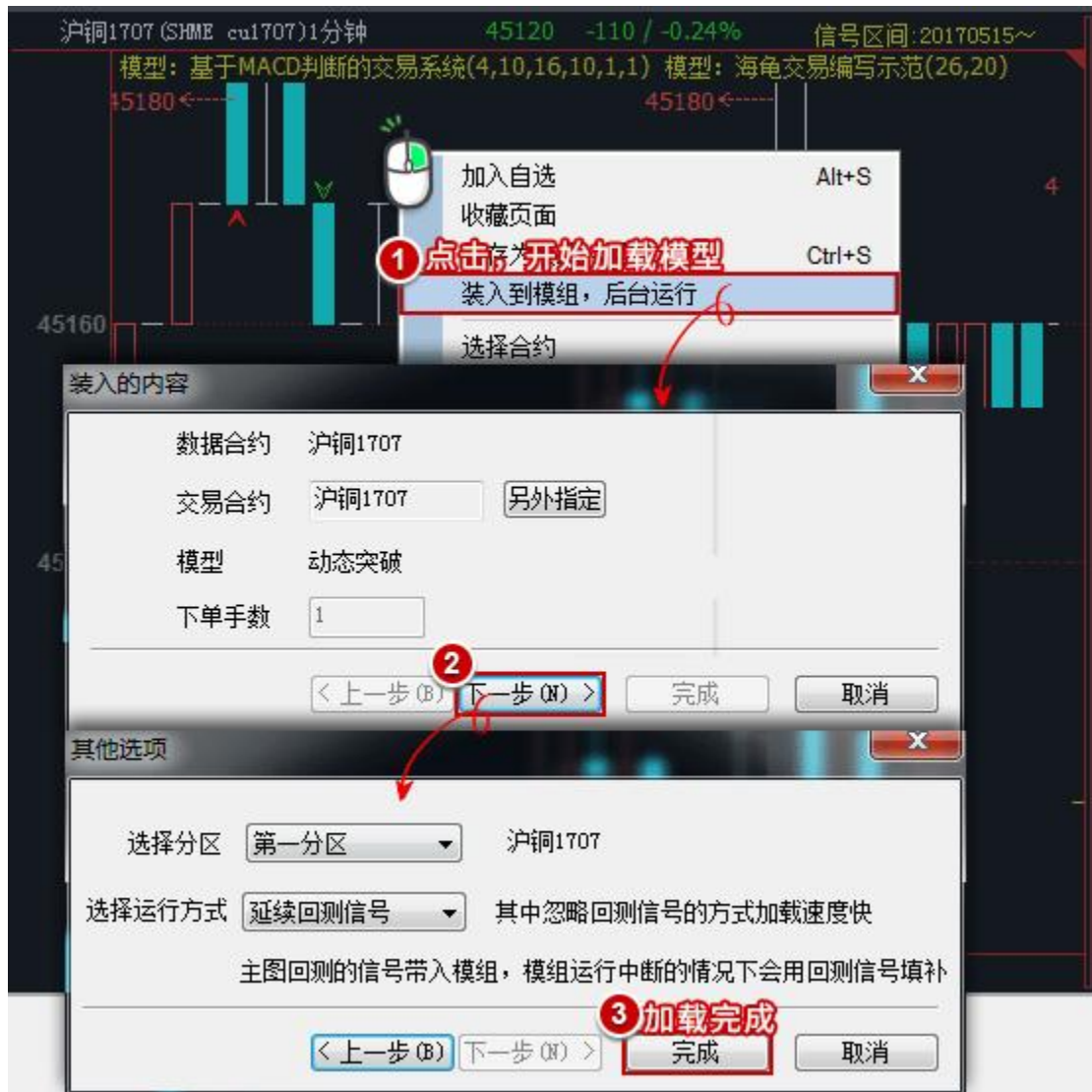
### (三) 运行模型自动交易

通过历史回测和参数优化，模型的执行效果得到了大幅提升。但是仅有模型是不能自动交易的，还需要借助量化平台运行，自动下单。

T9 针对不同的期货量化策略提供不同的运行平台，趋势策略里写算法交易的模型需要在期货合约运行模组上基于图表数据运行，独立的算法交易模型则需要在算法交易模型运行池中后台运行。

#### 1、添加模型到运行模组全自动运行

对每个策略单独分配资金，进行独立的资金管理、风险控制，这样的需求可以在运行模组中实现。如下图，是如何将模型加载到模组运行的操作：



完成上面的设置后，按照下图所示方法打开运行模组界面。

行业	日涨幅	日跌幅	60秒涨停	60秒跌停	成交量	持仓量	日增仓	日减仓	沉淀资金
序号	文华码	合约名称	最新	现手	买价	卖价	买量	卖量	>>
7	2182	沪银1712	3912	2	3911	3912	296	75	3
8	2110	沪铜1710	50110	64	50100	50110	23	1	1
9	2210	沪铝1710	16105	46	16100	16105	62	95	2
10	2240	沪锌1710	24450	124	24450	24455	12	14	2
11	2890	沪铅1710	19340	28	19335	19340	3	30	
12	7501	沪镍1801	84950	4	84930	84950	15	32	2
13	6209	沪锡1709	143380	2	143380	143440	2	2	
14	8361	焦炭1801	2107.0	4	2106.5	2107.0	3	8	1
15	8381	焦煤1801	1339.5	16	1339.0	1339.5	18	2	2
16	2771	郑煤1801	585.6	4	585.4	585.6	79	31	
17	21	铁矿1801	524.0	2	524.0	524.5	586	924	11
18	31	螺纹钢1801	3727	82	3726	3727	43	147	15
19	41	热卷1801	3842	6	3841	3842	39	6	5
20	8549	硅铁1709	7300	20	7296	7300	6	12	
21	8561	锰硅1801	7092	2	7092	7094	3	12	
22	8721	玻璃1801	1402	4	1401	1402	176	21	

点击后会弹出如下图界面，不同的页面区域分别显示模组分区、资金、信号、监控 K 线图和运行日志信息。



(1) 建立模组运行单元时交易合约的指定方式（如下图）



数据合约是指加载模型计算理论信号的合约，交易合约是模型出信号后实际交易的合约，可自己设置。

注：

模型中写入 Trade\_Other 指定交易合约函数，此处会自动抓取模型中编写的交易合约，不可设置；数据合约为主连合约或加权合约时，写入 Trade\_Other:Auto;可实现自动换月移仓。

(2) 运行单元的运行方式选择（如下图）



① 延续回测信号运行方式：模组运行单元中历史 k 线图显示回测的信号，后续以历史信号为基础，根据实时数据计算新信号。

模组重新启动影响：

如果本次启动与前一次退出之间产生新数据，缺失的数据会自动申请，并且对这部分数据进行历史回测。回测信号与历史信号同时保留，加载成功之后，延续这些信号继续运行。

**② 忽略回测信号运行方式：模组运行单元中历史 k 线图不显示信号，从加载时开始重新计算信号，不以任何历史信号为基础。**

模组重新启动影响：

如果本次启动与前一次退出之间没有产生新数据，由于没有缺失数据，所以历史信号保留，加载完成之后，延续历史信号运行。

如果本次启动与前一次退出之间产生新数据，缺失的数据会自动申请，但不会进行历史回测，并且清空所有的历史信号。加载成功之后，运行单元资金复原为初始资金，持仓清零，相当于重新开始运行。

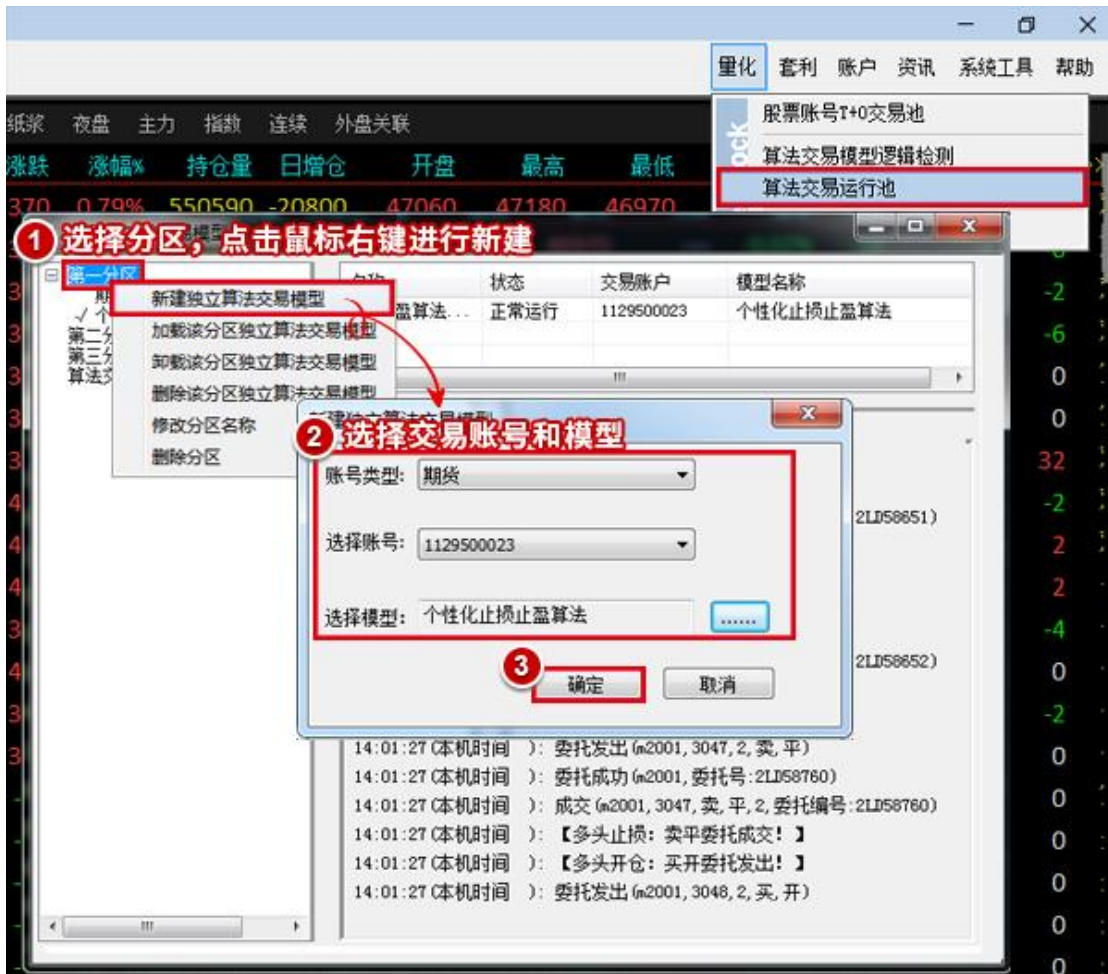
**注：**

选择忽略回测信号的运行方式，运行单元加载成功后，第一个有效信号必须是开仓信号。

## 2、独立算法交易模型运行池

独立的算法交易策略需要在专门的算法交易模型运行池中后台运行。如下图，是如何建立算法交易运行池：





建立好算法交易模型运行池后，可按照下图所示方法打开运行界面查看。



自选一	自选二	自选三	自选四	自选五	主力	
序号	文华码	合约名称	最新	买量	现手	买价
↑ 6	2290	沪金1812	272.55	192	2	272.50
↑ 7	2182	沪银1812	3735	333	2	3735
✓ 8	2107	沪铜1807	52810	33	2	52800
↑ 9	2207	沪铝1807	14700	82	4	14700
↑ 10	2237	沪锌1807	24635	81	6	24635
↑ 11	2887	沪铅1807	20615	7	2	20610
↑ 12	7507	沪镍1807	118240	17	4	118220
↑ 13	6209	沪锡1809	153300	1	2	153260
✓ 14	8369	焦炭1809	2041.0	26	2	2040.5
↑ 15	8389	焦煤1809	1244.0	50	8	1243.5
↑ 16	2779	郑煤1809	633.4	21	2	633.4
✓ 17	6890	螺纹钢1810	3748	436	38	3747

独立算法交易模型运行池

期货户 外盘户 股票户 11:23:18-Local

## 相关问题解答

### 1、模组单元状态列表中各项代表的含义？

下单信号：最后确认、下单的信号（包括信号消失），不显示没有确认的可能变化的信号。

信号手数：下单信号用户设置或模型写入的手数。

下单手数：模组当前单元运行过程中实际的下单手数。

单元持仓：模组当前单元实际持仓。

理论持仓： $BKVOL =$  根据信号下单手数计算的理论持仓，完全根据信号计算。

平仓盈亏：模组当前单元本次打开后产生的累计的平仓盈亏，根据单元持仓的实际成交价格计算。

浮动盈亏：模组当前单元持仓的浮盈，根据单元持仓持仓价格计算。

期初权益：模组本次运行开始时实际资金曲线的权益值。

当前权益：模组运行过程中实际资金曲线的权益值。

可用资金：根据实际资金曲线计算的实际可用资金，可实时查看模组的实际可用资金情况。

理论资金曲线： $MONEYTOT =$ 根据信号的理论持仓以及理论开、平仓价格进行计算。

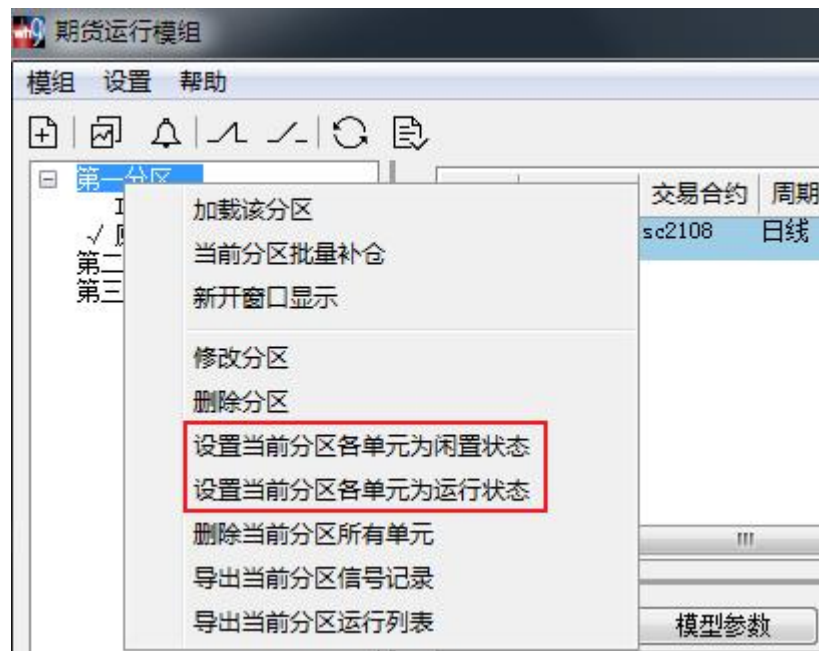
实际资金曲线： $(\text{初始权益} - \text{初始权益包含的浮动盈亏}) + \text{平仓盈亏}(\text{累计值}) + \text{当前浮动盈亏} - \text{手续费}(\text{累计值})$ 。

## 2、模组分区的使用技巧

模组分区像文件夹一样，用户可按品种、按策略、按周期等思路划分，建立不同的分区管理运行单元。

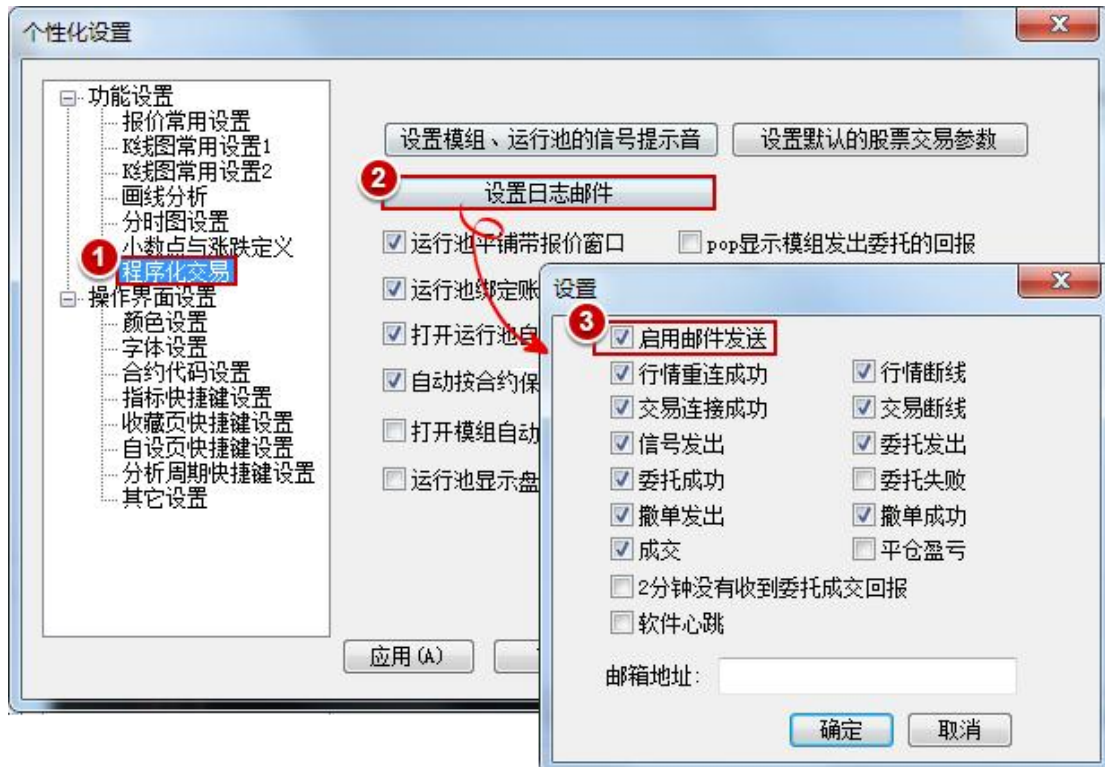
**设置当前分区各单元为闲置状态：**将当前分区下的运行单元设置为退出运行状态，可一次退出分区内所有单元。设置为闲置状态的单元不占用单元运行数量。

**设置当前分区各单元为运行状态：**将当前分区下所有闲置状态的单元恢复为运行状态，可一次打开分区内所有单元。



## 3、如果我不在电脑前，但想知道模型的运行情况怎么办？

答：可以使用【日志邮件】功能，该功能可以帮助您接收到模型运行动态的邮件，人不在电脑前一样可以监控自己的程序。在软件右上方【系统工具】->【个性化设置】中按下图的操作步骤勾选启用后，盒子和模组的交易动态会直接发送至绑定的邮箱中。



#### 4、模组监控 K 线图上的黄白线是什么？

答：黄白线分别为理论资金曲线和实际资金曲线；

理论资金曲线（白线）：根据信号指令计算的单元起始资金运行到当前的剩余可用资金；

实际资金曲线（黄线）：未加载模组运行单元前，实际资金曲线为单元的起始资金；

单元运行后，实际资金曲线=起始资金+当前单元的实际平仓盈亏+当前单元的实际浮动盈亏—手续费成本；

运行单元持续运行，实际资金曲线可以记录当前单元的实际盈亏情况，理论资金曲线与实际资金曲线的对比，可以了解运行单元的滑点成本。

注：

对数坐标和线性坐标：线性坐标表示的是价格变化的绝对值，而对数坐标表示的是价格变化的相对强度。

#### 5、模组运行单元 K 线图右键菜单的“重新计算历史信号”和“清除历史信号重新运行”是什么意思？

答：重新计算历史信号：是重新指定一个日期，从这个日期开始将历史的信号再算一遍，相当于重新回测一遍，历史信号会保留。

清除历史信号重新运行：赋给当前单元一个初始状态，以这个状态重新开始运行，这个状态包括持仓方向、手数、价格等，会忽略历史信号。

## 6、运行单元初始资金的作用

答：模组中每个运行的模型称为单元，单元之间独立运行。运行单元中的模型开仓需要占用保证金，设置单元的初始资金，便可用于单元开仓使用。单元运行所产生的资金的变化，也是基于初始资金的设置。

# 三、期货量化编写举例

## （一）趋势策略+算法下单

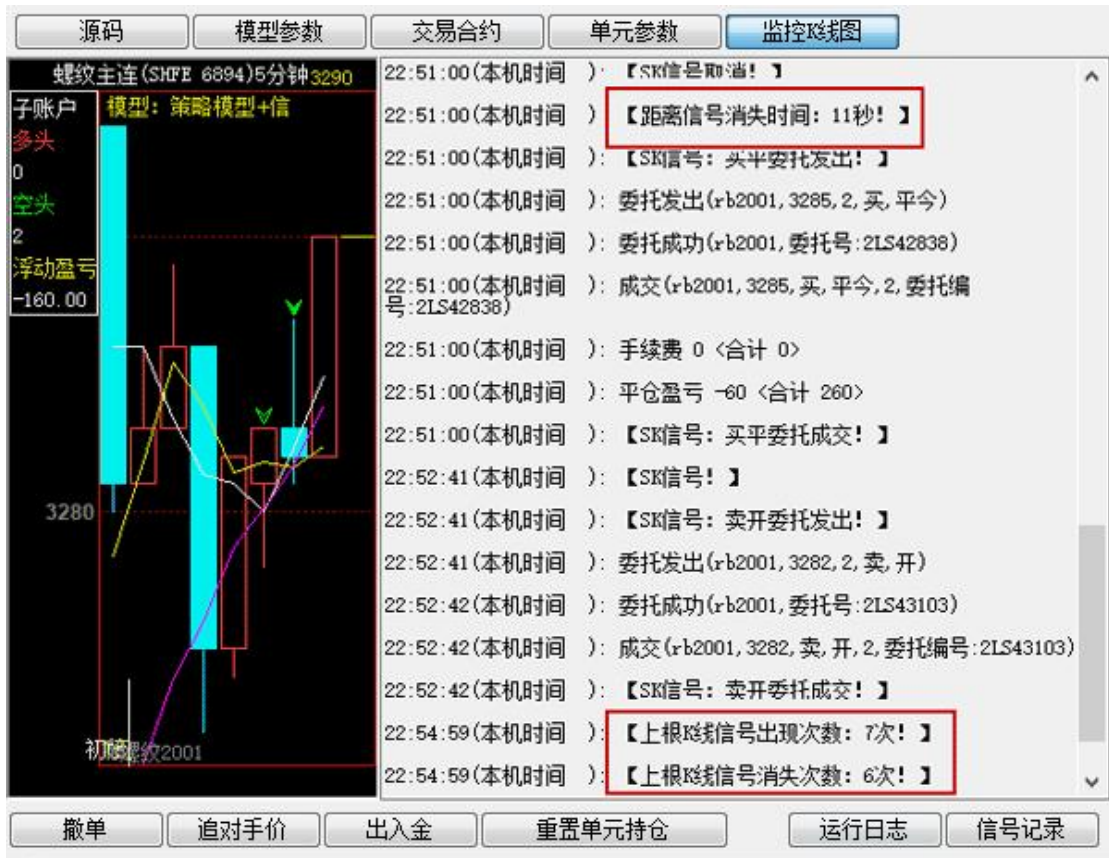
很多成功的量化投资者每年算一笔账，在滑点上的损失比交的手续费还要多。同时趋势策略中固有的信号处理机制难以满足用户日益丰满的思路，越来越多的投资者将研究重心转移到用算法模型对信号和下单的精细化控制方向上。

算法交易可以接管策略信号和委托过程，我们可以完全抛开系统默认的信号处理机制，重新制定一套适合当前策略的信号处理方式，对下单过程精细化管理。

### 1、案例一：策略模型+信号控制算法

指令价模型可以定位到更好的入场点，但也会出现信号消失的问题。太晚入场，会失去好的入场时机，导致策略收益下降；过早入场，可能会频繁出现信号消失，导致恢复持仓成本增加。我们需要找到一个平衡点，使信号消失成本更低同时还能获得更多的收益。

如下图，我们用算法交易接管趋势策略的信号，结合模型信号消失的特点自定义一套信号执行方案：出信号立即下单，当信号消失达到一定时长后再恢复持仓，算法交易动态统计给出每根 K 线上的信号消失时间和消失频率，我们通过调整信号的确认时间便可以达到降低信号消失成本同时提高整体收益的效果。



### 思路解析:

/\*

趋势策略均线交叉开平仓, 出现信号立刻下单, 信号消失且连续 T 秒没有再次出现则恢复持仓

\*/

### 编写示例:

Vars

Numeric N; //下单手数

Numeric T; //时间间隔

Numeric SH; //平仓参数

Numeric MA3; //3 周期均线

Numeric MA5; //5 周期均线

Numeric MA10; //10 周期均线

Numeric KCP; //K 线位置

Numeric SIGID; //指令序号

Numeric NOW; //当前时间

Numeric BIDP; //买一价

Numeric ASKP; //卖一价

Numeric BRP; //多头可用持仓

Numeric SRP; //空头可用持仓

Numeric OPFLG; //模型处理标志

Numeric BKDFLG; //买开处理标志

Numeric SKDFLG; //卖开处理标志

Numeric BPDFLG; //买平处理标志

Numeric SPDFLG; //卖平处理标志

Global\_Numeric SIGIDA; //记录指令序号

Global\_Numeric KCPA; //记录 K 线位置

Global\_Numeric KCPB; //记录 K 线位置

Global\_Numeric SIGF; //信号类型标志

Global\_Numeric SIGCF; //当前信号类型

Global\_Numeric SIGCN; //信号出现次数

Global\_Numeric SIGXN; //信号消失次数

Global\_Numeric SIGXT; //信号消失时间

Global\_Numeric SIGFLG; //信号处理标志

Global\_Numeric BKID; //买开委托

Global\_Numeric SKID; //卖开委托

Global\_Numeric BPID; //买平委托

Global\_Numeric SPID; //卖平委托

Global\_Numeric BKFLG; //买开标志

Global\_Numeric SKFLG; //卖开标志

Global\_Numeric BPFLG; //买平标志

Global\_Numeric SPFLG; //卖平标志

Global\_Numeric BKM; //买开委托手数

Global\_Numeric SKM; //卖开委托手数

Global\_Numeric BPM; //买平委托手数

Global\_Numeric SPM; //卖平委托手数

```
SignalNoTrading:1; //出信号不下单

Begin

//-----处理开启-----//

If(A_IsExchangeOpen() == 1) //如果是开盘状态

{

    If(RunStart() == 1) //如果已初始化成功

    {

        OPFLG = 1; //开启模型处理

    }

}

//-----变量赋值-----//

If(OPFLG == 1) //变量赋值

{

    N = 2; //下单手数

    T = 10; //时间间隔

    KCP = CurrentBar; //K 线位置

    NOW = CurrentTime(); //当前时间

    BIDP = Price("Bid1"); //买一价

    ASKP = Price("Ask1"); //卖一价

    BRP = F_BuyRemainPosition(); //多头可用持仓

    SRP = F_SellRemainPosition(); //空头可用持仓

    If(A_IsSHCode() == 1) //如果是上期所合约

    {

        SH = Enum_ExitToday; //平仓参数

    }

    Else //如果非上期所合约

    {

        SH = Enum_Exit; //平仓参数

    }

}

//-----信号发出-----//

If(1) //信号发出
```





```
{  
    MA3 = Ma(Close,3); //3 周期均线  
    MA5 = Ma(Close,5); //5 周期均线  
    MA10 = Ma(Close,10); //10 周期均线  
    PlotLine("Ma3",Ma3); //3 周期均线  
    PlotLine("Ma5",Ma5); //5 周期均线  
    PlotLine("Ma10",Ma10); //10 周期均线  
    If(Cross(Ma5,Ma10) == 1 && MarketPosition == 0) //如果满足做多条件  
    {  
        BK; //发出做多信号  
    }  
    Else If(CrossDown(Ma3,Ma10) == 1 && MarketPosition == 1) //如果满足平多条件  
    {  
        SP; //发出平多信号  
    }  
    Else If(CrossDown(Ma5,Ma10) == 1 && MarketPosition == 0) //如果满足做空条件  
    {  
        SK; //发出做空信号  
    }  
    Else If(Cross(Ma3,Ma10) == 1 && MarketPosition == -1) //如果满足平空条件  
    {  
        BP; //发出平空信号  
    }  
}  
//-----信号处理-----//  
If(OPFLG == 1) //信号处理  
{  
    If(KCP != KCPB) //如果是新 K 线  
    {  
        If(SIGCN > 0) //如果上根 K 线出现信号  
        {  
            Commentary("【上根 K 线信号出现次数: " + Text(SIGCN) + "次! 】");  
        }  
    }  
}
```

```
Commentary("【上根 K 线信号消失次数: " + Text(SIGXN) + "次! 】");

If(SIGFLG == 2) //如果信号已消失

{

    Commentary("【距离信号消失时间: " + Text(TimeDiff(SIGXT,NOW)) + "秒! 】");

}

}

SIGCN = 0; //信号出现次数

SIGXN = 0; //信号消失次数

KCPB = KCP; //记录 K 线位置

}

SIGID = F_CurrentSigIndex; //指令序号

If(SIGID != 0) //如果当笔行情发出指令

{

    If(SIGID != SIGIDA || KCP != KCPA) //如果当前信号是未处理过的新信号

    {

        SIGIDA = SIGID; //记录指令序号

        KCPA = KCP; //记录 K 线位置

        If(F_CurrentSig == Sig_BK) //如果信号类型为 BK

        {

            Commentary("【BK 信号! 】");

            BKDFLG = 1; //开启 BK 信号买开处理

            SIGF = 1; //BK 信号已发出

        }

        Else If(F_CurrentSig == Sig_SP) //如果信号类型为 SP

        {

            Commentary("【SP 信号! 】");

            SPDFLG = 1; //开启 SP 信号卖平处理

            SIGF = 2; //SP 信号已发出

        }

        Else If(F_CurrentSig == Sig_SK) //如果信号类型为 SK

        {

            Commentary("【SK 信号! 】");
```

```
SKDFLG = 1; //开启 SK 信号卖开处理

SIGF = 3; //SK 信号已发出

}

Else If(F_CurrentSig == Sig_BP) //如果信号类型为 BP

{

Commentary("【BP 信号!】");

BPDFLG = 1; //开启 BP 信号买平处理

SIGF = 4; //BP 信号已发出

}

SIGCN = SIGCN + 1; //信号出现次数

}

}

If(KCP == KCPA) //如果是当根 K 线

{

If(SIGCF != SIGF) //如果是新的信号类型

{

SIGFLG = 1; //信号已发出

SIGCF = SIGF; //当前信号类型

}

If(SIGFLG == 1) //如果信号已发出

{

If(F_CurrentSig == 0) //如果信号消失

{

If(SIGCF == 1) //如果当前是 BK 信号

{

Commentary("【BK 信号消失!】");

}

Else If(SIGCF == 2) //如果当前是 SP 信号

{

Commentary("【SP 信号消失!】");

}

Else If(SIGCF == 3) //如果当前是 SK 信号
```

```
{
    Commentary("【SK 信号消失!】");
}

Else If(SIGCF == 4) //如果当前是 BP 信号
{
    Commentary("【BP 信号消失!】");
}

SIGIDA = 0; //记录指令序号归 0
SIGXT = NOW; //信号消失时间
SIGXN = SIGXN + 1; //信号消失次数
SIGFLG = 2; //信号已消失
}
}

If(SIGFLG == 2) //如果信号已消失
{
    If(TimeDiff(SIGXT,NOW) <= T) //如果时间间隔 T 秒内
    {
        If(F_CurrentSig > 0) //如果信号出现
        {
            If(SIGCF == 1) //如果当前是 BK 信号
            {
                Commentary("【BK 信号重现!】");
            }

            Else If(SIGCF == 2) //如果当前是 SP 信号
            {
                Commentary("【SP 信号重现!】");
            }

            Else If(SIGCF == 3) //如果当前是 SK 信号
            {
                Commentary("【SK 信号重现!】");
            }
        }
    }
}
```

```
Else If(SIGCF == 4) //如果当前是 BP 信号
{
    Commentary("【BP 信号重现!】");
}

Commentary("【距离信号消失时间: " + Text(TimeDiff(SIGXT,NOW)) + "秒!】");
SIGFLG = 1; //信号已发出
}
}

Else If(TimeDiff(SIGXT,NOW) > T) //如果时间间隔 T 秒
{
    If(SIGCF == 1) //如果当前是 BK 信号
    {
        Commentary("【BK 信号取消!】");
        SPDFLG = 2; //开启 BK 信号卖平处理
    }

    Else If(SIGCF == 2) //如果当前是 SP 信号
    {
        Commentary("【SP 信号取消!】");
        BKDFLG = 2; //开启 SP 信号买开处理
    }

    Else If(SIGCF == 3) //如果当前是 SK 信号
    {
        Commentary("【SK 信号取消!】");
        BPDFLG = 2; //开启 SK 信号买平处理
    }

    Else If(SIGCF == 4) //如果当前是 BP 信号
    {
        Commentary("【BP 信号取消!】");
        SKDFLG = 2; //开启 BP 信号卖开处理
    }

    Commentary("【距离信号消失时间: " + Text(TimeDiff(SIGXT,NOW)) + "秒!】");
    SIGFLG = 0; //信号已取消
}
```

```
    }  
  }  
}  
  
//-----成交判断-----//  
If(OPFLG == 1) //成交判断  
{  
  If(BKFLG > 0) //如果有买开委托  
  {  
    If(F_OrderStatus(BKID) == Enum_Filled) //如果买开委托成交  
    {  
      If(BKFLG == 1) //如果是 BK 信号买开委托  
      {  
        Commentary("【BK 信号：买开委托成交！】");  
      }  
      Else If(BKFLG == 2) //如果是 SP 信号买开委托  
      {  
        Commentary("【SP 信号：买开委托成交！】");  
      }  
      BKFLG = 0; //买开标志归 0  
    }  
  }  
  If(SPFLG > 0) //如果有卖平委托  
  {  
    If(F_OrderStatus(SPID) == Enum_Filled) //如果卖平委托成交  
    {  
      If(SPFLG == 1) //如果是 SP 信号卖平委托  
      {  
        Commentary("【SP 信号：卖平委托成交！】");  
      }  
      Else If(SPFLG == 2) //如果是 BK 信号卖平委托  
      {  
        Commentary("【BK 信号：卖平委托成交！】");  
      }  
    }  
  }  
}
```

```
    }  
    SPFLG = 0; //卖平标志归 0  
  }  
}  
If(SKFLG > 0) //如果有卖开委托  
{  
  If(F_OrderStatus(SKID) == Enum_Filled) //如果卖开委托成交  
  {  
    If(SKFLG == 1) //如果是 SK 信号卖开委托  
    {  
      Commentary("【SK 信号： 卖开委托成交！】");  
    }  
    Else If(SKFLG == 2) //如果是 BP 信号卖开委托  
    {  
      Commentary("【BP 信号： 卖开委托成交！】");  
    }  
    SKFLG = 0; //卖开标志归 0  
  }  
}  
If(BPFLG > 0) //如果有买平委托  
{  
  If(F_OrderStatus(BPID) == Enum_Filled) //如果买平委托成交  
  {  
    If(BPFLG == 1) //如果是 BP 信号买平委托  
    {  
      Commentary("【BP 信号： 买平委托成交！】");  
    }  
    Else If(BPFLG == 2) //如果是 SK 信号买平委托  
    {  
      Commentary("【SK 信号： 买平委托成交！】");  
    }  
    BPFLG = 0; //买平标志归 0
```



```
    }
  }
}

//-----委托处理-----//

If(OPFLG == 1) //委托处理
{
  If(BKDFLG > 0) //如果已开启买开处理
  {
    If(BKFLG == 0) //如果没有买开委托
    {
      BKM = N - BRP; //买开委托手数

      If(BKM > 0) //如果买开委托手数大于 0
      {
        If(BKDFLG == 1) //如果是 BK 信号买开
        {
          Commentary("【BK 信号：买开委托发出！】");
        }

        Else If(BKDFLG == 2) //如果是 SP 信号买开
        {
          Commentary("【SP 信号：买开委托发出！】");
        }

        BKID = A_SendOrder(Enum_Buy, Enum_Entry, BKM, ASKP); //发出买开委托

        BKFLG = BKDFLG; //已发出买开委托
      }
    }
  }

  If(SPDFLG > 0) //如果已开启卖平处理
  {
    If(SPFLG == 0) //如果没有卖平委托
    {
      If(BRP > 0) //如果有多头可用持仓
      {
```



```
SPM = BRP; //卖平委托手数

If(SPDFLG == 1) //如果是 SP 信号卖平
{
    Commentary("【SP 信号： 卖平委托发出！】");
}

Else If(SPDFLG == 2) //如果是 BK 信号卖平
{
    Commentary("【BK 信号： 卖平委托发出！】");
}

SPID = A_SendOrder(Enum_Sell,SH,SPM,BIDP); //发出卖平委托
SPFLG = SPDFLG; //已发出卖平委托
}
}

If(SKDFLG > 0) //如果已开启卖开处理
{
    If(SKFLG == 0) //如果没有卖开委托
    {
        SKM = N - SRP; //卖开委托手数

        If(SKM > 0) //如果卖开委托手数大于 0
        {
            If(SKDFLG == 1) //如果是 SK 信号卖开
            {
                Commentary("【SK 信号： 卖开委托发出！】");
            }

            Else If(SKDFLG == 2) //如果是 BP 信号卖开
            {
                Commentary("【BP 信号： 卖开委托发出！】");
            }

            SKID = A_SendOrder(Enum_Sell,Enum_Entry,SKM,BIDP); //发出卖开委托
            SKFLG = SKDFLG; //已发出卖开委托
        }
    }
}
```

```
    }  
  }  
  If(BPDFLG > 0) //如果已开启买平处理  
  {  
    If(BPFLG == 0) //如果没有买平委托  
    {  
      If(SRP > 0) //如果有空头可用持仓  
      {  
        BPM = SRP; //买平委托手数  
        If(BPDFLG == 1) //如果是 BP 信号买平  
        {  
          Commentary("【BP 信号：买平委托发出！】");  
        }  
        Else If(BPDFLG == 2) //如果是 SK 信号买平  
        {  
          Commentary("【SK 信号：买平委托发出！】");  
        }  
        BPID = A_SendOrder(Enum_Buy,SH,BPM,ASKP); //发出买平委托  
        BPFLG = BPDFLG; //已发出买平委托  
      }  
    }  
  }  
End
```

**注：**

趋势策略里写算法模型时，BK/SK/BP/SP/BPK/SPK 信号指令仅用于在图表上查看信号标识，下单交易需写入发送委托单函数 A\_SendOrder，在模组/交易池运行检验下单效果。

## 2、案例二：策略模型+追价算法

投资者交易时往往有这样的矛盾：想成交在理想点位但经常不能及时成交，如果追求成交速度又不得不承担一部分滑点成本，如何才能既能保证成交又能使得交易的滑点更小呢？

如下图，我们使用算法交易接管策略委托过程，对下单进行更精细的处理：当行情平稳时采用对价委托；当遭遇极端行情价格出现较大偏离时，针对挂单方向区别处理，对开仓挂单进行撤单，避免出现更大的滑点，对平仓挂单进行市价追价，确保成交迅速离场，如此便可达到降低交易滑点，保证成交效率的效果。



### 思路解析:

/\*

限定价格偏离初始价格的范围，当行情波动平稳时，对价委托连续追价；

当价格偏离超过此范围时，在停止追价后对后续交易进行细化处理：

如果是开仓委托，则撤单终止交易；

如果是平仓委托，则最后一次委托采用市价委托确保成交。

\*/

### 编写示例:

Vars

Numeric N; //下单手数

Numeric T; //时间间隔

Numeric W; //委托价位倍数

Numeric PLM; //偏离价位倍数

Numeric SH; //平仓参数

Numeric MA3; //3 周期均线

Numeric MA5; //5 周期均线

Numeric MA10; //10 周期均线

Numeric KCP; //K 线位置

Numeric SIGID; //指令序号

Numeric NOW; //当前时间

Numeric NEWP; //最新价

Numeric BIDP; //买一价

Numeric ASKP; //卖一价

Numeric RLP; //涨停价

Numeric FLP; //跌停价

Numeric MINP; //最小变动价位

Numeric BRP; //多头可用持仓

Numeric SRP; //空头可用持仓

Numeric OPFLG; //模型处理标志

Numeric BKDFLG; //买开处理标志

Numeric SKDFLG; //卖开处理标志

Numeric BPDFLG; //买平处理标志

Numeric SPDFLG; //卖平处理标志

Global\_Numeric SIGIDA; //记录指令序号

Global\_Numeric KCPA; //记录 K 线位置

Global\_Numeric BKID; //买开委托

Global\_Numeric SKID; //卖开委托

Global\_Numeric BPID; //买平委托

Global\_Numeric SPID; //卖平委托

Global\_Numeric BKFLG; //买开标志

Global\_Numeric SKFLG; //卖开标志

```
Global_Numeric BPFLG; //买平标志
Global_Numeric SPFLG; //卖平标志
Global_Numeric BKDEL; //买开撤单标志
Global_Numeric SKDEL; //卖开撤单标志
Global_Numeric BPDEL; //买平撤单标志
Global_Numeric SPDEL; //卖平撤单标志
Global_Numeric BKCP; //买开初始价格
Global_Numeric SKCP; //卖开初始价格
Global_Numeric BPCP; //买平初始价格
Global_Numeric SPCP; //卖平初始价格
Global_Numeric BKM; //买开委托手数
Global_Numeric SKM; //卖开委托手数
Global_Numeric BPM; //买平委托手数
Global_Numeric SPM; //卖平委托手数
Global_Numeric BKP; //买开委托价格
Global_Numeric SKP; //卖开委托价格
Global_Numeric BPP; //买平委托价格
Global_Numeric SPP; //卖平委托价格
Global_Numeric BKT; //买开委托时间
Global_Numeric SKT; //卖开委托时间
Global_Numeric BPT; //买平委托时间
Global_Numeric SPT; //卖平委托时间

Setting

SignalNoTrading:1; //出信号不下单

Begin

//-----处理开启-----//

If(A_IsExchangeOpen() == 1) //如果是开盘状态
{
    If(RunStart() == 1) //如果已初始化成功
    {
        OPFLG = 1; //开启模型处理
    }
}
}
```

```
//-----变量赋值-----//  
  
If(OPFLG == 1) //变量赋值  
{  
    N = 2; //下单手数  
    T = 5; //时间间隔  
    PLM = 3; //偏离价位倍数  
    KCP = CurrentBar; //K 线位置  
    NOW = CurrentTime(); //当前时间  
    NEWP = Price("New"); //最新价  
    BIDP = Price("Bid1"); //买一价  
    ASKP = Price("Ask1"); //卖一价  
    RLP = Price("RiseLimit"); //涨停价  
    FLP = Price("FallLimit"); //跌停价  
    MINP = Price("MinPrice"); //最小变动价位  
    BRP = F_BuyRemainPosition(); //多头可用持仓  
    SRP = F_SellRemainPosition(); //空头可用持仓  
    If(A_IsSHCode() == 1) //如果是上期所合约  
    {  
        SH = Enum_ExitToday; //平仓参数  
    }  
    Else //如果非上期所合约  
    {  
        SH = Enum_Exit; //平仓参数  
    }  
}  
  
//-----信号发出-----//  
  
If(1) //信号发出  
{  
    MA3 = Ma(Close,3); //3 周期均线  
    MA5 = Ma(Close,5); //5 周期均线  
    MA10 = Ma(Close,10); //10 周期均线  
    PlotLine("Ma3",Ma3); //3 周期均线
```

```
PlotLine("Ma5",Ma5); //5 周期均线

PlotLine("Ma10",Ma10); //10 周期均线

If(Ref(Cross(Ma5,Ma10),1) == 1 && MarketPosition == 0) //如果满足做多条件

{

BK; //发出做多信号

}

Else If(Ref(CrossDown(Ma3,Ma10),1) == 1 && MarketPosition == 1) //如果满足平多条件

{

SP; //发出平多信号

}

Else If(Ref(CrossDown(Ma5,Ma10),1) == 1 && MarketPosition == 0) //如果满足做空条件

{

SK; //发出做空信号

}

Else If(Ref(Cross(Ma3,Ma10),1) == 1 && MarketPosition == -1) //如果满足平空条件

{

BP; //发出平空信号

}

}

//-----信号处理-----//

If(OPFLG == 1) //信号处理

{

SIGID = F_CurrentSigIndex; //指令序号

If(SIGID != 0) //如果当笔行情发出指令

{

If(SIGID != SIGIDA || KCP != KCPA) //如果当前信号是未处理过的新信号

{

SIGIDA = SIGID; //记录指令序号

KCPA = KCP; //记录 K 线位置

If(F_CurrentSig == Sig_BK) //如果信号类型为 BK

{

Commentary("【BK 信号!】");
```

```
BKDFLG = 1; //开启买开处理
}

Else If(F_CurrentSig == Sig_SP) //如果信号类型为 SP
{
Commentary("【SP 信号!】");
SPDFLG = 1; //开启卖平处理
}

Else If(F_CurrentSig == Sig_SK) //如果信号类型为 SK
{
Commentary("【SK 信号!】");
SKDFLG = 1; //开启卖开处理
}

Else If(F_CurrentSig == Sig_BP) //如果信号类型为 BP
{
Commentary("【BP 信号!】");
BPDFLG = 1; //开启买平处理
}
}
}
}

//-----成交判断-----//
If(OPFLG == 1) //成交判断
{
If(BKFLG == 1) //如果有买开委托
{
If(F_OrderStatus(BKID) == Enum_Filled) //如果买开委托成交
{
Commentary("【BK 信号: 买开委托成交!】");
BKFLG = 0; //买开标志归 0
BKDEL = 0; //买开撤单标志归 0
}
}
}
Else If(F_OrderStatus(BKID) == Enum_Canceled) //如果买开委托已撤
```



```
{  
Commentary("【BK 信号：买开委托已撤！】");  
If(BKDEL == 1) //如果是价格偏离过大撤单  
{  
BKFLG = 0; //买开标志归 0  
}  
Else If(BKDEL == 2) //如果是时间间隔 T 秒追价  
{  
  
If(F_OrderFilledLot(BKID) > 0) //如果买开委托部分成交  
{  
BKM = BKM - F_OrderFilledLot(BKID); //买开委托手数  
}  
If(BKM > 0) //如果买开委托手数大于 0  
{  
BKP = ASKP; //买开委托价格  
Commentary("【BK 信号：买开委托追价！】");  
BKID = A_SendOrder(Enum_Buy, Enum_Entry, BKM, BKP); //发出买开委托  
BKT = NOW; //买开委托时间  
}  
}  
BKDEL = 0; //买开撤单标志归 0  
}  
Else If(F_OrderStatus(BKID) == Enum_Declared || F_OrderStatus(BKID) == Enum_FillPart) //如果买开委托挂  
单  
{  
If(BKDEL == 0) //如果未撤单  
{  
If(Abs(NEWP - BKCP) > PLM * MINP) //如果价格偏离过大  
{  
Commentary("【BK 信号：价格偏离过大！】");  
Commentary("【BK 信号：买开委托撤单！】");
```

```
F_DeleteOrder(F_OrderContractNo(BKID)); //撤掉买开委托挂单

BKDEL = 1; //已发出撤掉买开委托挂单

}

Else If(TimeDiff(BKT,NOW) >= T) //如果时间间隔 T 秒

{

Commentary("【BK 信号：买开委托撤单！】");

F_DeleteOrder(F_OrderContractNo(BKID)); //撤掉买开委托挂单

BKDEL = 2; //已发出撤掉买开委托挂单

}

}

}

}

If(SPFLG == 1) //如果有卖平委托

{

If(F_OrderStatus(SPID) == Enum_Filled) //如果卖平委托成交

{

Commentary("【SP 信号：卖平委托成交！】");

SPFLG = 0; //卖平标志归 0

SPDEL = 0; //卖平撤单标志归 0

}

Else If(F_OrderStatus(SPID) == Enum_Canceled) //如果卖平委托已撤

{

Commentary("【SP 信号：卖平委托已撤！】");

If(F_OrderFilledLot(SPID) > 0) //如果卖平委托部分成交

{

SPM = SPM - F_OrderFilledLot(SPID); //卖平委托手数

}

If(BRP > 0 && SPM > 0 && SPM <= BRP) //如果卖平委托手数不超过多头可用持仓

{

If(SPDEL == 1) //如果是价格偏离过大追价

{

SPP = FLP; //卖平委托价格
```

```
}  
  
Else If(SPDEL == 2) //如果是时间间隔 T 秒追价  
{  
  
SPP = BIDP; //卖平委托价格  
SPT = NOW; //卖平委托时间  
SPDEL = 0; //卖平撤单标志归 0  
}  
  
Commentary("【SP 信号： 卖平委托追价！】");  
SPID = A_SendOrder(Enum_Sell,SH,SPM,SPP); //发出卖平委托  
}  
  
}  
  
Else If(F_OrderStatus(SPID) == Enum_Declared || F_OrderStatus(SPID) == Enum_FillPart) //如果卖平委托挂单  
{  
  
If(SPDEL == 0) //如果未撤单  
{  
  
If(Abs(NEWP - SPCP) > PLM * MINP) //如果价格偏离过大  
{  
  
Commentary("【SP 信号： 价格偏离过大！】");  
Commentary("【SP 信号： 卖平委托撤单！】");  
F_DeleteOrder(F_OrderContractNo(SPID)); //撤掉卖平委托挂单  
SPDEL = 1; //已发出撤掉卖平委托挂单  
} Else If(TimeDiff(SPT,NOW) >= T) //如果时间间隔 T 秒  
{  
  
Commentary("【SP 信号： 卖平委托撤单！】");  
F_DeleteOrder(F_OrderContractNo(SPID)); //撤掉卖平委托挂单  
SPDEL = 2; //已发出撤掉卖平委托挂单  
}  
}  
  
}  
  
}  
  
}  
  
If(SKFLG == 1) //如果有卖开委托  
{
```

```
If(F_OrderStatus(SKID) == Enum_Filled) //如果卖开委托成交
{
    Commentary("【SK 信号：卖开委托成交！】");
    SKFLG = 0; //卖开标志归 0
    SKDEL = 0; //卖开撤单标志归 0
}
Else If(F_OrderStatus(SKID) == Enum_Canceled) //如果卖开委托已撤
{
    Commentary("【SK 信号：卖开委托已撤！】");
    If(SKDEL == 1) //如果是价格偏离过大撤单
    {
        SKFLG = 0; //卖开标志归 0
    }
    Else If(SKDEL == 2) //如果是时间间隔 T 秒追价
    {
        If(F_OrderFilledLot(SKID) > 0) //如果卖开委托部分成交
        {
            SKM = SKM - F_OrderFilledLot(SKID); //卖开委托手数
        }
        If(SKM > 0) //如果卖开委托手数大于 0
        {
            SKP = BIDP; //卖开委托价格
            Commentary("【SK 信号：卖开委托追价！】");
            SKID = A_SendOrder(Enum_Sell, Enum_Entry, SKM, SKP); //发出卖开委托
            SKT = NOW; //卖开委托时间
        }
    }
    SKDEL = 0; //卖开撤单标志归 0
}
Else If(F_OrderStatus(SKID) == Enum_Declared || F_OrderStatus(SKID) == Enum_FillPart) //如果卖开委托挂单
{
```

```
If(SKDEL == 0) //如果未撤单
{
    If(Abs(NEWP - SKCP) > PLM * MINP) //如果价格偏离过大
    {
        Commentary("【SK 信号：价格偏离过大！】");
        Commentary("【SK 信号：卖开委托撤单！】");
        F_DeleteOrder(F_OrderContractNo(SKID)); //撤掉卖开委托挂单
        SKDEL = 1; //已发出撤掉卖开委托挂单
    }
    Else If(TimeDiff(SKT,NOW) >= T) //如果时间间隔 T 秒
    {
        Commentary("【SK 信号：卖开委托撤单！】");
        F_DeleteOrder(F_OrderContractNo(SKID)); //撤掉卖开委托挂单
        SKDEL = 2; //已发出撤掉卖开委托挂单
    }
}

If(BPFLG == 1) //如果有买平委托
{
    If(F_OrderStatus(BPID) == Enum_Filled) //如果买平委托成交
    {
        Commentary("【BP 信号：买平委托成交！】");
        BPFLG = 0; //买平标志归 0
        BPDEL = 0; //买平撤单标志归 0
    }
    Else If(F_OrderStatus(BPID) == Enum_Canceled) //如果买平委托已撤
    {
        Commentary("【BP 信号：买平委托已撤！】");
        If(F_OrderFilledLot(BPID) > 0) //如果买平委托部分成交
        {
            BPM = BPM - F_OrderFilledLot(BPID); //买平委托手数
```

```
}  
  
If(SRP > 0 && BPM > 0 && BPM <= SRP) //如果买平委托手数不超过空头可用持仓  
  
{  
  
If(BPDEL == 1) //如果是价格偏离过大追价  
  
{  
  
BPP = RLP; //买平委托价格  
  
}  
  
Else If(BPDEL == 2) //如果是时间间隔 T 秒追价  
  
{  
  
BPP = ASKP; //买平委托价格  
  
BPT = NOW; //买平委托时间  
  
BPDEL = 0; //买平撤单标志归 0  
  
}  
  
Commentary("【BP 信号：买平委托追价！】");  
  
BPID = A_SendOrder(Enum_Buy,SH,BPM,BPP); //发出买平委托  
  
}  
  
}  
  
Else If(F_OrderStatus(BPID) == Enum_Declared || F_OrderStatus(BPID) == Enum_FillPart) //如果买平委托挂  
单  
  
{  
  
If(BPDEL == 0) //如果未撤单  
  
{  
  
If(Abs(NEWP - BPCP) > PLM * MINP) //如果价格偏离过大  
  
{  
  
Commentary("【BP 信号：价格偏离过大！】");  
  
Commentary("【BP 信号：买平委托撤单！】");  
  
F_DeleteOrder(F_OrderContractNo(BPID)); //撤掉买平委托挂单  
  
BPDEL = 1; //已发出撤掉买平委托挂单  
  
}  
  
}  
  
Else If(TimeDiff(BPT,NOW) >= T) //如果时间间隔 T 秒  
  
{  
  
Commentary("【BP 信号：买平委托撤单！】");
```

```

F_DeleteOrder(F_OrderContractNo(BPID)); //撤掉买平委托挂单

BPDEL = 2; //已发出撤掉买平委托挂单

}

}

}

}

}

//-----委托处理-----//

    If(OPFLG == 1) //委托处理
    {

If(BKDFLG == 1) //如果已开启买开处理
{

If(BKFLG == 0) //如果没有买开委托
{

BKM = N; //买开委托手数

BKP = ASKP; //买开委托价格

BKCP = BKP; //买开初始价格

Commentary("【BK 信号：买开委托发出！】");

BKID = A_SendOrder(Enum_Buy,Enum_Entry,BKM,BKP); //发出买开委托

BKT = NOW; //买开委托时间

BKFLG = 1; //已发出买开委托

}

}

If(SPDFLG == 1) //如果已开启卖平处理
{

If(SPFLG == 0) //如果没有卖平委托
{

If(BRP > 0) //如果有多头可用持仓
{

SPM = BRP; //卖平委托手数

SPP = BIDP; //卖平委托价格

SPCP = SPP; //卖平初始价格

```

```
Commentary("【SP 信号： 卖平委托发出！】");
SPID = A_SendOrder(Enum_Sell,SH,SPM,SPP); //发出卖平委托
SPT = NOW; //卖平委托时间
SPFLG = 1; //已发出卖平委托
}
}
}
If(SKDFLG == 1) //如果已开启卖开处理
{
If(SKFLG == 0) //如果没有卖开委托
{
SKM = N; //卖开委托手数
SKP = BIDP; //卖开委托价格
SKCP = SKP; //卖开初始价格
Commentary("【SK 信号： 卖开委托发出！】");
SKID = A_SendOrder(Enum_Sell,Enum_Entry,SKM,SKP); //发出卖开委托
SKT = NOW; //卖开委托时间
SKFLG = 1; //已发出卖开委托
}
}
If(BPDFLG == 1) //如果已开启买平处理
{
If(BPFLG == 0) //如果没有买平委托
{
If(SRP > 0) //如果有空头可用持仓
{
BPM = SRP; //买平委托手数
BPP = ASKP; //买平委托价格
BPCP = BPP; //买平初始价格
Commentary("【BP 信号： 买平委托发出！】");
BPID = A_SendOrder(Enum_Buy,SH,BPM,BPP); //发出买平委托
BPT = NOW; //买平委托时间
```



```
BPFLG = 1; //已发出买平委托
```

```
}  
}  
}  
}  
}  
End
```

### 相关问题解答:

#### 1、T9 编写语法:

[https://www.wenhua.com.cn/guide/myQuant\\_zjyfs2.htm](https://www.wenhua.com.cn/guide/myQuant_zjyfs2.htm)

#### 2、常用系统函数说明:

[https://www.wenhua.com.cn/guide/myQuant\\_zjyfs3.htm](https://www.wenhua.com.cn/guide/myQuant_zjyfs3.htm)

#### 3、T9 常见编写和机制说明:

[https://www.wenhua.com.cn/guide/myquant\\_zjyfs6.htm](https://www.wenhua.com.cn/guide/myquant_zjyfs6.htm)

## (二) 独立算法交易策略

独立的算法交易模型不以 K 线数据为基础，直接调用盘口数据或交易账户信息构建策略。可通过盘口报价和逐笔成交数据挖掘市场信息，调用交易账户的持仓信息、挂单信息、委托回报等对委托过程进行精细化处理，对多个账号批量下单等智能管控。

### 独立的算法交易模型实现个性化下单需求

软件中提供几种默认的止盈止损策略，但有时还是无法完全满足用户的多样性需求。例如，想要在盈利达到一定数值时再启动跟踪止盈策略，系统的跟踪止盈策略是实现不了的，这时我们可以用算法模型结合账户持仓详情，编制属于自己的止盈策略。

### 案例：独立算法交易模型实现个性化止盈止损

如下图，是自定义的跟踪止盈算法策略，用独立算法交易模型调用合约盘口价格和账户中的合约持仓价格、数量，实时对比监测，当盈利达到一定值时再启动跟踪止盈，实现灵活

的止盈止损效果。

名称	状态	交易账户	模型名称
个性化止损止盈算法...	正常运行	1129500023	个性化止损止盈算法

源码	日志
	10:53:24 (本机时间) ): 个性化止损止盈算法_1129500023 (1), 执行成功!
	10:53:26 (本机时间) ): 【多头开仓: 买开委托发出! 】
	10:53:26 (本机时间) ): 委托发出 (m2001, 3048, 2, 买, 开)
	10:53:26 (本机时间) ): 委托成功 (m2001, 委托号: 2LD44114)
	10:53:26 (本机时间) ): <b>算法交易后台计算, 自动止损止盈</b>
	10:53:26 (本机时间) ): 【多头开仓: 买开委托成交! 】
	10:56:01 (本机时间) ): 【多头止盈: 多头止盈开启! 】
	10:56:01 (本机时间) ): 【多头止盈: 卖平委托发出! 】
	10:56:01 (本机时间) ): 委托发出 (m2001, 3050, 2, 卖, 平)
	10:56:01 (本机时间) ): 委托成功 (m2001, 委托号: 2LD44496)
	10:56:01 (本机时间) ): 成交 (m2001, 3050, 卖, 平, 2, 委托编号: 2LD44496)
	10:56:01 (本机时间) ): 【多头止盈: 卖平委托成交! 】
	10:56:01 (本机时间) ): 【多头开仓: 买开委托发出! 】
	10:56:01 (本机时间) ): 委托发出 (m2001, 3051, 2, 买, 开)
	10:56:02 (本机时间) ): 委托成功 (m2001, 委托号: 2LD44498)
	10:56:02 (本机时间) ): 成交 (m2001, 3051, 买, 开, 2, 委托编号: 2LD44498)
	10:56:02 (本机时间) ): 【多头开仓: 买开委托成交! 】
	10:56:11 (本机时间) ): 【多头止损: 卖平委托发出! 】

## 思路解析:

/\*

止盈思路: 买开后, 动态价差达到三个点以上, 启动跟踪止盈策略

止损思路: 买开后, 动态价差达到三个点以上, 启动跟踪止损策略

\*/

## 编写示例:

Data

data0:"m1905";

Vars

Global\_Numeric N;//下单手数

Global\_Numeric M1;//价位倍数 1

Global\_Numeric M2;//价位倍数 2

```
Global_Numeric M3; //价位倍数 3
Global_Numeric BID; //买一价
Global_Numeric ASK; //卖一价
Global_Numeric NEWP; //最新价
Global_Numeric MINP; //最小变动价位
Global_Numeric BKC; //做多条件
Global_Numeric BKID; //买开委托
Global_Numeric BKFLG; //买开标志
Global_Numeric BKAP; //买开委托成交均价
Global_Numeric BHP; //多头最高价
Global_Numeric SPID; //卖平委托
Global_Numeric SPFLG; //卖平标志
Global_Numeric BRP; //多头可用持仓
Global_Numeric BZYFLG; //多头止盈标志

Begin
N = 2; //下单手数
M1 = 3; //价位倍数 1
M2 = 1; //价位倍数 2
M3 = 3; //价位倍数 3
BKC = 1; //做多条件
BID = Price("Bid1"); //买一价
ASK = Price("Ask1"); //卖一价
NEWP = Price("New"); //最新价
MINP = Price("MinPrice"); //最小变动价位
BRP = F_BuyRemainPosition(); //多头可用持仓

if(A_IsExchangeOpen() == 1) //如果当前状态是开盘
{
//-----成交判断-----//

if(BKFLG == 1) //如果有买开委托
{
if(F_OrderStatus(BKID) == Enum_Filled) //如果买开委托成交
{
Commentary("【多头开仓：买开委托成交！】");
BKAP = F_OrderFilledPrice(BKID); //买开委托成交均价
BKFLG = 2; //买开委托已成交
}
}
}

if(SPFLG > 0) //如果有卖平委托
{
if(F_OrderStatus(SPID) == Enum_Filled) //如果卖平委托成交
{
if(SPFLG == 1) //如果是多头止盈卖平委托
```

```
{
    Commentary("【多头止盈：卖平委托成交！】");
}
else if(SPFLG == 2) //如果是多头止损卖平委托
{
    Commentary("【多头止损：卖平委托成交！】");
}
BKFLG = 0; //买开标志归 0
SPFLG = 0; //卖平标志归 0
BZYFLG = 0; //多头止盈标志归 0
}
}
//-----委托处理-----//

if(BKFLG == 0) //如果没有买开委托
{
    if(BKC == 1) //如果满足做多条件
    {
        Commentary("【多头开仓：买开委托发出！】");
        BKID = A_SendOrder(Enum_Buy,Enum_Entry,N,ASK); //以对价发出 N 手的买开委托
        BKFLG = 1; //已发出买开委托
    }
}

if(BKFLG == 2 && SPFLG == 0) //如果买开委托已成交，且没有卖平委托
{
    if(BZYFLG == 0) //如果多头止盈未开启
    {
        if(NEWP >= BKAP + M1 * MINP) //如果最新价大于等于买开委托成交均价加 M1 个价位
        {
            Commentary("【多头止盈：多头止盈开启！】");
            BHP = NEWP; //多头最高价设为最新价
            BZYFLG = 1; //多头止盈开启
        }
    }
    else if(BZYFLG == 1) //如果多头止盈已开启
    {
        BHP = max(NEWP,BHP); //多头最高价
        if(NEWP <= BHP - M2 * MINP) //如果最新价小于等于多头最高价减 M2 个价位
        {
            if(BRP > 0) //如果有多头可用持仓
            {
                Commentary("【多头止盈：卖平委托发出！】");
                SPID = A_SendOrder(Enum_Sell,Enum_Exit,BRP,BID); //以对价发出多头可用持仓手
```

数的卖平委托

```
        SPFLG = 1; //已发出多头止盈卖平委托
        BZYFLG = 2; //多头止盈已卖平
    }
}
}
if(BZYFLG != 2) //如果多头止盈未卖平
{
    if(NEWP <= BKAP - M3 * MINP) //如果最新价小于等于买开委托成交均价减 M3 个价位
    {
        Commentary("【多头止损：卖平委托发出！】");
        SPID = A_SendOrder(Enum_Sell,Enum_Exit,BRP,BID); //以对价发出多头可用持仓手数的卖平委托
        SPFLG = 2; //已发出多头止损卖平委托
    }
}
}
}
End
```

## 相关问题解答

### 1、算法交易可否定义参数？

答：算法交易不可以定义参数。

### 2、算法交易模型是否支持定义序列变量？

答：不支持，因序列变量在每个 BAR 只能有一个值，这个值在行情更新时会不断刷新，直到最后一个 tick 才能将值保存下来，因此，序列变量无法记录盘中每个 tick 运行公式产生的数据，例如，我们要是想对每个 tick 计数，用序列变量就做不到。

### 3、T9 委托索引的取值是怎样的？

答：A\_SendOrder 函数支持返回委托单索引，数值型。通常与 F\_OrderStatus（返回某个委托单的状态）函数连用，判断某个委托单的委托状态，实现对不同委托状态的委托单的差异化处理。参考如下格式取得委托单索引：

```
BKID =data0. A_SendOrder(Enum_Buy,Enum_Entry,2,data0.Price("New"));
```

### 4、枚举值都有哪些函数？

答：

Enum_AmericanOption	返回美式期权的枚举值
Enum_Buy	返回买卖状态的买入枚举值
Enum_CallOption	返回看涨期权的枚举值
Enum_Canceled	返回委托状态的已撤单枚举值
Enum_Canceling	返回委托状态的正在撤单枚举值
Enum_Declare	返回委托状态的正在申报枚举值
Enum_Declared	返回委托状态的已申报枚举值
Enum_Deleted	返回委托状态的已废除枚举值
Enum_Entry	返回开平仓状态的开仓枚举值
Enum_EuropeanOption	返回欧式期权的枚举值
Enum_Exit	返回开平仓状态的平仓枚举值
Enum_ExitToday	返回开平仓状态的平今仓枚举值
Enum_Filled	返回委托状态的全部成交枚举值
Enum_FillPart	返回委托状态的部分成交枚举值
Enum_PutOption	返回看跌期权的枚举值
Enumu_Sell	返回买卖状态的卖出枚举值

示例：

```
A_SendOrder(Enum_Sell, Enum_Exit, data0.F_BuyRemainPosition(), data0.Price("bid1")); //
对当前公式应用的交易账户对应合约发出卖出平仓委托，以买一价平仓公式应用交易账户当前多头
可用持仓
```

```
F_GetLastOpenOrderIndex(Enum_Buy, Enum_Entry); //返回公式应用对应的交易帐户中当前合
约最后一个买开委托的未成交委托单索引
```

## 四、期货量化回测详解

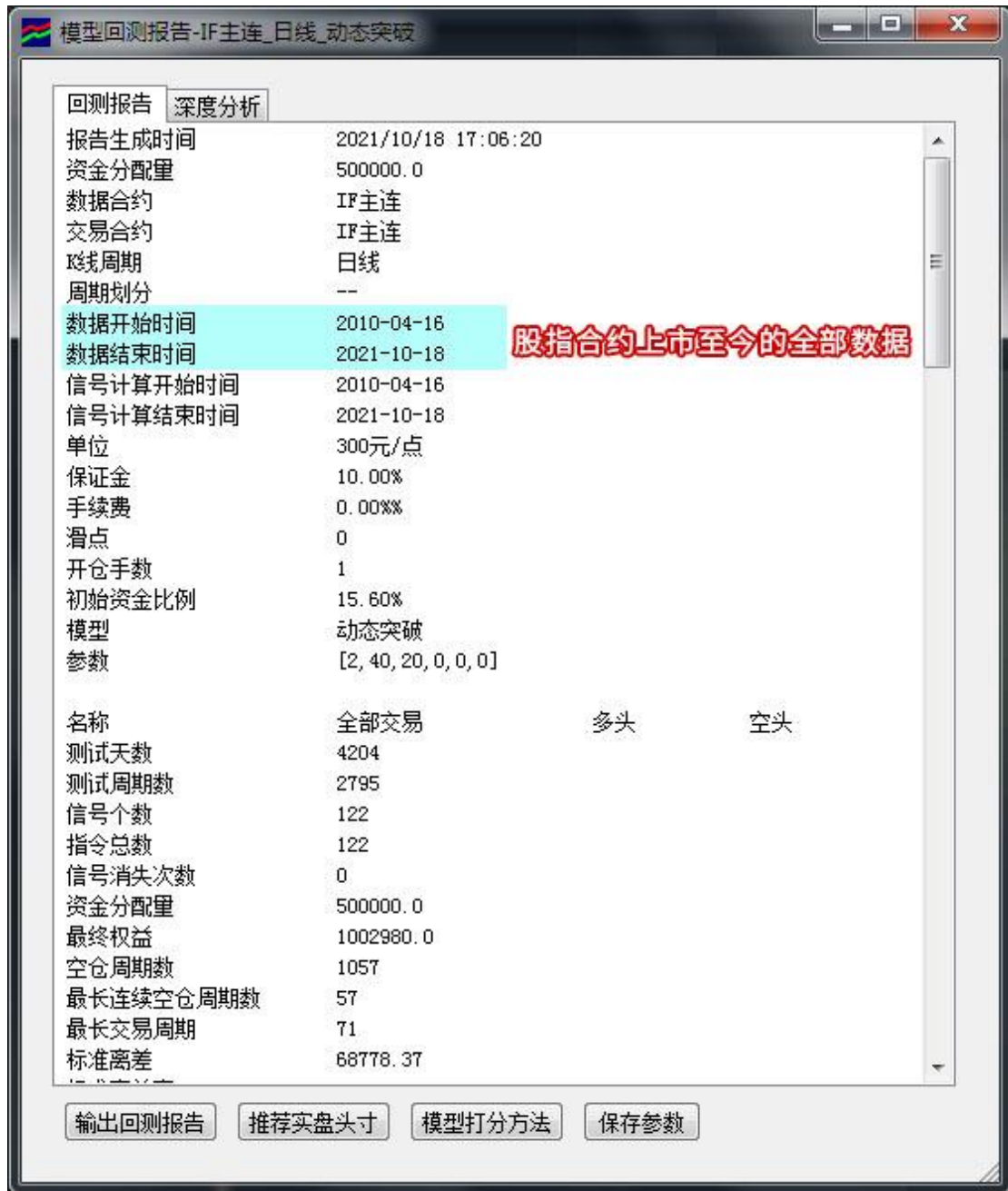
### （一）单合约回测

#### 1、海量数据回测模型

模型编写好后需要先对其进行效果测试，如果样本数据少，数据不精确，得到的测试结果就很难反映出模型的真实盈利效果。因此历史数据量和数据粒度决定了模型效果的测试深度，也反映出测试平台价值的高低。

##### （1）案例：国内合约提供从开市至今的全部历史数据

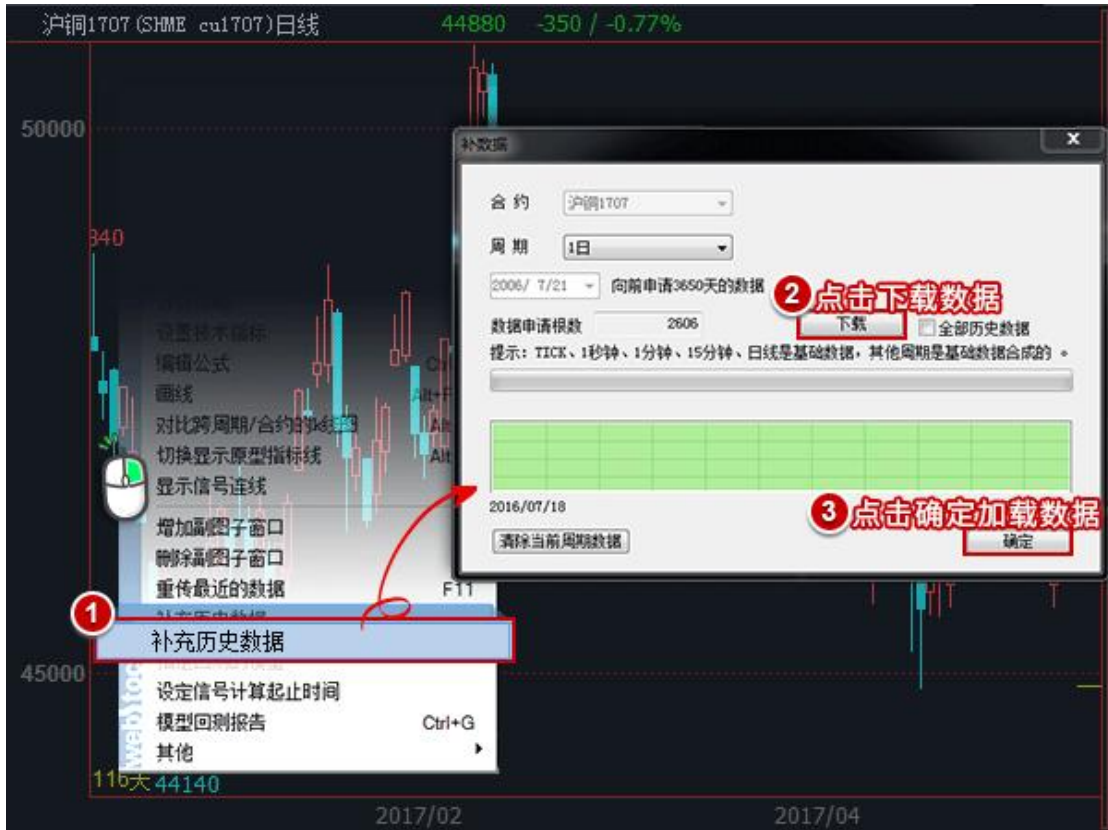
检验趋势模型的效果，通常需要选择连续且全面的历史数据作为测试样本，如果测试的数据不够多，无法包含市场各个阶段的行情变动，模型效果真实性就会大打折扣。T9 提供合约从上市以来的所有数据，可以多维度剖析模型。如下图，是以股指主连合约上市至今的全部数据进行的效果测试。



## (2) 申请海量历史数据的操作方法

如下图所示，是如何在主图点击鼠标右键来申请合约数据：





### 相关问题解答：

#### 1、因为网络断线或其他原因，存在错误的数据，如何将错误数据更正？

答：当您发现 K 线数据存在错误时，可通过 k 线图右键->补充历史数据，在弹出界面选择清除当前周期数据，这样错误数据就会被删除，然后再重新补充数据。

#### 2、为什么在【补充数据】中找不到想补充的合约周期。

答：加载列表中所列出的合约周期是基础数据；其他周期的数据由这些基础数据合成，因此，在申请数据的时候只对基础数据做申请。

#### 合成原则：

用 tick 合成的数据：逐笔回测数据。

用 1 秒合成的数据：量能周期、秒周期，自定义秒周期。

用 1 分钟合成的数据：15min 以下的周期，以及自定义分钟周期。

用 15 分钟合成的周期：15min, 30min, 1h, 2h, 3h, 4h, 自定义小时周期。

用日线合成的周期：日线及以上周期

### 3、哪些周期需要补充数据？

答：理论上来说，做回测都需要提前补充数据。但考虑到逐笔 TICK 类数据量大，手动补充时间长，为提高操作效率，模型中用到 MultSig 逐笔 TICK 函数时，回测的时候软件会在后台自动补充数据，不需要手动补充。除此之外，其他的模型都需要提前补充好回测需要的基础数据。

### 4、滑点是什么意思，设置多少合适？

答：滑点是指委托价格与成交价格之间的差值。

回测参数中设置滑点，目的是为了让回测效果最大限度拟合实盘的成交情况，具体设置多大的滑点要根据交易的合约和委托方式调整。

例如，以对价方式发委托，可以设置 1-2 个滑点，使回测结果尽可能的贴合品种行情特征，更接近实盘效果。

### 5、为什么已经补充了数据，但是选择信号计算起始时间不能选择到补充的时间？

答：回测之前需要将数据合约和交易合约的数据都补充一致再回测。

例如，模型中含有 Trade\_Other:Auto; 函数指定了交易合约再加载到加权合约上测试时，数据合约是加权合约，交易合约是主连，补充了加权合约数据之后，还需要同步补充主连合约的数据再进行回测，如下图操作。



### 6、为什么无法补充全部的数据，在补充数据中只能看到最近几个月？

答：这种情况是因为回测合约选择的是具体月份合约，所以只能补充到自合约上市以来的数

据。如果需要使用连续的历史数据进行回测，建议使用加权合约。

如：沪铜 1805 是 2017 年 5 月上市交易，所以只能申请到 2017 年 5 月上市以来的数据，而沪铜加权则支持 1996 年品种上市以来全部数据。

#### 7. 如何查看 RB1505 等已交割合约的数据？

答：在 K 线图界面右键->选择历史年度合约，可以查看到历史年度合约的具体行情数据。

## 2、回测报告分析检测模型好坏

模型回测需要计算每一笔历史交易的盈亏、回撤等进而得到收益率、最大回撤、胜率等我们关注的指标数据，这些数据如果全靠手动计算几乎是实现不了的。量化交易软件充分利用了计算机强大的运算能力，可瞬间出具含有众多参考项目的详细报告，分分钟就能检验模型好坏。

### (1) 案例 1：利用分析报告 360 度检测模型

下图是一个分析报告的一部分，图中①④指标是我们通常最关注的的数据，很多人会认为收益率高，胜率高的模型就是好模型，真的是这样的么？让我们再来看看图中的②③⑤衡量指标。

回测报告		深度分析	
回测期间最大权益	1950520		
回测期间最小权益	472710		
最终权益	1539200		
收益	1039200	1265190	-225990
<b>收益率</b>	<b>207.84%</b>	<b>253.04%</b>	<b>-45.20%</b>
年化单利收益率	35.35%		
月化单利收益率	2.91%		
年化复利收益率	21.08%		
月化复利收益率	1.58%		
平均保证金收益率	-0.76%	5.40%	-9.14%
扣除最大盈利后收益率	18.88%	64.07%	-59.15%
扣除最大亏损后收益率	219.01%	263.27%	-34.03%
夏普比率	2.84		
索提诺比率	11.04		
权益离散度	144493.14		
权益离散系数	28.90%		
本金风险率	5.46%		
年化收益风险比率	6.48		
<b>权益最大回撤</b>	<b>1211300.00</b>		
权益最大回撤时间	2021/07/21		
<b>权益最大回撤比率</b>	<b>71.93%</b>		
权益最大回撤比率时间	2021/07/21		
权益最长未创新高周期数	926		
权益最长未创新高时间段	2017/12/11 - 2021/09/29		
损益最大回撤	835000.00		
损益最大回撤时间	2021/05/27		
损益最大回撤比率	63.23%		
损益最大回撤比率时间	2021/05/27		
损益最长未创新高周期数	893		
损益最长未创新高时间段	2017/12/15 - 2021/08/17		
<b>胜率</b>	<b>28.81%</b>	<b>26.47%</b>	<b>28.00%</b>
交易次数	59	34	25
盈利次数	16	9	7
亏损次数	42	24	18
持平次数	1	1	0
最大持续盈利次数	3	2	3
最大持续盈利次数出现时间	2016/03/23 - 2016/06/02		
最大持续亏损次数	18	15	11
最大持续亏损次数出现时间	2019/04/24 - 2020/10/26		
<b>盈亏比率</b>	<b>5.34</b>	<b>8.10</b>	<b>1.06</b>
平均每次盈利	127802.50	209592.22	22644.29
平均每次亏损	23943.81	25880.83	21361.11

报告显示该模型在这段交易过程中，最大回撤高达 1211300，最大回撤比已达到 71%，说明这并不是一个稳定的模型。如此大的回撤一方面会在实盘中带来权益的锐减，另一方面也会使交易者的心态受到严重的影响。再看⑤指标，空头交易的盈亏比率较低，说明该模型的空头交易绝大多数是亏损的，在实盘交易中该模型的做空操作很可能成为短板。

所以在衡量一个模型的好坏时，我们要充分利用软件提供的测算报告，根据各个测算指标全面考量模型。

## (2) 案例 2：效果测试结果的升级应用

回测报告中的各项统计数据也为我们拟合一些其他的经典理论提供了数据基础和计算依据，可以更深层次的挖掘模型的隐含特性，全方位剖析模型效果。

例如著名的凯利公式： $f^* = (b \cdot p - (1-p)) / b$ （其中  $f^*$  为开仓最佳资金比例、 $b$  为模型的盈亏比率、 $p$  为胜率）。我们从回测报告中可以直接得到胜率和盈亏比的数值，将这些数据代入到公式中进而得到该交易系统的最佳开仓比例，投资者可以根据自身交易系统所计算出的  $f^*$  值乘以心理承受损失的比例去控制最大仓位。

损益最长未创新高周期数	1040		
损益最长未创新高时间段	2017/08/08 - 2021/11/18		
<b>胜率</b>	<b>37.04%</b>	28.57%	35.90%
交易次数	81	42	39
盈利次数	26	12	14
亏损次数		3	3
持平次数			
最大持续盈利次数	3	3	3
最大持续盈利次数出现时间	2015/03/14 - 2016/06/06		
最大持续亏损次数		7	5
最大持续亏损次数出现时间	2018/12/21 - 2019/07/12		
<b>盈亏比率</b>	<b>3.15</b>	5.09	1.89
平均每次盈利	219953.46	284644.17	164504.29
平均每次亏损	69854.71	55913.21	86826.96
平均每次盈利/权益最大回撤	0.16		
平均每次盈利率/平均每次亏损率	4.48		
平均每次收益/平均每次亏损	0.38		
平均每次收益	26619.75	44051.43	7847.18
最大每手盈利	7778.68		
最大每手亏损	3060.00		

开仓的最佳资金比例 =  $(b \cdot p - (1-p)) / b$

=  $(3.15 \cdot 0.3704 - 0.6296) / 3.15$

= 17.05%

b: 盈亏比率    p: 胜率

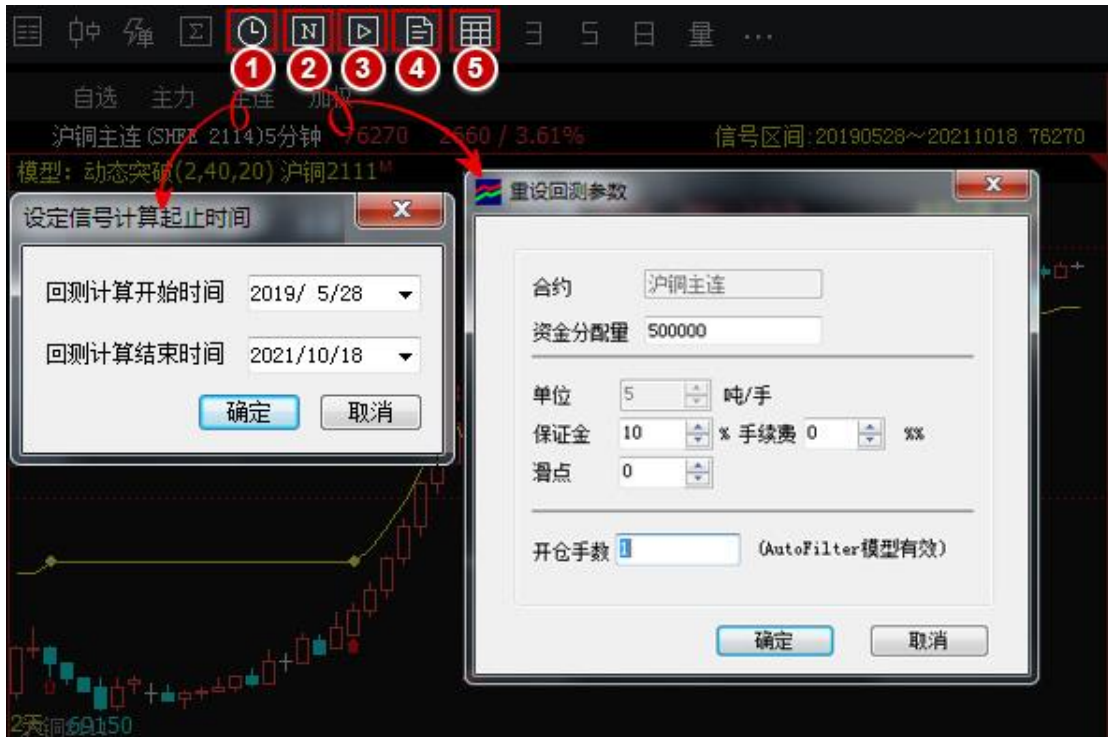
## (3) 进行收益率测算的操作方法






如下图所示是如何进行模型的回测。



## 相关问题解答:

1、如何修改回测时初始资金等相关参数?



- ① : 设定信号计算起止时间。
- ② : 设置回测参数，如初始资金，开仓手数，保证金，手续费等。
- ③ : 装入到模组后台运行，将 k 线图上回测的合约和模型加载到模组中运行。
- ④ : 查看回测报告
- ⑤ : 查看信号明细

## 2、T9 是否支持逐笔回测？

答：支持；模型中写入 MultSig 函数时，支持逐笔回测。

## 3、效果测试指标项说明：

注：

对于盈利、亏损的计算，带有\*的是都是以从开仓到持仓为 0 算一次交易计算盈亏，未标注\*的是按照一开一平来计算盈亏。

测试天数	从回测数据开始到结束的自然日
------	----------------

测试周期数	从回测数据开始到结束的 K 线根数
信号个数	信号出现的总个数
执行信号个数	信号执行的总个数
信号消失次数	信号消失的总次数
初始资金	初始化的资金
初始资金使用率	手数*每手保证金率*第一个信号的开仓价*交易单位/初始资金
最大资金使用率	在持仓周期内(开仓保证金/当前权益)的最大值
平均资金使用率	((开仓保证金/当前权益) 求和)/持仓周期数
杠杆倍数	杠杆倍数 = 最大资金使用率 / 保证金率
回测期间最大权益	回测期间最大的权益 整个测试过程中每个周期已缴保证金+剩余可用资金+持仓浮盈所得结果中的最大值
回测期间最小权益	回测期间最小的权益 整个测试过程中每个周期已缴保证金+剩余可用资金+持仓浮盈所得结果中的最小值
最终权益	包括当前的可用资金和浮动盈亏
收益	总盈利-总亏损（已扣除手续费）
收益率	收益/初始资金
年化单利收益率	收益率/（测试天数/365）



月化单利收益率	收益率/（测试天数/30）
年化复利收益率	$(\text{最终权益}/\text{初始资金})^{(365/\text{测试天数})} - 1$
月化复利收益率	$(\text{最终权益}/\text{初始资金})^{(30/\text{测试天数})} - 1$
平均保证金收益率	（单次交易收益/开仓保证金）求和/交易次数 $= (\sum ((\text{每次交易盈利数值} - \text{手续费}) / \text{保证金})) / \text{交易次数}$
扣除最大盈利后收益率	$(\text{最终权益} - \text{最大盈利} - \text{初始资金}) / \text{初始资金}$
扣除最大亏损后收益率	$(\text{最终权益} + \text{最大亏损} - \text{初始资金}) / \text{初始资金}$
夏普比率	<p><math>(\text{年化单利收益率} - \text{无风险利率}) / \text{收益率的标准离差率}</math></p> <p>计算公式：夏普比率 = <math>[E(R_p) - R_f] / \sigma_p</math>;</p> <p>其中：</p> <p><math>E(R_p)</math>：年化单利收益率</p> <p><math>R_f</math>：无风险利率（大约是 1.5%）</p> <p><math>\sigma_p</math>：收益率的标准差率（年化标准差率）= 标准差率 / <math>\sqrt{\text{测试天数} / 365}</math></p> <p>标准差率 = 标准离差 / 初始资金</p>
索提诺比率	<p><math>(\text{年化单利收益率} - \text{无风险利率}) / \text{下行收益率标准差}</math></p> <p>其中：</p> <p>年化单利收益率 = 收益率 / （测试天数 / 365）</p> <p>无风险利率 = 1.5%</p> <p>下行收益率标准差 = 下行标准差率 / <math>\sqrt{\text{测试天数} / 365}</math></p> <p>下行标准差率 = 下行标准离差 / 初始资金</p>

	$\text{下行标准离差} = \sqrt{\frac{1}{n} \sum_{i=1}^n (r_i - \bar{r})^2 f(t)}$ <p>n=交易次数</p> <p>r=每次收益</p> <p><math>\bar{r}</math>=平均收益</p> <p>if <math>r &lt; \bar{r}</math>, <math>f(t)=1</math></p> <p>if <math>r &gt; \bar{r}</math>, <math>f(t)=0</math></p>
<b>权益离散度</b>	权益离散度=单次盈亏的平方和/交易次数，再开方
<b>权益离散系数</b>	权益离散系数=权益离散度/初始资金*100%
<b>权益最大回撤</b>	从回测开始到结束，动态权益计算出来的波段从高点到低点回撤的最大值
<b>权益最大回撤时间</b>	权益最大回撤出现的时间
<b>权益最大回撤比率</b>	(权益回撤/权益回撤时的权益) 的最大值
<b>权益最大回撤比率时间</b>	权益最大回撤比率出现的时间
<b>权益最长未创新高周期数</b>	权益最长未创新高周期数
<b>权益最长未创新高时间段</b>	权益最长未创新高时间段
<b>损益最大回撤</b>	从回测开始到结束，动态损益计算出来的波段从高点到低点回撤的最大值（损益最大回撤是以持仓等于 0 时的资金为标准计算的）

损益最大回撤时间	损益最大回撤出现的时间
损益最大回撤比率	(损益回撤/损益回撤时的权益)的最大值
损益最大回撤比率时间	损益最大回撤比率出现的时间
损益最长未创新高周期数	损益最长未创新高周期数
损益最长未创新高时间段	损益最长未创新高时间段
本金风险率	(初始资金 - 回测期间最小权益) / 初始资金
年化收益风险比率	年化单利收益率/本金风险率
胜率	(盈利次数+持平次数) / 交易次数
*盈亏比率	盈利交易的平均每次盈利/亏损交易的平均每次亏损 平均每次盈利 = 总盈利/盈利交易次数 平均每次亏损 = 总亏损/亏损交易次数
*平均每次盈利	平均每次盈利交易的盈利=总盈利/总盈利次数 (计算手续费)
*平均每次亏损	平均每次亏损交易的亏损=总亏损/总亏损次数 (计算手续费)
*平均每次盈利/权益最大回撤	平均每次盈利/权益最大回撤
*平均每次盈利率/平均每次亏损率	平均每次盈利率/平均每次亏损率 平均每次盈利率 = SUM (单次盈利率) / 计算单次盈利率次数 平均每次亏损率 = SUM (单次亏损率) / 计算单次亏损率次数 单次盈利率 = 上一次持仓为0到今次持仓为0期间的盈利占初始资金的

	百分比（盈利/初始资金） 单次亏损率 = 上一次持仓为0到今次持仓为0期间的亏损占初始资金的百分比（亏损/初始资金）
<b>*平均每次收益/平均每次亏损</b>	平均每次收益/平均每次亏损
<b>平均每次收益</b>	平均每笔交易的收益 = (总盈利 - 总亏损) / 交易笔数 (总成交量的 1/2)
<b>最大每手盈利</b>	每手盈利的最大值 (每手盈利: 对于每次交易, 用该次交易的盈利值除以这次交易过程中的成交手数)
<b>最大每手亏损</b>	每手亏损的最大值 (每手亏损: 对于每次交易, 用该次交易的亏损值除以这次交易过程中的成交手数)
<b>平均每手每次收益</b>	平均每手每次交易的收益
<b>总盈利/总亏损</b>	总盈利/总亏损
<b>总盈利</b>	盈利的总和
<b>总亏损</b>	亏损的总和
<b>*单次最大盈利</b>	单次交易盈利的最大值
<b>*单次最大亏损</b>	单次交易亏损的最大值
<b>单次最大盈利占比率</b>	单次最大盈利/总盈利
<b>单次最大亏损占比率</b>	单次最大亏损/总亏损
<b>收益/单次最大盈利</b>	收益/单次最大盈利

收益/单次最大亏损	收益/单次最大亏损
*交易次数	发生交易的次数
*盈利次数	盈利的交易次数
*亏损次数	亏损的交易次数
*持平次数	持平的交易次数
*最大持续盈利次数	最大持续盈利的次数
*最大持续盈利次数 出现时间	最大持续盈利的次数出现的时间
*最大持续亏损次数	最大持续亏损的次数
*最大持续亏损次数 出现时间	最大持续亏损的次数出现的时间
空仓周期数	空仓的周期数
最长连续空仓周期 数	最长连续空仓的周期数
*最长持仓周期数	持仓最长周期数
平均持仓周期数	平均多少根 K 线发生一笔交易=测试周期数/交易次数
*平均盈利持仓周期 数	平均多少根 K 线发生一笔盈利的交易=测试周期数/盈利次数
*平均亏损持仓周期 数	平均多少根 K 线发生一笔亏损的交易=测试周期数/亏损次数

平均持仓手数	每根 K 线的总持仓的累加 / 持仓周期数
最大持仓手数	在持仓周期内持仓手数最大值
平均资金使用金额	开仓保证金求和/持仓周期数
最大资金使用金额	在持仓周期内，开仓保证金最大值
最大资金使用金额时间	在持仓周期内，最大资金使用金额的时间点
手续费	手续费合计
手续费/收益	手续费合计/收益
滑点损耗	滑点损耗合计
滑点损耗/收益	滑点损耗合计/收益
成交额	成交额合计 成交额=成交价*（开仓或者平仓手数）*交易单位
模型得分	根据盈利能力、稳定性、风险暴露、敏感度计算的综合得分

## 6、深度分析各图表项说明（点击蓝色文字查看说明）

收益/风险：

- (1) [损益曲线图](#)
- (2) [权益曲线图](#)
- (3) 损盈分布图
- (4) 损盈直方图
- (5) 权益回撤直方图

(6) 连续亏损次数

浮动盈亏：

(1) 权益面积图

(2) 浮盈面积图

(3) 浮亏面积图

(4) 浮盈分布图

(5) 浮亏分布图

(6) 浮盈直方图

(7) 浮亏直方图

阶段统计：

(1) 月度统计

(2) 日统计

效率分析：

(1) **总效率分布图**

(2) 建仓效率分布图

(3) 平仓效率分布图

损盈分析：

(1) **胜率分析**

(2) 盈利分析

(3) 亏损分析

(4) **回撤分析**

### 3、敏感性测试了解模型脾性

模型的效果受很多因素影响，如滑点、手续费、杠杆倍数等，我们将模型的这种特性称之为模型的敏感性。敏感性小的模型是更稳定、更好的模型。

将模型的敏感性绘制成图表，就可更直观的分析出模型的优劣。例如，在分析滑点对模型收益率的影响时，用横坐标表示滑点，纵坐标表示收益率，很明显滑点对收益率影响小的

模型是更优的模型。

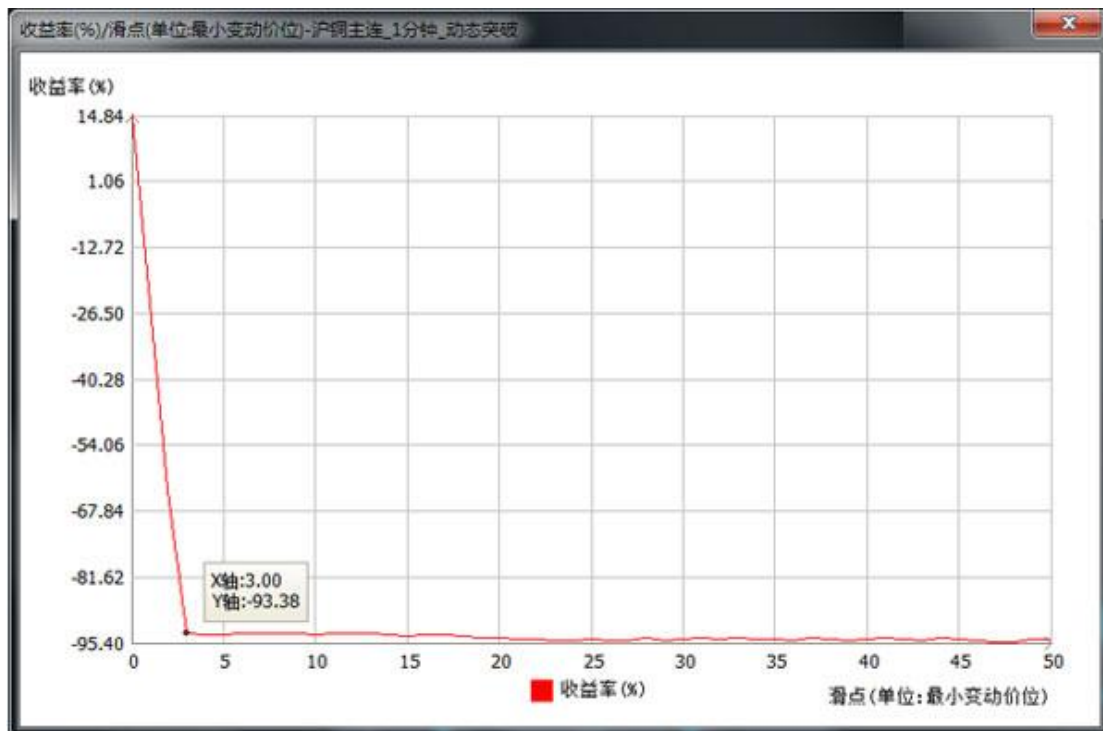
### (1) 案例：盈利的模型就是好模型么？

如下图所示，这是一个盈利的模型，也许 40.92%的胜率和 8.44%的最大回撤会让我们觉得这个模型还算可以，但这一定是一个好模型么？

测试天数	1583		
测试周期数	78492		
指令总数	5504		
信号消失次数	0		
初始资金	500000.00		
最终权益	714690.00		
空仓周期数	42216		
最长连续空仓周期数	125		
最长交易周期	92		
标准离差	2154.32		
标准离差率	27.62		
夏普比率	33.35		
盈亏总平均/亏损平均	0.08		
最大回撤	52690.00		
最大回撤时间	2011/06/22 09:00		
<b>最大回撤比</b>	<b>8.44%</b>		
最大回撤比时间	2010/11/16 11:09		
风险率	1.99%		
收益率/风险率	4.97		
每手最大回撤	20260.00		
每手平均盈亏	78.01		
盈利率	42.94%	17.82%	25.
年化单利收益率	9.90%		
月化单利收益率	0.81%		
年化复利收益率	8.59%		
月化复利收益率	0.68%		
<b>胜率</b>	<b>40.92%</b>		
风险收益评级	496.51%		
平均盈利/最大回撤	0.03		
平均盈利/平均亏损	1.67	1.65	1.6
净利润	214690.00	89080.00	125
总盈利	1867360.00	980990.00	886
总亏损	1652670.00	891910.00	760
总盈利/总亏损	1.13	1.10	1.1
其中持仓净盈	150.00	0.00	150
交易次数	2752.00	1484.00	126
盈利比率	0.40	0.40	0.4

让我们从一个二维关系去审视这个模型，看一下滑点和收益之间的关系：从下图可很明显的看出，滑点每增大一点，收益率就会大幅下降；当滑点为 3 个最小变动价位的时候模型已经失去了盈利能力。而实际交易的时候，我们很难做到 0 滑点，这个模型也就很难盈利。





分析报告得到的是各衡量指标的独立数据，而敏感性测试可告诉我们一些重要指标的变化对盈利、胜率等的影响，让我们深度了解模型的脾性。

敏感性分析横坐标提供手续费、滑点、开仓手数、杠杆倍数以及模型参数等影响因素；纵坐标提供收益率、胜率、平均收益、盈亏比率、平均每次收益/权益最大回撤、权益最大回撤比率、总交易次数和权益最大回撤指标。投资者可以根据策略的分析侧重点自由搭配组合，测算模型的敏感性。

## (2) 进行敏感性测试的操作方法

如下图所示，是如何进行敏感性测试。



#### 4、参数优化让模型达到最优

交易过程中有时会发现一段时间内表现很好的模型，过了一段时间就好像失效了一样，这种情况是由于模型参数不再适应当前行情引起的，我们需要尽快寻找新的最优参数，而在海量的历史数据中仅凭人工去寻找如大海捞针，费时费力，机会渺茫。“参数优化”，可在指定的参数范围内用计算机很快筛选出最适合当前行情的参数。

(1) 案例：利用参数优化，让止损参数顺势而为

下图是对螺纹品种跟踪止损策略模型的回测资金曲线。结束了小半年的单边上涨行情后，市场极速下跌进入调整形态，从图上可清楚的看到资金曲线在近下半年不再保持稳定上升形态，说明原来的止损价差参数已经不能适应现在的市场，模型已经失效。我们可以利用参数优化功能，对当前的参数进行优化，筛选出符合当前行情的参数组合。



下图是利用参数优化对模型的参数进行枚举和遗传后的结果。参数优化功能可以在指定的范围内利用计算机筛选出最适合当前行情的参数。在使用新参数后，黄色资金曲线更平滑稳定，新的参数更能适应市场行情，使模型真正达到顺势而为的效果。



## (2) 进行参数优化的操作步骤

① 先使用枚举功能，进行大范围海选，如下图所示是如何进行枚举：



设置参数关系减少参数优化时间：

有些模型各个参数间有严格的逻辑管理，以下面的回测模型为例：

Params

Numeric Length1(5);

Numeric Length2(10);

Vars

Numeric MA5;

Numeric MA10;

Begin

MA5=Ma(Close, Length1);

MA10=Ma(Close, Length2);

If(CrossUp(MA5, MA10))

BK;

If(CrossDown(MA5, MA10))

SP;

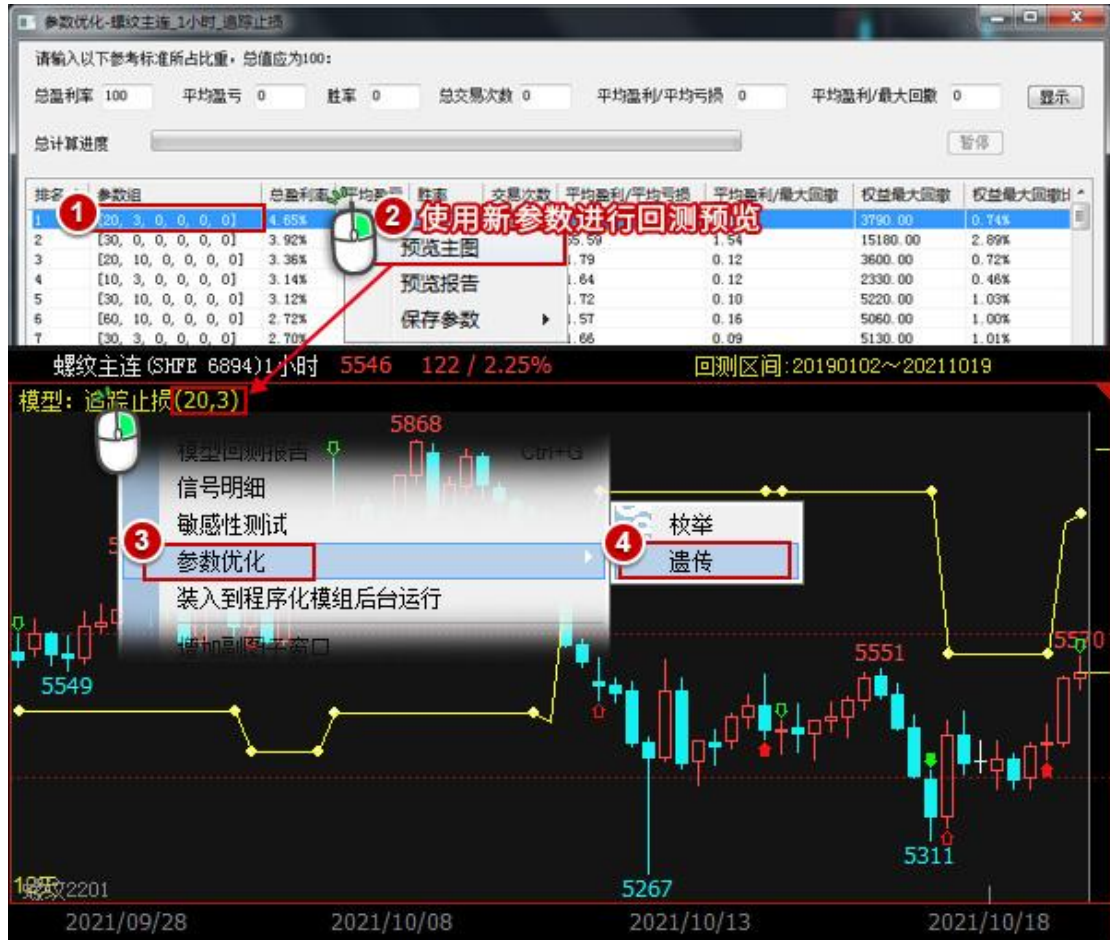
End

模型中的两个变量必然遵循一大一小规则,这样的模型我们可以首先为他们设置参数关系,如  $N1 < N2$ , 这样在优化时,所有比  $N2$  大的  $N1$  值都不用参加计算,运算量可以减少  $1/2$ . 这样每配置一个参数关系,计算次数就减少一半;配置 4 个,就减少到原来的  $1/16$ ,以此类推,有效减少了参数优化时间。

- ② 参数优化计算完会以排序的方式显示优化出来的参数组,可按下图步骤将优化后的参数保存至备选参数组或直接替换为默认参数组。



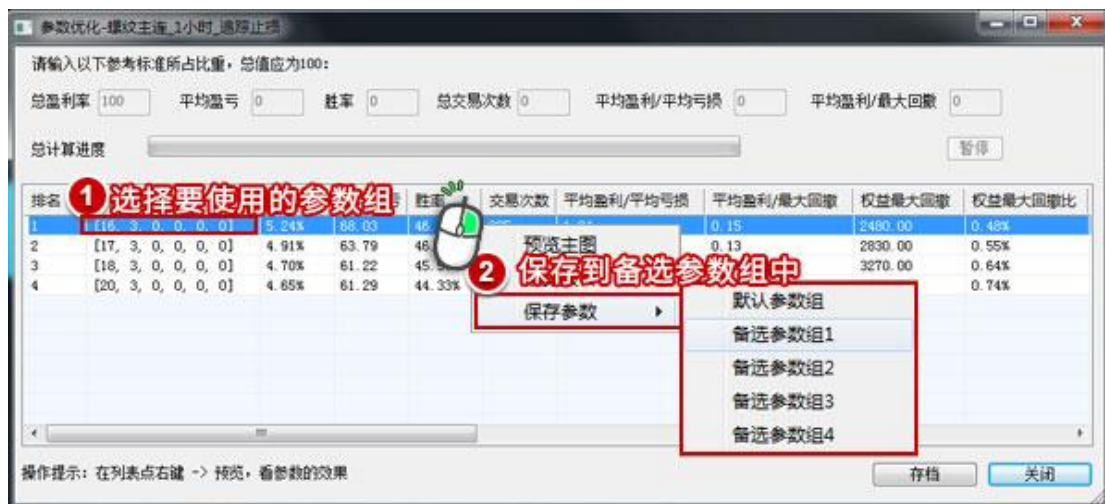
- ③ 如需对枚举的结果进一步精调,可在参数优化列表选择要优化的参数组->预览主图,如下图,用枚举出来的新参数对 K 线图回测,再对新参数进行遗传。



④ 如下图所示设置好优化的精调范围，参考标准的比重后，点击“确定”按钮，开始进行精调。



⑤ 按照和枚举同样的方法来保存精调后的结果，以便进行回测（如下图所示）。



⑥ 如下图，在 K 线图右键菜单->设置技术指标/量化模型->选择模型->调入参数，在参数界面选择参数优化后保存的参数组，使用新参数进行回测。





## 相关相关问题解答:

### 1、为什么有“枚举”和“遗传”两种参数优化方式，原理是什么？

答：枚举是在每个参数最小值与最大值之间抽选几个效果最好的参数值，是对参数的初步海选；遗传是在枚举好的参数值基础上进行微调，让参数达到最优。

#### (1) 枚举：

根据设置的“步长”挑选出所有满足条件的参数，进行全排列，在其中挑选出最优参数值。

#### (2) 遗传：

使用枚举得到的参数组，按步长 1 上下变动每一个参数，分别与剩余参数组合找到最优参数组。

### 2、参数优化页面上的参考标准表示什么含义？

答：参考标准是用于投资者在参数优化后，根据个人的分析需求对各项指标按权重划分，对优选出的参数进行排序显示。权重数值大小根据个人偏好来设置，可以对关注度高的指标设置大一点的权重。

例如，设置总盈利率权重为 60，平均盈亏为 40，优化后的参数组将按照总盈利率排名\*60%+平均盈亏排名\*40%的加权排名来排序显示。



参数优化

螺纹钢指数\_5分钟\_BOLL

请输入以下参考标准所占比重，总值应为100:

总盈利率 60 平均盈亏 40 胜率 0 总交易次数 0 平均盈利/平均亏损 0 平均盈利/最大回撤 0

总计算进度

排名	参数组	总盈利率	平均盈亏	胜率	交易次数	平均盈利/平均亏损	平均盈利
1	[25, 11, 2, 0, 0, 0]	0.12%	16.22	37.84%	37	2.71	0.18
2	[25, 26, 2, 0, 0, 0]	0.10%	17.50	39.29%	28	2.37	0.20
3	[25, 11, 6, 0, 0, 0]	0.08%	100.00	50.00%	4	2.43	0.57
4	[26, 11, 2, 0, 0, 0]	0.11%	14.32	40.54%	37	2.28	0.15
5	[26, 26, 2, 0, 0, 0]	0.09%	16.43	39.29%	28	2.33	0.19
6	[26, 11, 6, 0, 0, 0]	0.07%	66.00	40.00%	5	2.88	0.50
7	[25, 21, 2, 0, 0, 0]	0.07%	12.00	36.67%	30	2.48	0.18
8	[26, 21, 2, 0, 0, 0]	0.04%	6.45	38.71%	31	2.07	0.16
9	[15, 26, 2, 0, 0, 0]	0.02%	3.00	30.00%	30	2.45	0.16
10	[15, 11, 6, 0, 0, 0]	0.00%	0.00	0.00%	0	0.00	0.00

操作提示：在列表点右键 -> 预览，看参数的效果

存档 关闭

## （二）多合约回测

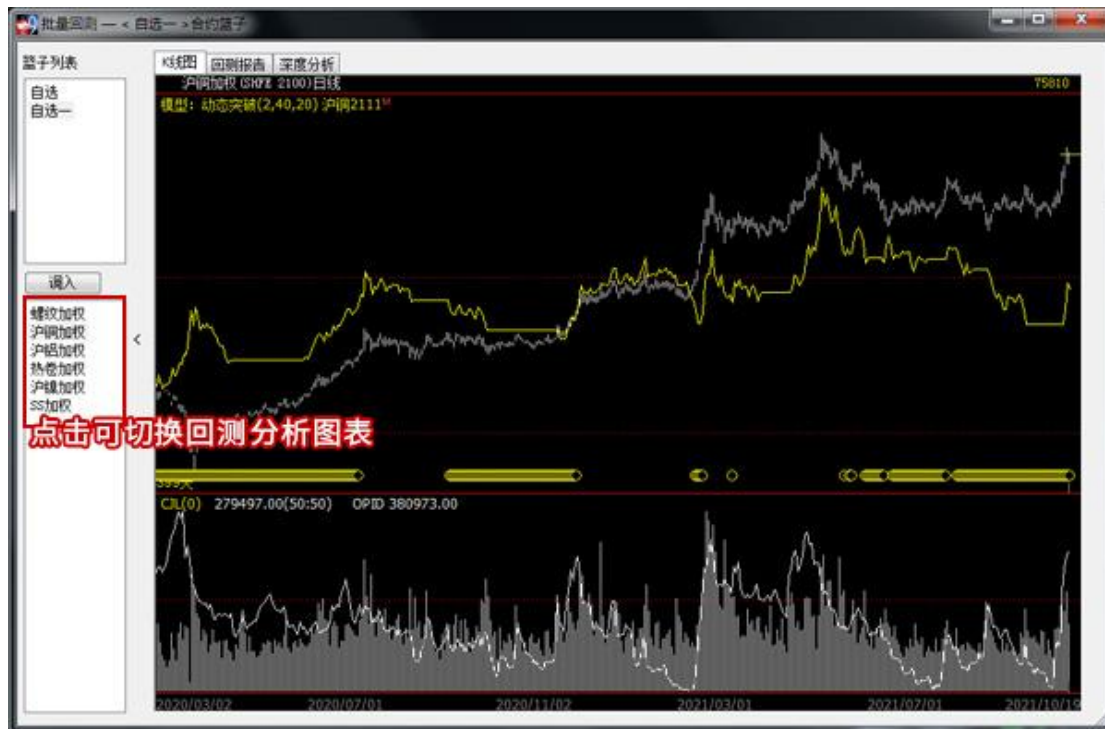
### 1、单模型多合约批量回测选最优合约

交易策略是不具备普适性的，我们很难构建出一个适用于所有合约的交易系统，但针对一个合约制定一个适合的模型则相对简单，同样的，针对一个模型也势必会存在一组最佳的交易合约。

软件的“批量回测”功能，可同步回测多个备选合约，快速出具各个合约的独立分析报告，投资者通过对比关注的收益指标便可快速挑选出符合策略的最佳合约。

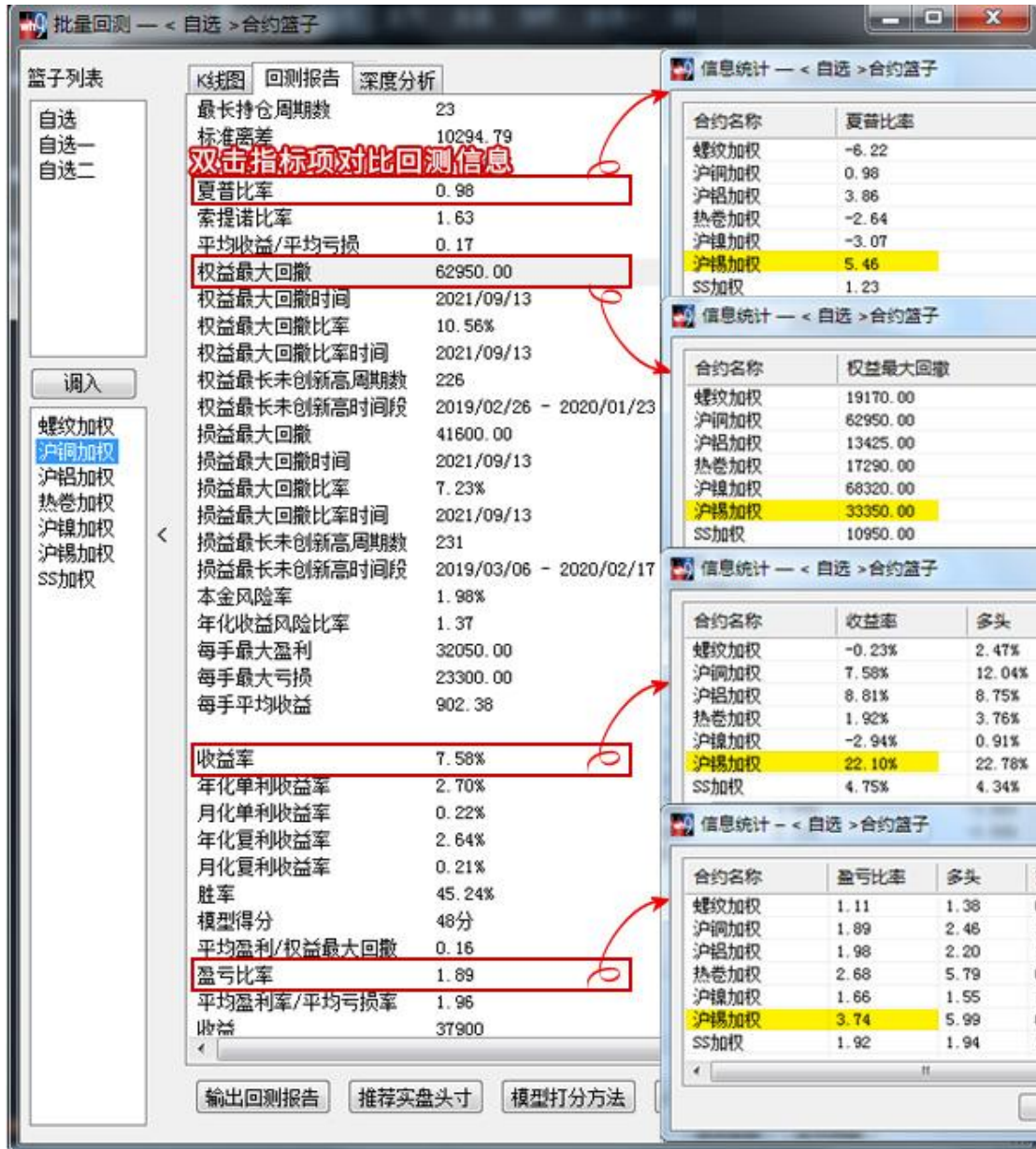
#### （1）案例：多合约同步回测，批量筛选最优合约

比如，用同一个策略模型对上期所部分加权合约进行批量回测，筛选最佳交易合约。回测后每一个合约都生成一份独立的回测报告和独立的资金曲线图，投资者点击合约名称可切换回测报告分析，极大程度的简化了操作步骤。

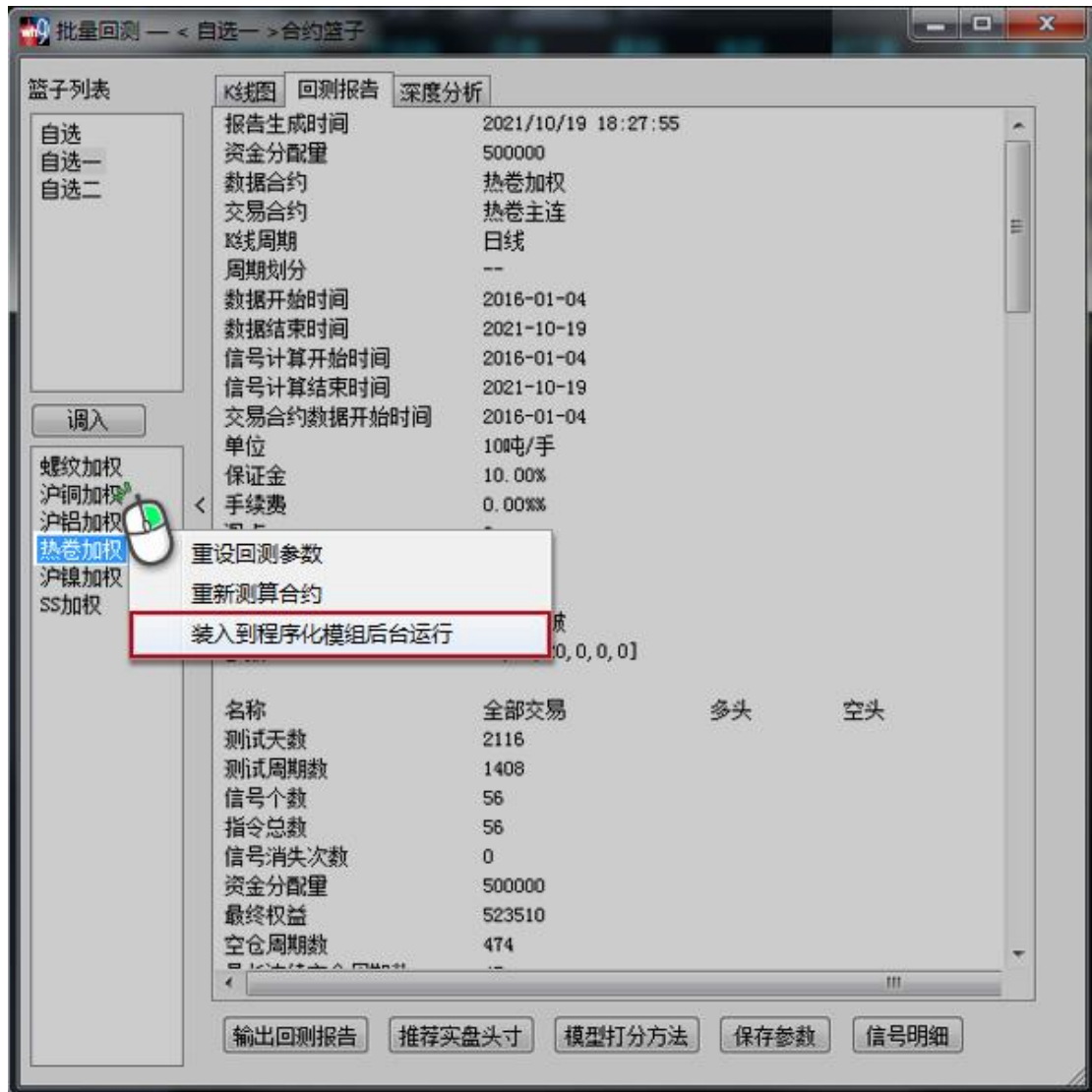


如上图，是沪铜合约的回测资金曲线，盈利效果还算不错，那它是否就是最当前策略的最佳交易合约呢？当我们展开各指标项进行横向对比时，可以看到沪铜合约盈利率虽高，但是权益最大回撤竟高达 62950，扣除掉最大盈利后的收益率已为负值，看来沪铜合约在测试期间的盈利并不稳定。

而沪锡收益率、盈亏比率、夏普比率等都是所有备选合约中最高的，回撤也是较小的，在测试阶段内一直保持着稳定的盈利效果，是当前策略最佳的交易合约。

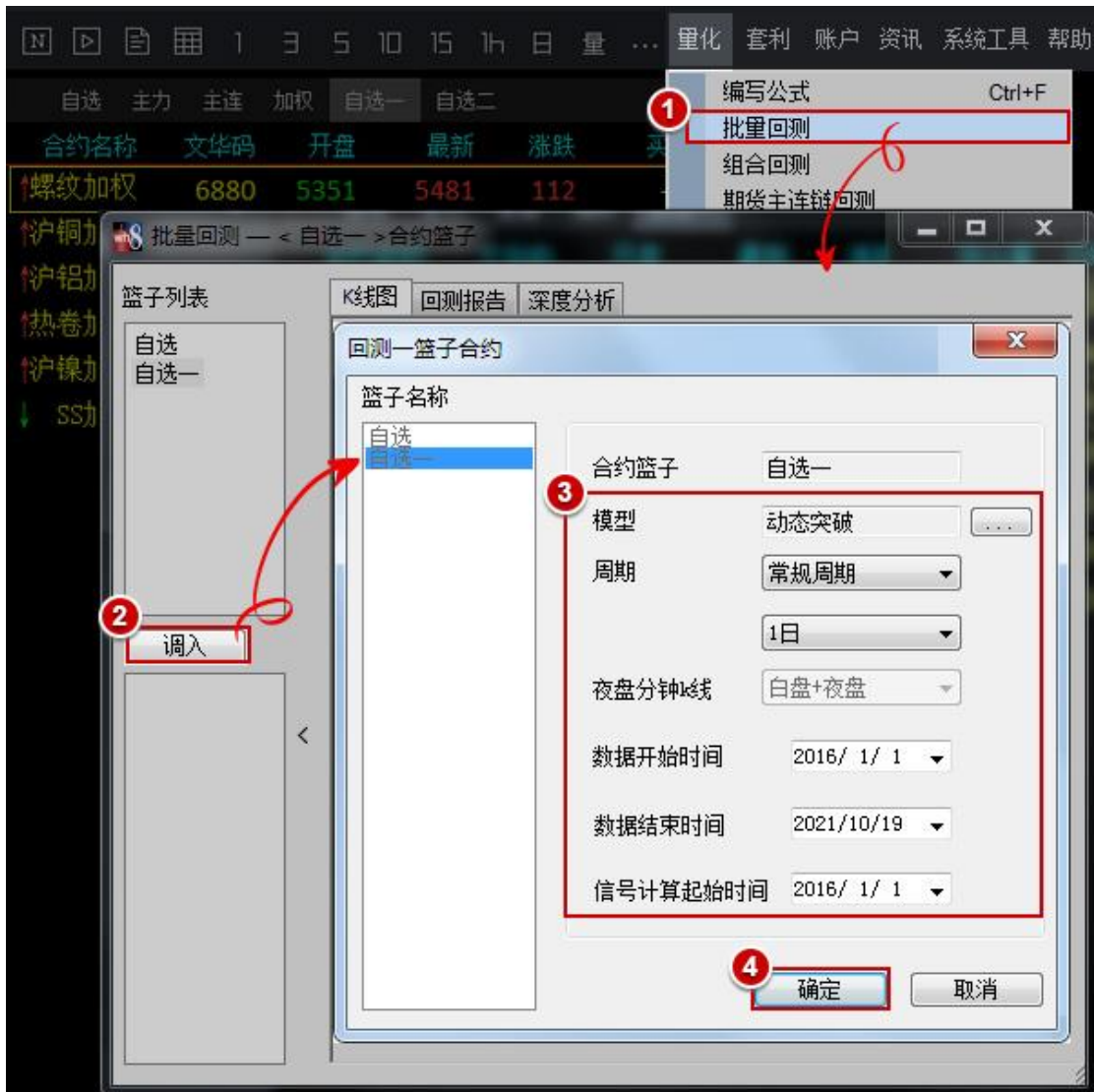


如下图，优选交易合约后，可以直接一键添加到模组实际运行。



## (2) 批量回测的操作步骤:

如下图，是批量回测的建立步骤。



## 2、多模型多合约组合回测分析整体盈利

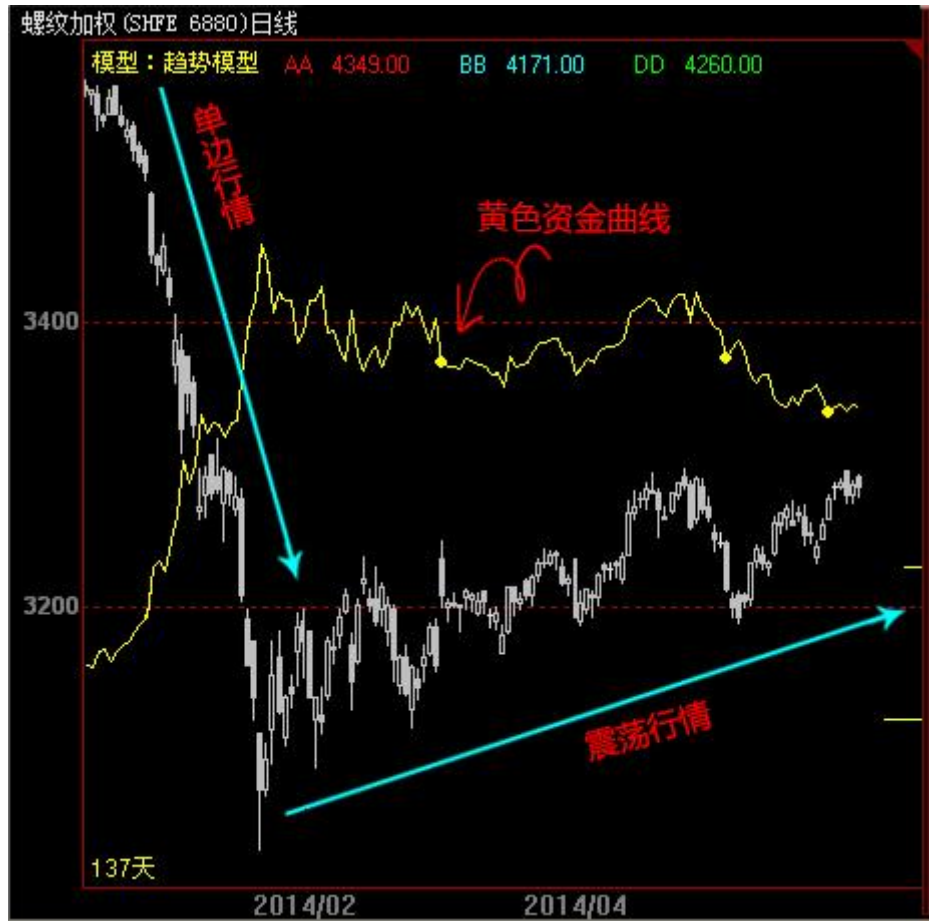
“不要把鸡蛋放到同一个篮子中”，同理，我们在交易时也要避免单一品种或单一策略的运行模式，通过多合约、多模型组合的形式来合理分散交易风险，实现共振盈利。

单模型的回测只能看到一个模型的收益效果，如何才能查看一个策略组合的收益效果呢？软件的组合回测功能可对多模型、多合约组合进行回测，提供组合分析报告、资金统计和各类资金曲线分析工具，供交易者全面分析策略组合和各成员的盈利能力。

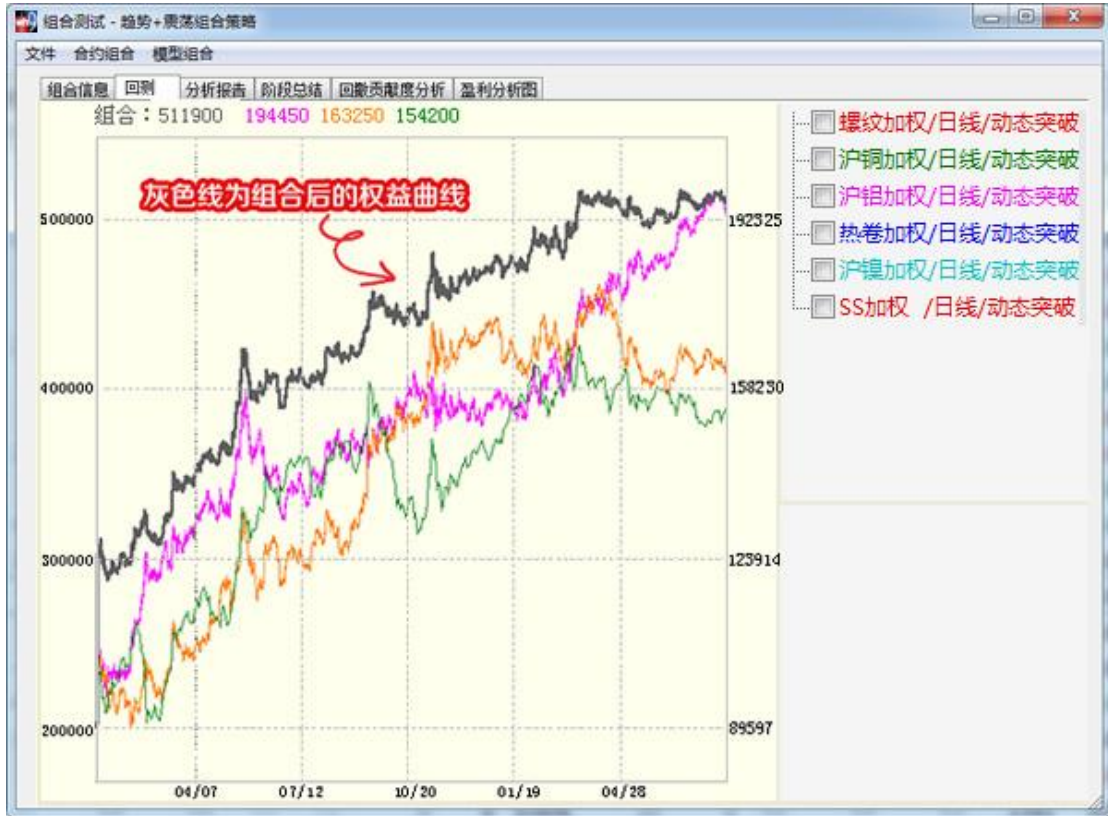
### (1) 案例：组合回测分析交易组合整体盈利

趋势行情下获利相对比较容易，趋势模型往往能够取得不错的回报，但在震荡行情下，

由于趋势策略不能适应频繁波动的行情，反而会使盈利的资金回吐甚至反盈为亏，如下图。



对此，我们可以使用多模型组合的形式，在一个合约上同时运行趋势策略和震荡策略。如下图，是螺纹合约“趋势+震荡+波段”策略组合交易的回测效果，彩色线为组合成员的权益曲线，灰色线为组合整体的权益曲线。可见，在震荡行情下，震荡策略的盈利冲抵了趋势模型的亏损；在趋势行情下交易组合则共振盈利，实现了财富的双重增长！

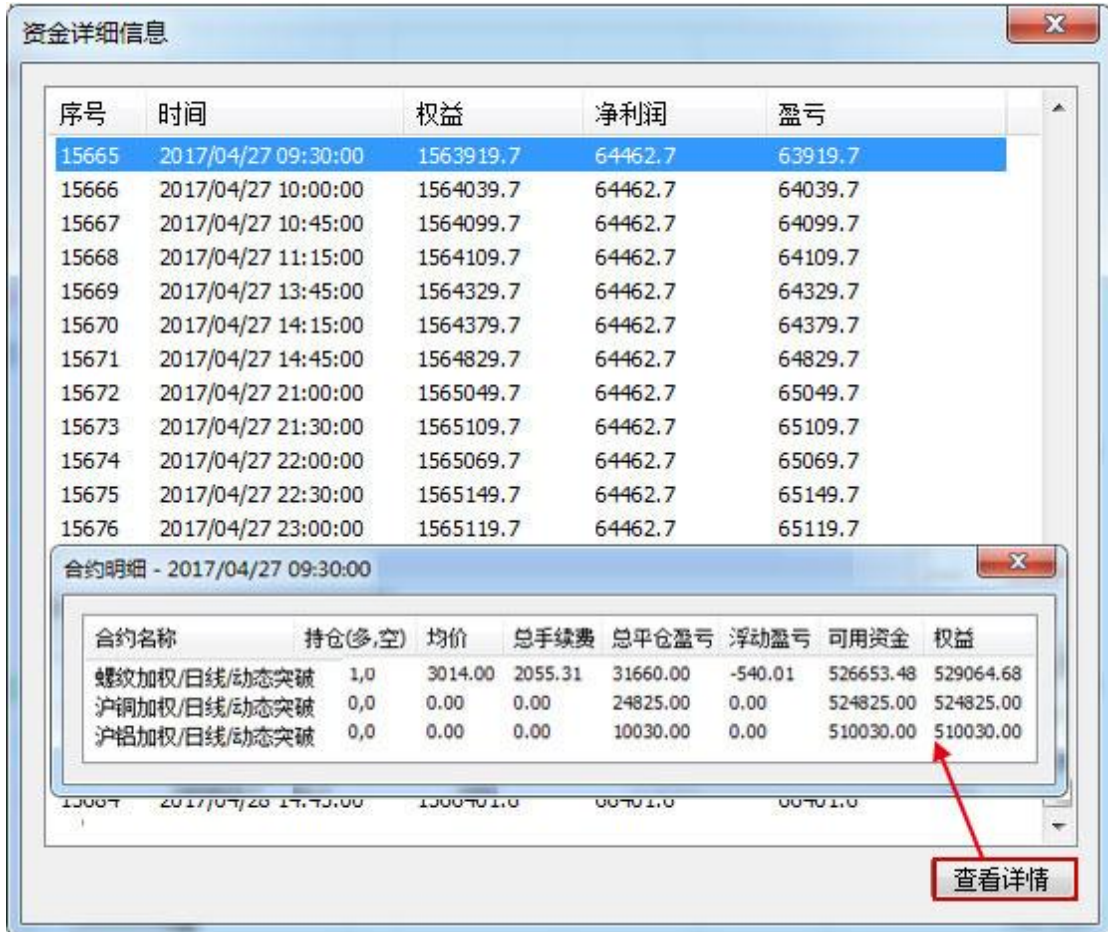


如下图，是组合回测的资金曲线分析界面。右键菜单中还提供组合的权益资金曲线、净利润曲线、盈亏曲线以及资金详细信息等统计工具，帮助投资者深层分析组合的收益效果。





- 权益资金曲线，记录每根 K 线上的权益波动；
- 净利润曲线，记录平仓盈亏累加值的波动，每完整交易一次则统计一次；
- 盈亏曲线，记录每根 K 线上的浮动盈亏累加值波动；
- 资金详细信息，记录每根 K 线上的组合资金概况和组合中所有模型的具体交易信息，如下图所示，用户可结合合约明细和资金概况具体分析组合成员的盈利水平。



**资金详细信息**

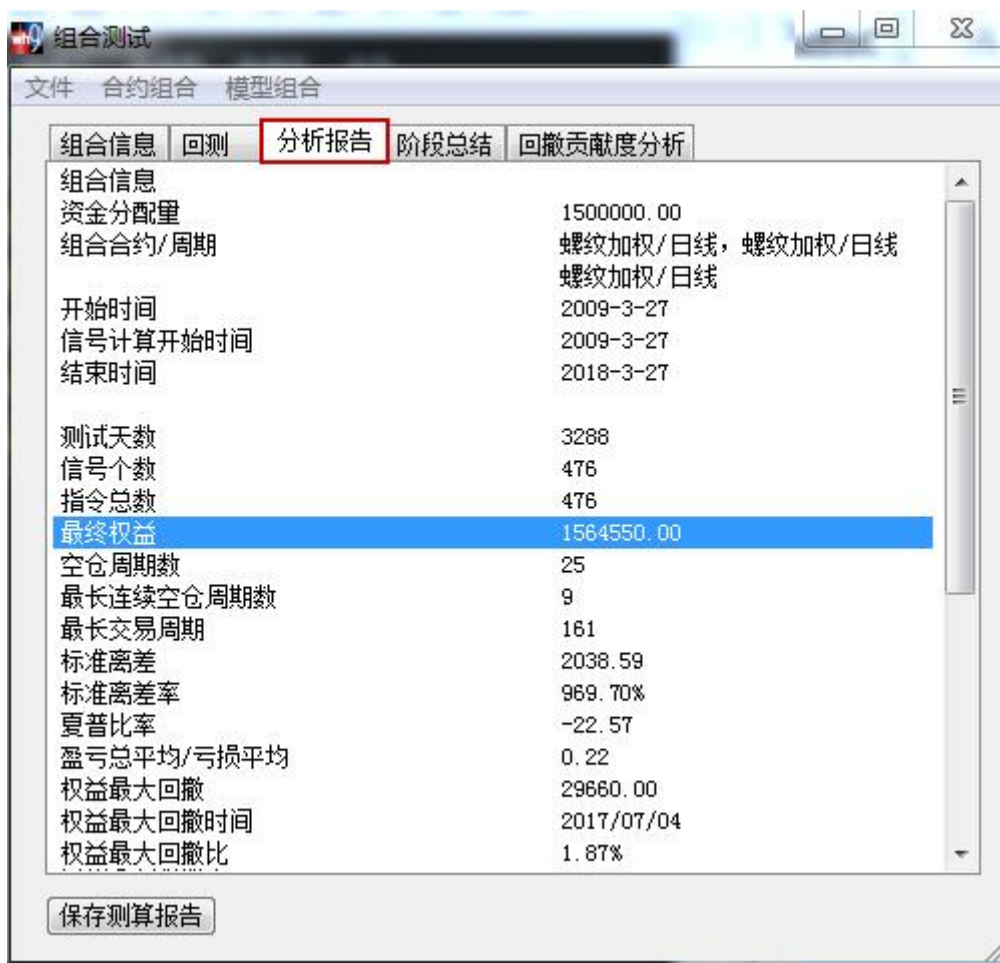
序号	时间	权益	净利润	盈亏
15665	2017/04/27 09:30:00	1563919.7	64462.7	63919.7
15666	2017/04/27 10:00:00	1564039.7	64462.7	64039.7
15667	2017/04/27 10:45:00	1564099.7	64462.7	64099.7
15668	2017/04/27 11:15:00	1564109.7	64462.7	64109.7
15669	2017/04/27 13:45:00	1564329.7	64462.7	64329.7
15670	2017/04/27 14:15:00	1564379.7	64462.7	64379.7
15671	2017/04/27 14:45:00	1564829.7	64462.7	64829.7
15672	2017/04/27 21:00:00	1565049.7	64462.7	65049.7
15673	2017/04/27 21:30:00	1565109.7	64462.7	65109.7
15674	2017/04/27 22:00:00	1565069.7	64462.7	65069.7
15675	2017/04/27 22:30:00	1565149.7	64462.7	65149.7
15676	2017/04/27 23:00:00	1565119.7	64462.7	65119.7

**合约明细 - 2017/04/27 09:30:00**

合约名称	持仓(多,空)	均价	总手续费	总平仓盈亏	浮动盈亏	可用资金	权益
螺纹加权/日线/动态突破	1,0	3014.00	2055.31	31660.00	-540.01	526653.48	529064.68
沪铜加权/日线/动态突破	0,0	0.00	0.00	24825.00	0.00	524825.00	524825.00
沪铝加权/日线/动态突破	0,0	0.00	0.00	10030.00	0.00	510030.00	510030.00

查看详情

如下图，是组合回测的分析报告。利用各项指标将不可估量的市场风险量化展现给投资者，投资者可以此分析组合的风险-收益水平，调整交易组合，合理分散风险。



如下图，是组合回测的回撤贡献度分析报告。提供组合回撤贡献度和回撤比贡献度分析图表，计算组合成员的贡献度大小。投资者可以对得分较低的模型进行调整，提高组合整体收益和抗风险能力（回撤贡献度得分的计算方法详见本页常见问题7）。



如下图，是组合回测的阶段分析性报告。具体罗列了组合在各个周期阶段的交易和收益情况，方便投资者统计账户交易情况，进行风控管理。

组合回测 - 趋势震荡组合效果

文件 合约组合 模型组合

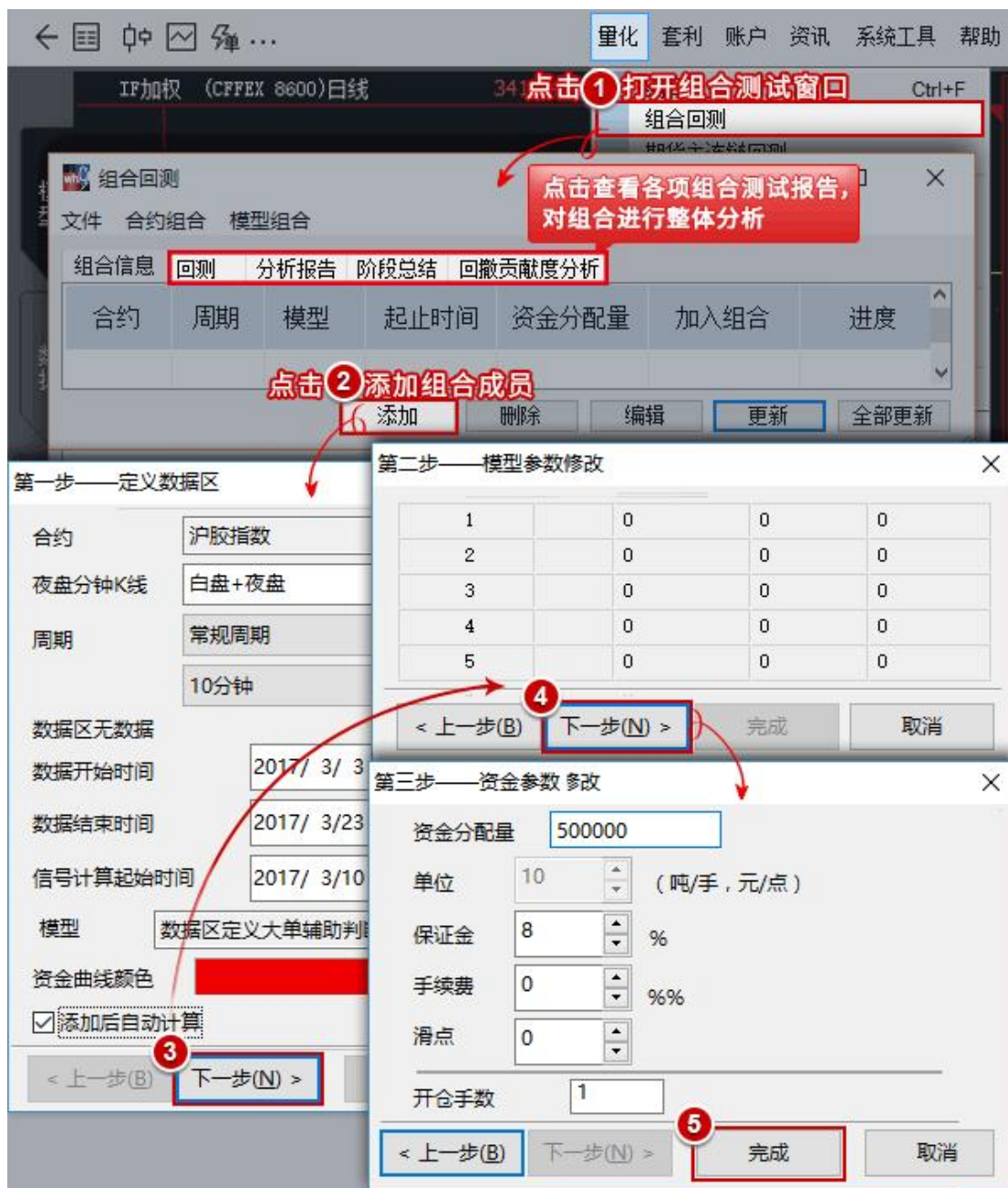
组合信息 回测 分析报告 **阶段总结** 回撤贡献度分析

年度分析						
年份	权益	净利润	盈利率	胜率	平均盈利/亏损	净利润增长速度
2016年	1554820.00	54900.00	3.66%	40.00%	2.52	3.67%
季度分析						
季	权益	净利润	盈利率	胜率	平均盈利/亏损	净利润增长速度
2016年第四季度	1554820.00	24830.00	1.62%	42.08%	2.47	1.07%
月度分析						
月份	权益	净利润	盈利率	胜率	平均盈利/亏损	净利润增长速度
2016年11月	1548080.00	15530.00	1.01%	42.67%	3.17	0.85%
周分析						
周	权益	净利润	盈利率	胜率	平均盈利/亏损	净利润增长速度
2016年11月25日	1543300.00	5150.00	0.33%	40.00%	1.85	0.39%
日分析						
日期	权益	净利润	盈利率	胜率	平均盈利/亏损	净利润增长速度
2016年04月11日	1507330.00	2110.00	0.14%	50.00%	2.09	0.17%
2016年04月12日	1508060.00	730.00	0.05%	66.67%	3.79	-0.03%
2016年04月13日	1507740.00	-320.00	-0.02%	75.00%	1.97	-0.07%
2016年04月14日	1507870.00	130.00	0.01%	0.00%	0.00	0.03%
2016年04月15日	1508130.00	260.00	0.02%	100.00%	∞	0.01%
2016年04月18日	1508790.00	660.00	0.04%	40.00%	3.00	0.03%

保存分析报告 图表分析

## (2) 组合回测操作步骤:

如下图①-⑤所示是如何对组合策略进行测试:



相关常见问题解答:

1、已经进行过测试的组合，能否保存起来以便于下一次直接调出？

答：可以，通过下图所示的方法对当前组合进行保存。再次打开点击【打开现有组合文件】即可。

组合回测 - 趋势震荡组合效果

文件		合约组合	模型组合
新建	阶段总结	回撤贡献度分析	
打开上次组合	模型	起止时间	资金分配量
打开现有组合文件	震荡调整	2015/12/31-2018/05/25	500000
<b>保存组合</b>	趋势模型	2015/12/31-2018/05/25	500000
另存组合	波段策略	2015/12/31-2018/05/25	500000
删除组合			
导入现有组合文件			
导入			
导出			

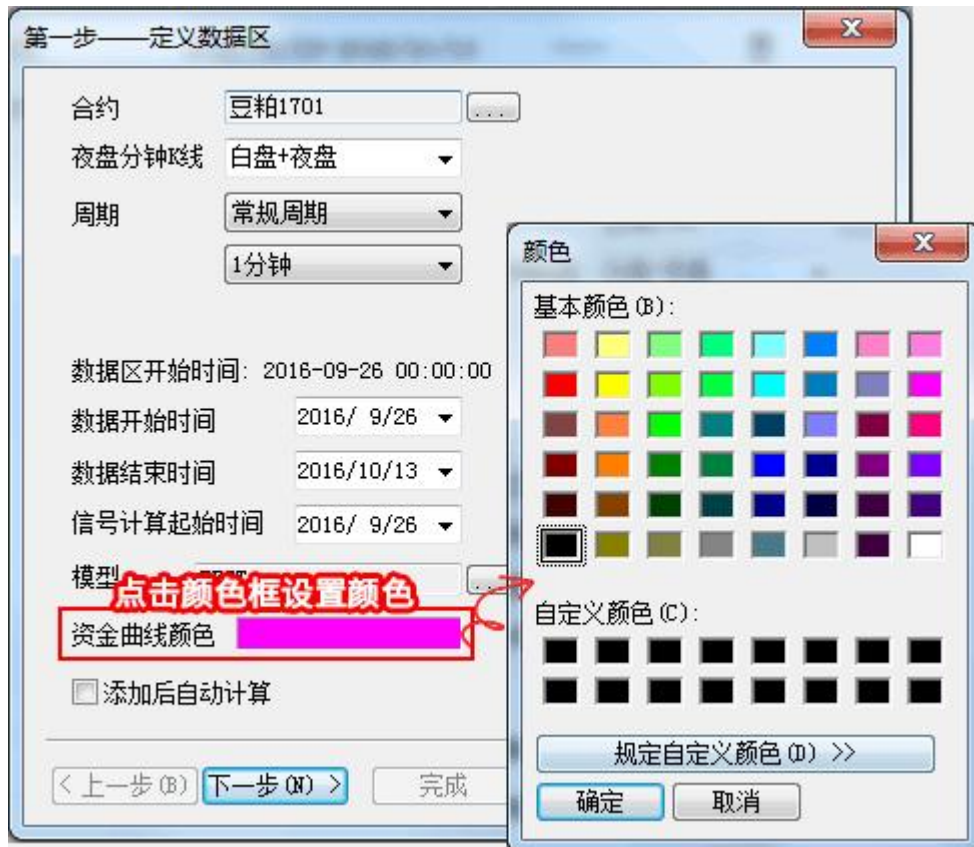
点击对组合进行保存

## 2、为什么添加组合成员后“进度”中显示的是未计算？

答：这是由于在添加组合成员时没有勾选【添加后自动计算】；选中未计算的组合成员，点击鼠标右键->更新，进行计算即可。

## 3、组合成员的资金曲线颜色可修改么？

答：可修改，如下图所示是如何修改资金曲线颜色：



#### 4、已经添加的组合成员如何修改合约参数？

答：选中要修改的组合成员，点击鼠标右键菜单->编辑，即可对组合成员参数进行修改。

#### 5、为什么组合测试界面的【回测】中，有左右两个纵坐标，分别代表什么？

答：左侧纵坐标为各策略组合后的资金曲线坐标，右侧纵坐标为各个策略的资金曲线坐标。

#### 6、组合测试界面的【阶段总结】中，权益增长速度如何计算？

答：权益增长速度=（本期净利润-上期净利润）/本期期初权益。

#### 7、回撤贡献度得分越高越好么？

答：是的，回撤贡献度分数越高说明回撤越小，贡献越大。

回撤贡献度得分算法：

每个策略在每个时点上比较，回撤值最大的得1分，次之得2分，以此类推。每个策略总分就是每个时点得分之和。如在某一个时点上，策略A的回撤大，策略B的回撤小，那么策略A得1分，策略B得2分。每个时点上都会计算得分，策略A和策略B的得分分别加和就是各自的总得分。得分高说明策略的回撤小，贡献大。

注：回撤值=回撤前的最大权益-当前权益

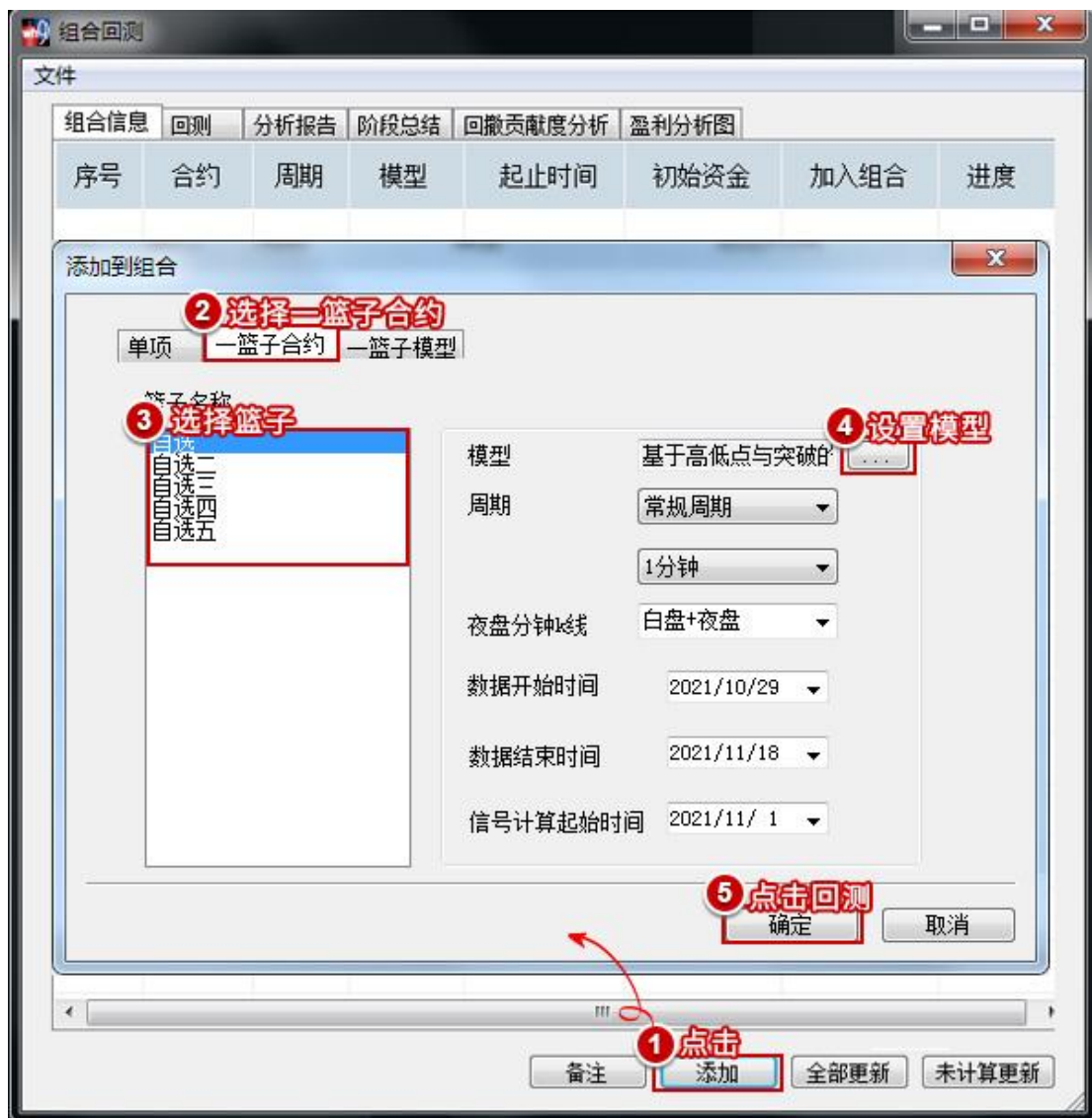
回撤比贡献度得分算法：

回撤比贡献度以每个时点的最大回撤比（最大回撤/最大回撤前的最大权益）做比较，计算各个点的得分之后加起来计算总得分。

### 8、如何实现快速将同一模型加载至不同合约进行测试？

答：利用组合回测->回测一篮子合约可实现该想法。

先将要回测的合约加入到自选篮子中。如下图，在组合回测界面点击右下角【添加】按钮选择【一篮子合约】，在弹出的窗口中选择要加载的自选篮子，选好回测的模型、周期等参数，点击【确定】按钮，即可用同一模型对不同合约进行测试。



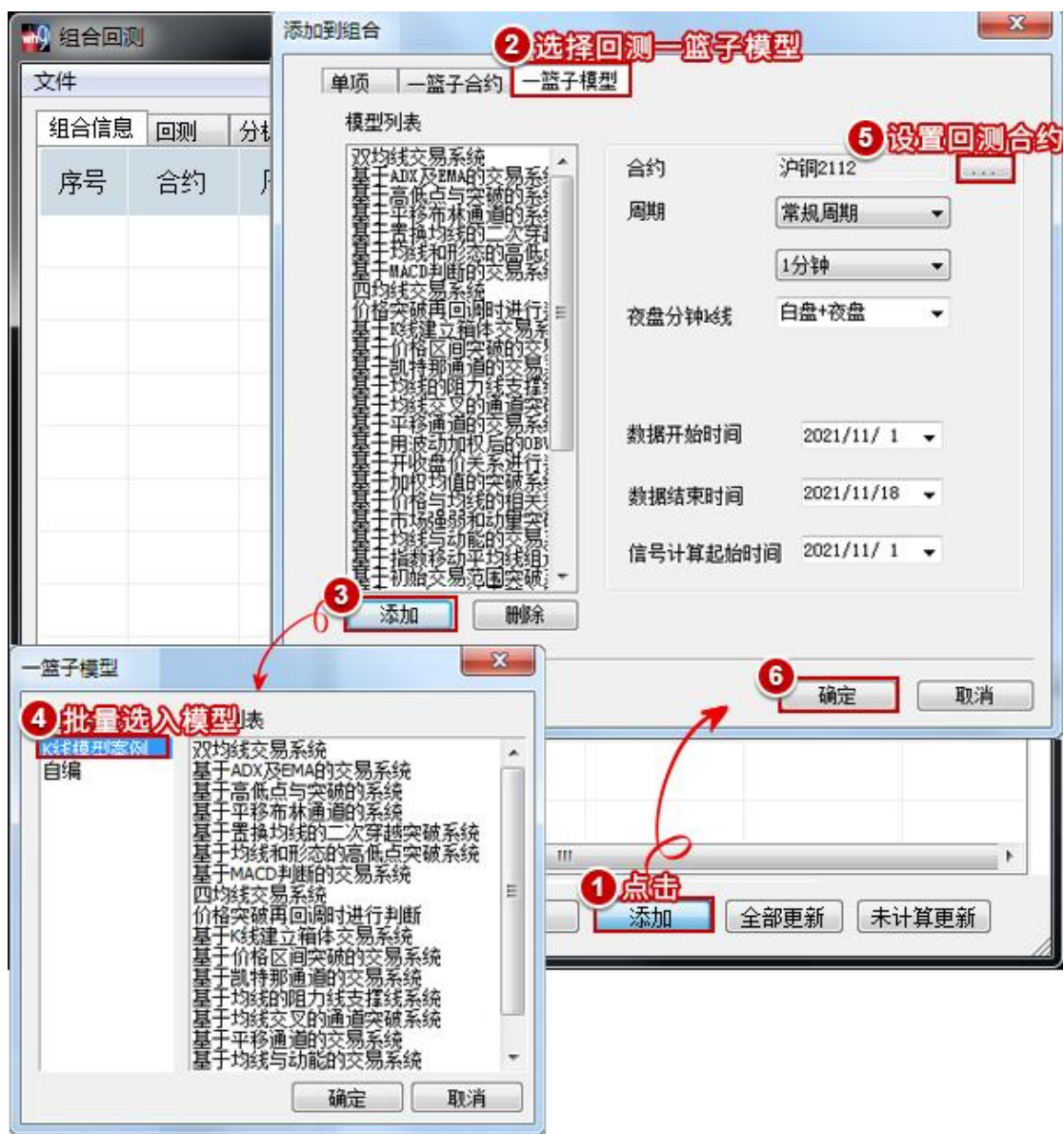
### 9、如何实现快速将同一合约同一周期加载不同模型进行测试？

答：利用组合回测->回测一篮子模型功能可实现该想法。

如下图，在组合测试界面点击右下角【添加】按钮，选择【一篮子模型】，在弹出的窗口中



批量选入要加载的模型。在弹出窗口中设置一些选项，点击【确定】，可实现快速对同一合约同一周期加载不同模型进行测试。



10、我已经设置了回测参数，为什么还提示“没有设置参数的合约不能参与回测”？

答：弹出这个提示说明在回测之前没有对篮子中的合约设置参数，

可以在软件右上角->量化->编写公式->设置->设置期货回测参数进行批量设置。

### (三) 期货主连链回测

期货是主力交易方式，为检验策略对主力合约的连续下单效果，一些投资者会将策略加

载到主连上测试运行，这样便存在两个问题：

一方面，主连只是对主力合约数据的机械拼接，换月时产生的大幅跳空会影响信号计算；另一方面，换月前后两个合约的持仓趋势可能发生改变，存在信号转换的风险，而无法体现出主力合约真实的连续交易效果。

针对这种情况，T9 特推出期货主连链回测，不再使用品种主连自身的 K 线数据计算信号，而是采用各个月份合约自挂牌以来的全部 K 线数据计算信号，规避品种主连 K 线数据换月跳空、新旧合约趋势相反等情况对量化计算准确性的影响，主力换月时，旧主力月份平仓，新主力根据月份合约的 K 线数据重新计算开仓信号。帮助投资者洞察策略效果，客观评价策略好坏。

### 案例：主连链回测分析主力合约连续交易效果

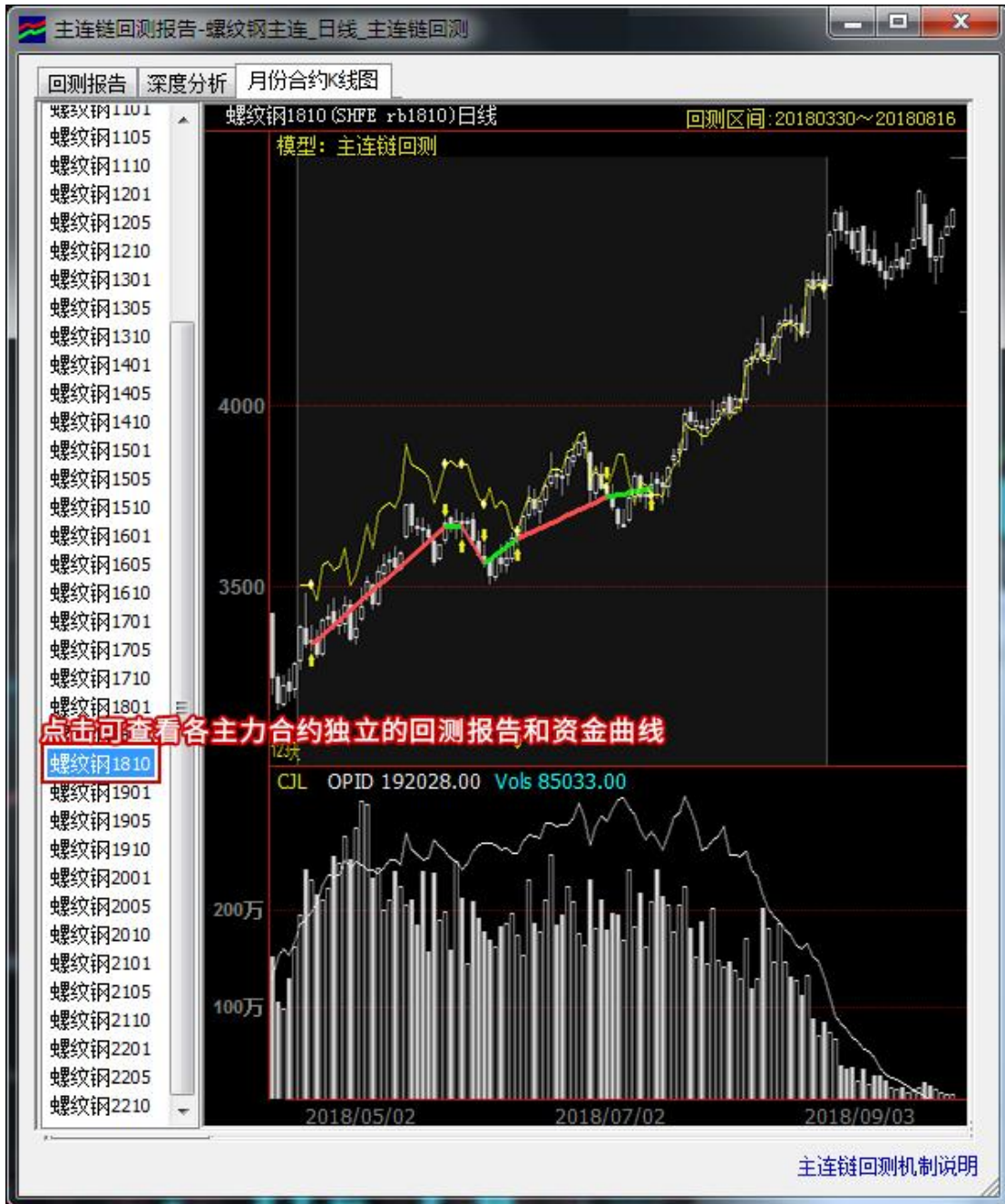
模型中不写入 `Trade_Other` 语句，直接加载到品种主连上，系统会自动调用主连链回测。如下图，回测后在 K 线图上点击鼠标右键->模型回测报告，可查看策略在主力合约上连续运行的效果分析报告和资金曲线，供投资者分析策略连续交易的整体效果，便于调试交易策略，降低试错成本。



如下图，在2019年12月6日螺纹钢2001换月到螺纹钢2005时，2001合约是空头开仓信号，2005合约是多头开仓信号，换月时，系统对2001合约进行了平仓操作，并重新计算了2005合约的开仓信号，规避了2005合约行情上涨的损失。



如下图，主连链回测还提供从信号计算开始时间到数据结束时间范围内，所有主力合约的独立回测报告和资金曲线。可查看到策略在每一个主力合约上的盈利效果，对策略在不同测试区间内分段调试。



注:

- 1、加载合约必须是主连合约;
- 2、模型中不能写有 Trade\_Other 函数;

### 主连链回测机制:

主连链回测针对多个年度的主力月份合约进行连续交易的模型回测计算,用长期的历史数据验证策略的有效性。

主力换月时,回测计算的处理机制如下:

旧主力合约，清仓；

新主力合约，延续使用旧主力清仓后的资金，但是重新计算信号。

注：

主连链回测只能用于分钟、小时、日线周期；

主连链回测不支持运行优化函数；

主连链回测不支持外盘合约；

相关问题解答：

1、主连链回测机制总结：

<https://www.wenhua.com.cn/popwin/wh8zhulianlian.htm>

2、一开一平过滤模型在主连链回测时，资金曲线图上有连续的两个 BK 信号？

答：这个是正常的，在换月的时候，对旧主力合约平仓，在新主力上重新计算信号开仓，所以出现了两个连续的 BK 信号。“成交明细”列表中记录了这笔清仓的具体成交信息。

## （四）独立算法交易模型回测

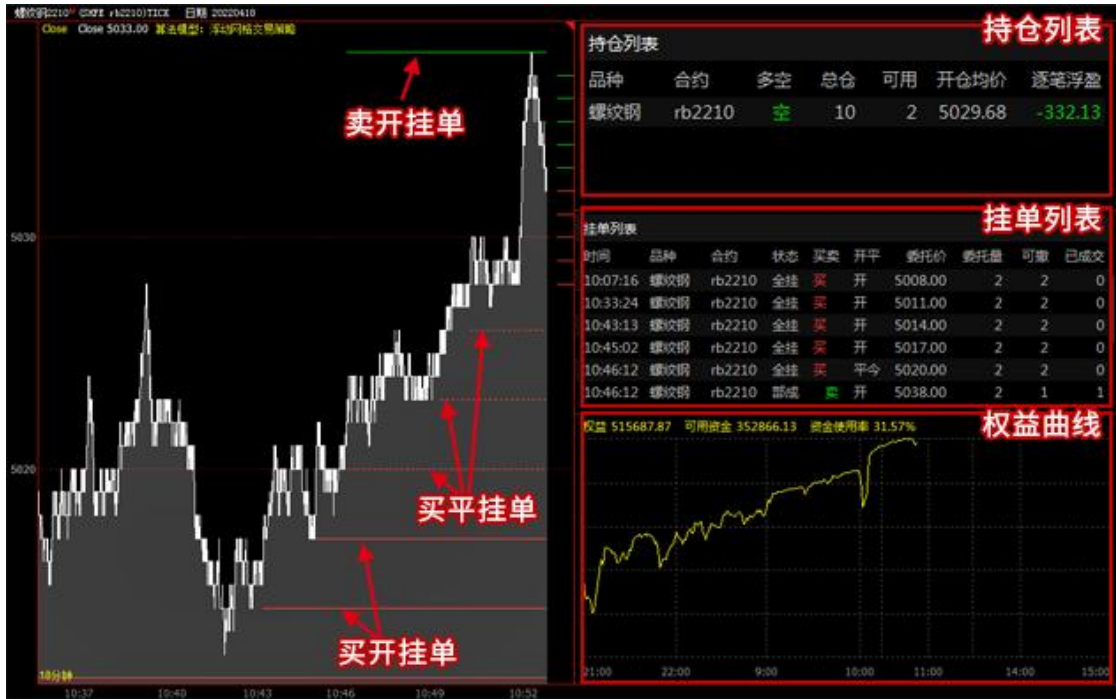
独立的算法交易模型通过挖掘盘口报价和逐笔成交数据，从极为短暂的市场变化中寻求获利，把交易策略应用到秒周期级别。这类高频率的下单策略在投入实盘使用前，往往需要进行更为精细的可行性分析。

T9 的数据粒度可支持算法高频模型逐笔 Tick 精细回测，提供可视化的盘口算法模型回测平台，直观展现每一笔的下单情况和资金变动，为投资者提供精准的策略检验。

### 1、案例：盘口算法模型回测

T9 支持虚拟撮合的回测平台，帮你验证模型好坏。T9 的虚拟撮合机制既考虑到了委托价等于对手价时，对手盘挂单量不足，部分成交的情况，同时，当委托价等于排队价时，还会计算挂单成交比例，按比例撮合成交，虚拟真实交易过程。

如下图，软件提供可视化的盘口算法模型回测平台，能够同时展示 TICK 数据变化、持仓信息、挂单详情以及权益曲线变化过程。



回测后，提供 360 度视角的多维度回测报告，如下图，用户可结合模型回测报告、回测交易明细以及回测运行日志，详细的了解模型的虚拟交易效果，360 度的了解自己的模型。



## 2、算法模型回测操作方法

在 tick 图界面上，加载独立的算法交易模型，即可对算法模型进行逐笔 tick 回测计算。

点击鼠标右键，可切换算法模型回测日期、调整回测速度、查看模型回测报告、回测交易明细以及回测运行日志等操作。





### 3、算法模型回测机制

- (1) 算法交易模型逻辑检测仅支持独立的算法交易模型类型；
- (2) 算法模型回测界面由左侧的 tick 图和右侧的持仓列表、挂单列表、资金曲线图组成：
  - ①回测期间，显示买卖 1-5 档价格线，红线为买价，绿线为卖价；
  - ②回测期间，在 tick 图上用射线显示现有的挂单，左边起点是挂单的时间点，红色为买挂单，绿色为卖挂单；实线为开仓挂单，虚线为平仓挂单；



(3) 回测完成时，撤所有挂单，挂单射线和挂单列表也会清空；

(4) 算法模型回测的合约，由 Data 数据区、Def\_TickData 和主图合约组成，包括国内外期货、套利和股票。

### 相关问题解答：

#### 1、算法交易模型回测可以回测更远的数据吗？

答：不可以。算法交易模型属于高频日内交易策略，回测只提供一个交易日的 tick 数据，不支持更远的连续数据回测。

#### 2、提示“当前交易日未收盘不支持算法交易模型回测”怎么办？

答：当天 tick 数据在收盘后 30 分钟可以回测算法模型，盘中可在 K 线图上点击右键->切换日期，选择历史交易日进行回测。

#### 3、如何调整算法模型回测的资金参数？

答：工具条上“设置模型资金参数”，支持设置合约集中每个合约的资金参数；

## （五）期货多因子分析

T9 精选期货因子库囊括量价、动量、期限结构、持仓、基本面、BETA 等一系列因子，以期货市场全品种为标的，提供单因子筛选、多因子排名分析等功能。可借助多因子综合排名信息，对多个品种在同一时间点的强弱情况进行对比分析，构建对冲组合，实现期货横截面交易策略。

### 案例：期货横截面交易策略

横截面策略是一种泛套利策略，主要是指的是在同一个时间点（或者同一个时间区间），同时做多、做空多个品种，形成一个多空对冲组合，以期多空两个方向交易的盈利之和为正的一类策略。

横截面策略盈利的逻辑是基于期货价格的变动：强者恒强/弱者恒弱的假设，即期货价格具有一定惯性，趋势行情具有延续性，前期上涨幅度较大的合约，后期还能够继续上涨的概率更大。

例如：使用期货品种筛选功能，对市场所有的期货品种按照涨跌幅和资金流向进行排序，根据综合排名结果做多前 5 的品种，做空后 5 的品种，形成一个对冲组合，买入上涨强势的品种，卖出处于弱势的品种进行对冲获利。

**期货多因子分析**

因子名称	方向	操作
量价因子 - 涨幅	正向	×
量价因子 - 资金流向	正向	×

综合排名 正序 前 5 名 开始筛选

序号	品种	涨幅	资金流向(万)
1	焦炭加权	2.27%	68223.49
2	铁矿加权	1.88%	166198.27
3	PVC加权	1.93%	36849.31
4	沪银加权	1.05%	45667.02
5	锰硅加权	1.70%	16998.82

数据日期: 2021/08/05 加入自选

**期货多因子分析**

因子名称	方向	操作
量价因子 - 涨幅	反向	×
量价因子 - 资金流向	反向	×

综合排名 正序 前 5 名 开始筛选

序号	品种	涨幅	资金流向(万)
1	玻璃加权	-1.67%	-52715.83
2	PTA加权	-2.32%	-26511.21
3	LPG加权	-2.95%	-12333.72
4	郑醇加权	-1.06%	-43215.46
5	螺纹加权	-0.61%	-87161.45

数据日期: 2021/08/05 加入自选

持仓	期权	行权	委托	成交	预备单	条件单	损益单	风控单	资金	合约	云条件单	出入金		
品种	合约	多	总仓	可用	今仓	今可用	开仓均价	盈利价差	逐笔浮盈	损益	价值	资金占比	盯市盈盈	投保
铁矿	i2109	多	1	1	1	1	1229.50		1.08%	200	18,443	200		
锰硅	SM109	多	1	1	1	1	7828.0		-0.34%	-20	5,871	-20		
PVC	v2109	多	1	1	1	1	9220.0		-0.36%	-25	6,915	-25		
焦炭	j2109	多	1	1	1	1	2697.50		0.99%	400	40,463	400		
沪银	ag2112	多	1	1	1	1	5277.0		0.13%	15	11,873	15		
螺纹	rb2110	空	1	1	1	1	5566.0		-0.12%	-10	8,349	-10		
玻璃	FG109	空	1	1	1	1	3026.0		-0.22%	-20	9,078	-20		
郑醇	MA109	空	1	1	1	1	2720.0		0.49%	20	4,080	20		
PTA	TA109	空	1	1	1	1	4936.0		0.54%	20	3,702	20		
LPG	pg2109	空	1	1	1	1	4725.0		-1.41%	-200	14,175	-200		
10个			10						0.31%	380	122,948	380		

如下图，一周以后，共盈利 639950 元，收益率 5.18%。

持仓	期权	行权	委托	成交	预备单	条件单	损益单	风控单	资金	合约
品种	合约	多	总仓	可用	开仓均价	逐笔浮盈	浮盈比例			
锰硅	SM109	多	100	100	7697.3	116,340	19.56%			
焦炭	j2109	多	100	100	2742.545	2,659,550	58.93%			
铁矿	i2109	多	100	100	1184.95	-2,149,500	-147.73%			
PVC	v2109	多	100	100	9155.6	-25,325	-3.71%			
沪银	ag2112	多	100	100	5287.9	-227,835	-19.72%			
螺纹	rb2110	空	100	100	5576.7	243,690	30.46%			
PTA	TA109	空	100	100	5189.440	-59,280	-14.89%			
郑醇	MA109	空	100	100	2707.1	112,130	28.81%			
玻璃	FG109	空	100	100	3023.0	296,060	34.33%			
LPG	pg2109	空	100	100	4818.1	-325,880	-21.81%			
10个			1000			639,950	5.18%			

调用方法:

如下图，点击右上角菜单【量化】->期货多因子分析。

期货多因子分析

因子名称	方向	操作
量价因子 - 涨幅	正向	×
量价因子 - 资金流向	正向	×

+ 因子

综合排名 正序 前 5 名 开始筛选

<input checked="" type="checkbox"/>	序号	品种	涨幅	资金流向(万)
<input checked="" type="checkbox"/>	1	焦炭加权	2.27%	68223.49
<input checked="" type="checkbox"/>	2	铁矿加权	1.88%	166198.27
<input checked="" type="checkbox"/>	3	PVC加权	1.93%	36849.31
<input checked="" type="checkbox"/>	4	沪银加权	1.05%	45667.02
<input checked="" type="checkbox"/>	5	锰硅加权	1.70%	16998.82

数据日期: 2021/08/05 加入自选

注:

- ① 多因子分析使用上一交易日的数据计算;
- ② 根据因子所选方向排序，并列排名时，下一排名需要加上并列排名个数;
- ③ 并列排名的品种，按代码顺序展示;
- ④ 空值排到最后，并列排名。

## 五、期货量化运行详解

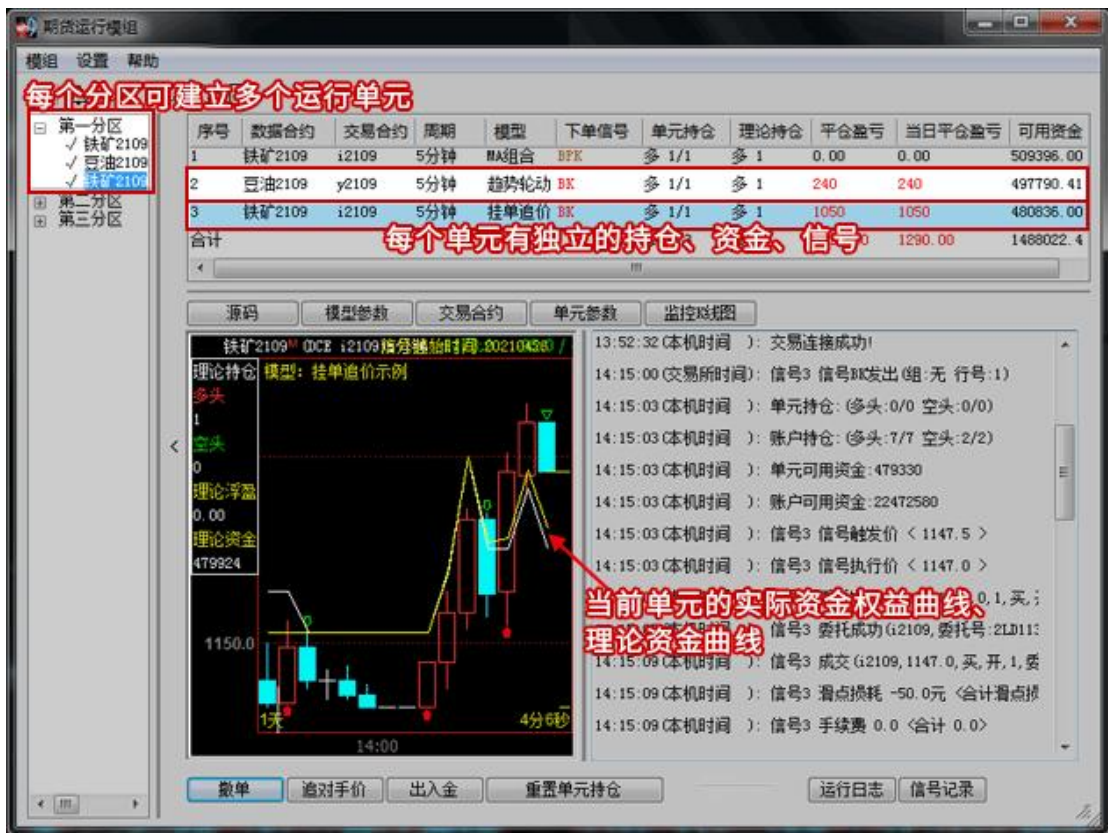
### (一) 期货模组实现自动的头寸管理、风险控制

模组中每个运行的模型称为单元，各自分配一定量的资金运行下单。模组采用这种独立

的单元管理策略，可以实现更高端、智能的资金管理和风险控制。

为了分散风险，投资者通常会选择多品种、多策略、多周期组合交易，软件能否保证各模型有序执行，并准确无误的计算每个模型的交易数据则至关重要。模组为每个单元分配独立的资金，用来管理当前策略的持仓和资金变化，各单元独立运行，为交易保驾护航。

如下图，在一个分区内同时运行 3 个单元，各个单元之间独立运行互不干扰。在运行模组界面可以看到各个单元的持仓、资金、下单方向和分区的合计持仓、可用资金等基本信息，可基于单元的历史运行效果显示连续的理论资金曲线和真实资金曲线。



模组还支持查看每一个分区的组合资金曲线，如下图的黑色资金曲线，表示的是“螺纹-趋势+震荡”交易组合的实际资金曲线，直观反映交易组合的整体收益情况。



### TIPS:运行单元和实盘账户的关系

T9 的大部分客户，通常会在一个合约上运行不同特点的多个模型，组合交易分散风险。模组中每个运行的模型称为单元，所有单元之间独立运行。

#### 1、单元的资金与实盘账户的资金是不同的

单元的资金是由用户随意分配的，和实盘账户资金没有直接关系。

大多数情况下，单元的资金之和不超过账户总资金数。用户在分配单元资金时会参考实盘账户的总资金量，把实盘账户的总资金，按照风险分散的逻辑，合理分配到各个单元。

#### 2、单元的持仓与实盘账户的持仓也是不同的

单元的持仓，是这个单元的模型出信号自动下单的结果。

在用户不进行手动交易的情况下，各个单元持仓之和等于实盘账户的总持仓。

## 以下是模组的三个典型应用

### 1、模组实现资金管理方法

仓位管理和风险控制是资金管理的重中之重，模组平台独立的单元管理，为投资者在策略中实施资金管理提供了便利。可以直接获取当前运行模型的盈亏和资金变化，一些复杂的

账户资金管理操作都可以在模组中通过量化语句轻松管控。

## 思路 1：通过资金管理函数控制独立运行单元开仓头寸

行情波动反复无常，你还在使用固定手数进行下单？运行模组中使用独立的单元，管理策略的资金，投资者可以按照当前运行单元中的资金比例下单或者按资金使用率下单，来合理控制开仓头寸。

**关键字：SetDealPercent 按当前单元资金比例下单**

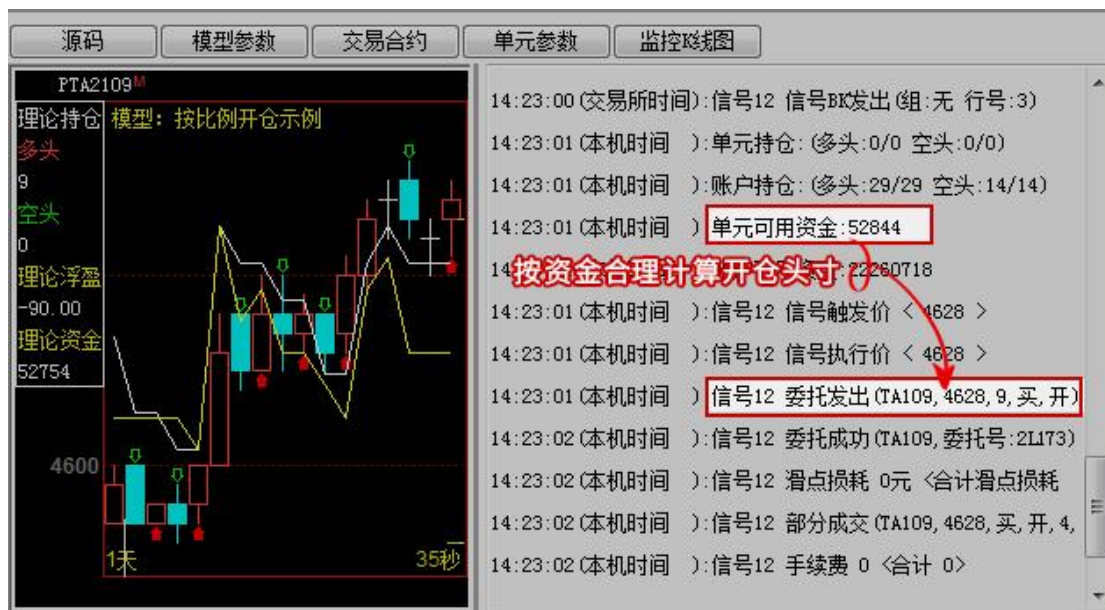
用法：写入 Setting 字段下表示每次按单元的理论资金的 fPercent 比例下单。

计算公式：fPercent 比例可下单手数=(前期可用资金+平仓释放的保证金+平仓盈亏)\*fPercent/(最新价\*保证金比例\*交易单位)。

例如：Setting

```
SetDealPercent:20; //每次按当时可用资金的 20%下单
```

如下图，根据当前单元可用资金 20%的比例开仓，贴合账户盈利情况科学合理的设计头寸大小。



## 思路 2：通过资金管理函数对运行单元进行风险控制

行情的异常波动我们很难在模型中全部都量化出来并加以限制，但是行情波动带来的影响会直接体现在资金的变化上，那么是否可以调用到策略的资金来控制进出场呢？运行模组有专门的运行单元统计资金和持仓，在编写策略时可直接调用到当前单元的资金权益，配合



策略优化函数灵活设计进出场条件。

**关键字：**MoneyTot 返回当前单元的理论权益

MoneyTot=模组当前单元可用资金+持仓保证金，模型进行仓位控制、下单手数等资金管理时使用。

例如：`If (MoneyTot<HHV (MoneyTot, 0)*0.95) Sell; //权益回撤一定幅度时平仓止损`

如下图，是当运行单元资金回撤一定幅度时止损，实时跟踪单元资金变动，及时避免资金回撤带来的风险，锁住账户盈利。



## 2、理论持仓、单元持仓二套持仓管理机制

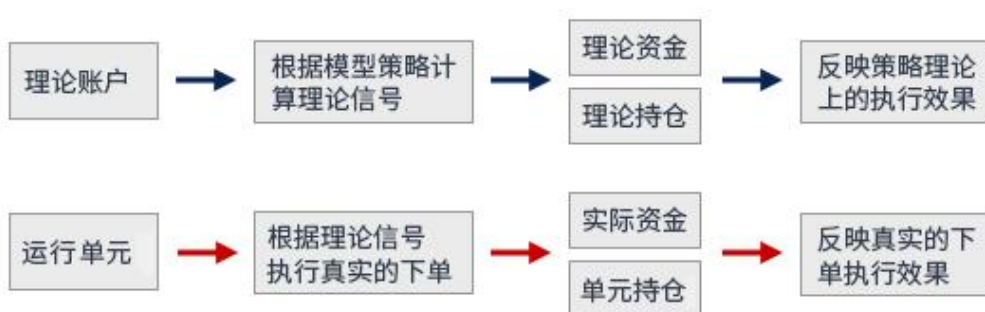
受制于市场流动性、期货撮合成交机制、网络速度等因素，量化信号不可能 100%按照模型里设定的价格方式完美成交。鉴于此，模组使用理论账户和运行单元，对理论上的信号、持仓和信号执行的实际持仓，分别管理。

### (1) 理论持仓和单元持仓的概念

什么是理论持仓，什么是单元持仓？他们之间又有怎样的执行关系呢？通俗点来说：按照模型计算的理论上应该执行的信号即为理论信号，根据理论信号计算得出的资金和持仓分别为理论资金和理论持仓；

模型运行时根据理论信号执行真实的委托，并将真实的成交结果反映到运行单元中，实际交易产生的资金和持仓则为实际资金和单元持仓。

如下图是理论账户和运行单元的关系示例。



二套持仓管理，保障各个单元有条不紊的运行，各单元的资金得到有效的监控。投资者对比理论账户和运行单元的持仓和资金，也可以清楚了解到策略的执行效果和真实交易的滑点成本大小等。

在下图的模组界面中，①为理论信号，②为按照理论信号计算的理论持仓，③为单元持仓，④为理论资金曲线，⑤为实际资金曲线。



如上图，当策略持续运行，实际资金曲线记录的是当前单元的实际盈亏情况，通过理论资金曲线与实际资金曲线的对比，投资者便能了解到当前单元的滑点成本大小。

## (2) 如何提高信号执行的成交比率

T9 软件在编写模型时，可针对交易指令指定委托价格方式，来提高成交率，降低信号下单不成交的概率，或者取得有利的价格成交，降低交易成本。

可以对不同的信号设置不同的委托方式，或者极端情况下终止下单，例如，`BK(N,Price);//`建仓 N 手，以 Price 价格发委托；下单价格 Price 可以指定为最新价/排队价/对手价/自动连续追价/超价/市价/触发价/指定价。

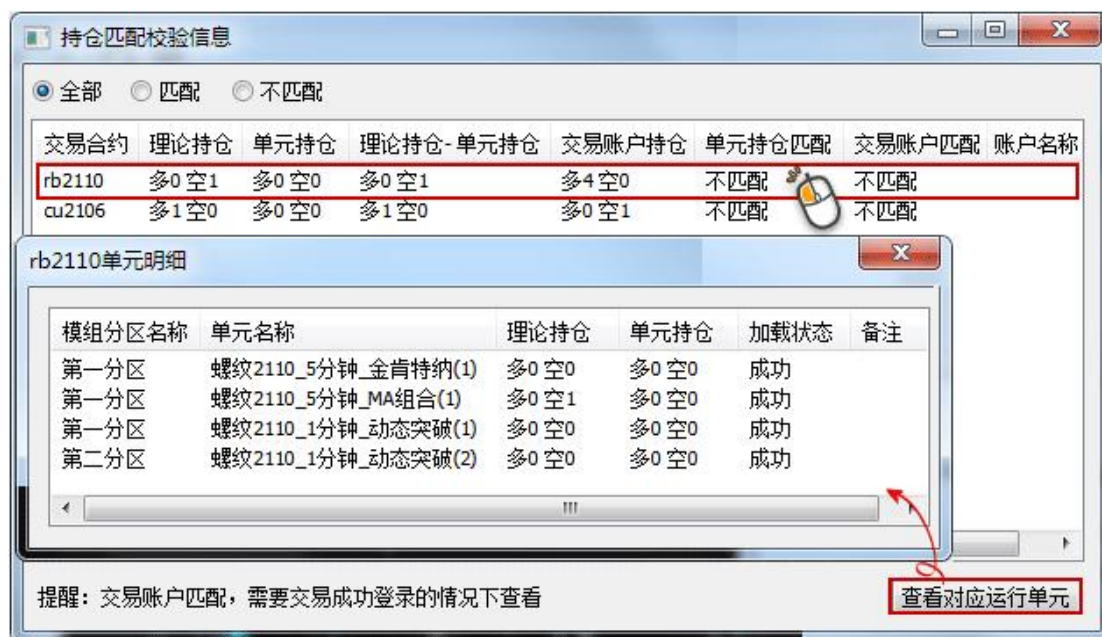
每一个信号可以采用不同委托价格实现不同的目的。例如：要的是便宜的成交价位，不在乎能成交几手，不便宜宁可不成交，可以用排队价；要得是保证全部成交，不在乎成交价位，可以用市价...

## (3) 持仓匹配校验

模组单元正常运行时，理论持仓和单元持仓也应该是一致的，但是如果网络环境不稳定或委托方式不合适导致不成交，都会影响到单元的运行，所以也建议投资者养成持仓校验的习惯，及时了解持仓匹配情况。

机构投资者运行单元数量较多，一个一个去检查账户持仓是否匹配需要的时间太长，软件提供智能的持仓匹配校验功能，一键查询真实账户持仓和策略持仓是否匹配，同时可以查看运行单元是否正常加载以及加载失败的原因。

操作方法：模组界面工具条，最右边的匹配校验信息  按钮调出，如下图。



#### (4) 手动干预的方法

情况一：运行单元盘中运行过程中，自动交易发出了委托形成挂单，长时间未成交

解决办法：可点击如图示“追对手价”按钮进行手动干预，撤掉原有的委托挂单，以当前的对手价重新发送委托（点击一次追一次），以此确保及时成交。

通过这个追价按钮的成交，运行单元会把成交价格自动带入作为权益计算的依据，权益计算会按照追价成交的价格计算。

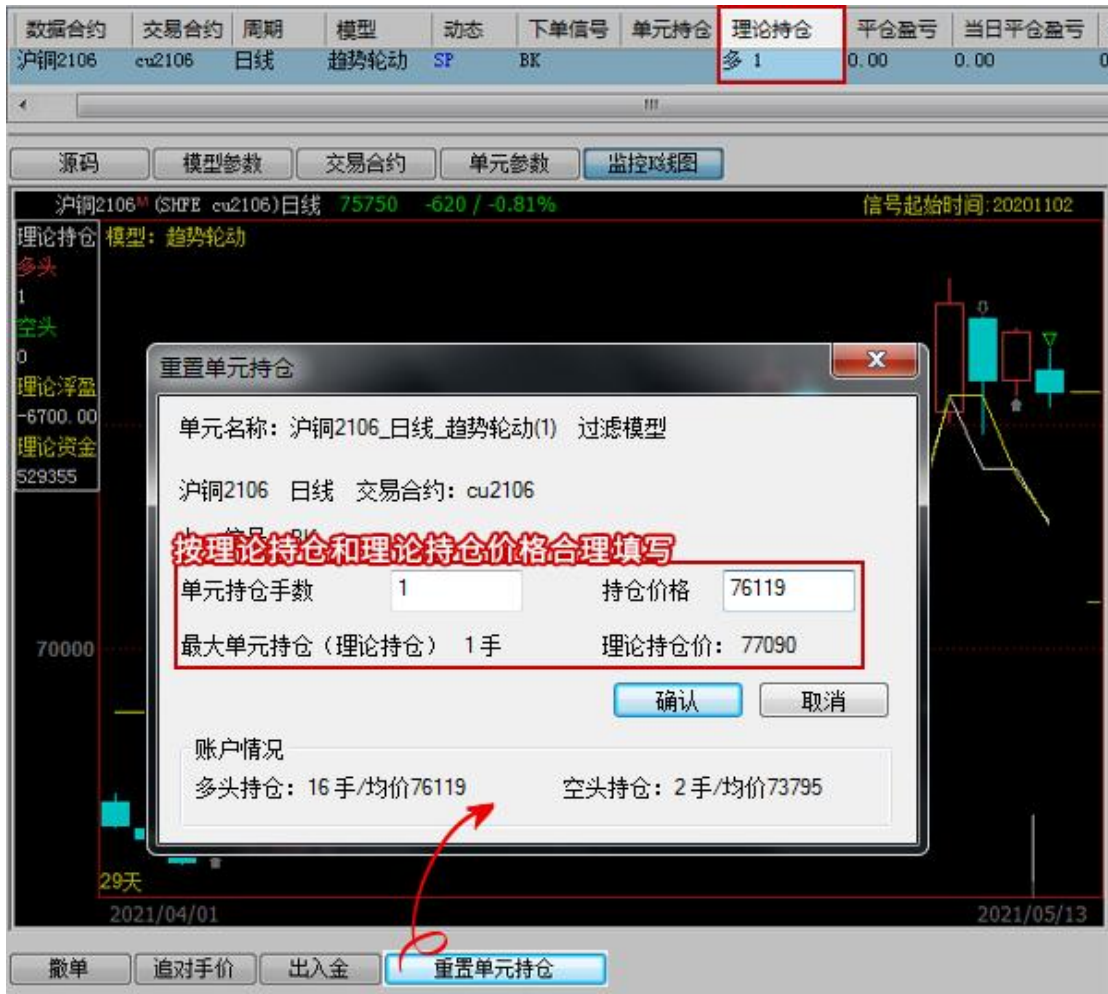


情况二：实际账户中有持仓，理论持仓有持仓，单元持仓小于理论持仓，该怎么办？

在用户做不到软件持续运行，在交易时段关机了，或者电脑断电非正常关机等情况，重新启动模组一般会出现这个现象。

解决办法：如下图，通过“重置单元持仓”功能，将交易账户中的持仓带入到当前单元，就可以保持理论账户和单元持仓匹配。

通过重置单元持仓，会把输入的价格、持仓数量带入作为权益计算的依据，权益计算会按照你填写的价格/手数计算。



### (5) 模组的持仓管理机制和 TB 持仓同步机制的区别

T9 有理论持仓管理、单元持仓管理二套管理机制。T9 认可和接受流动性、网络造成的运行单元实际持仓和理论持仓不一致的实际情况，T9 能管理这种差异，可以根据单元实际持仓绘制模型的实际权益曲线。

TB 只有一种理论持仓管理机制，无法管理理论持仓和实际持仓不一致的情况，或者采用简单粗暴的持仓同步方式（延后几秒补仓强制实际持仓和理论持仓一致）；或者不去理会实际持仓和理论持仓的不一致。TB 只能用理论持仓绘制模型的权益曲线。

因为信号执行是有时效的，TB 延后几秒的补仓的成交价格难免和信号价格有出入，会导致模型的权益曲线和实际情况有出入，另外因为延后补仓提高交易成本会降低模型盈利率，甚至因为延后补仓导致亏损。

### 3、产品申购/赎回管理

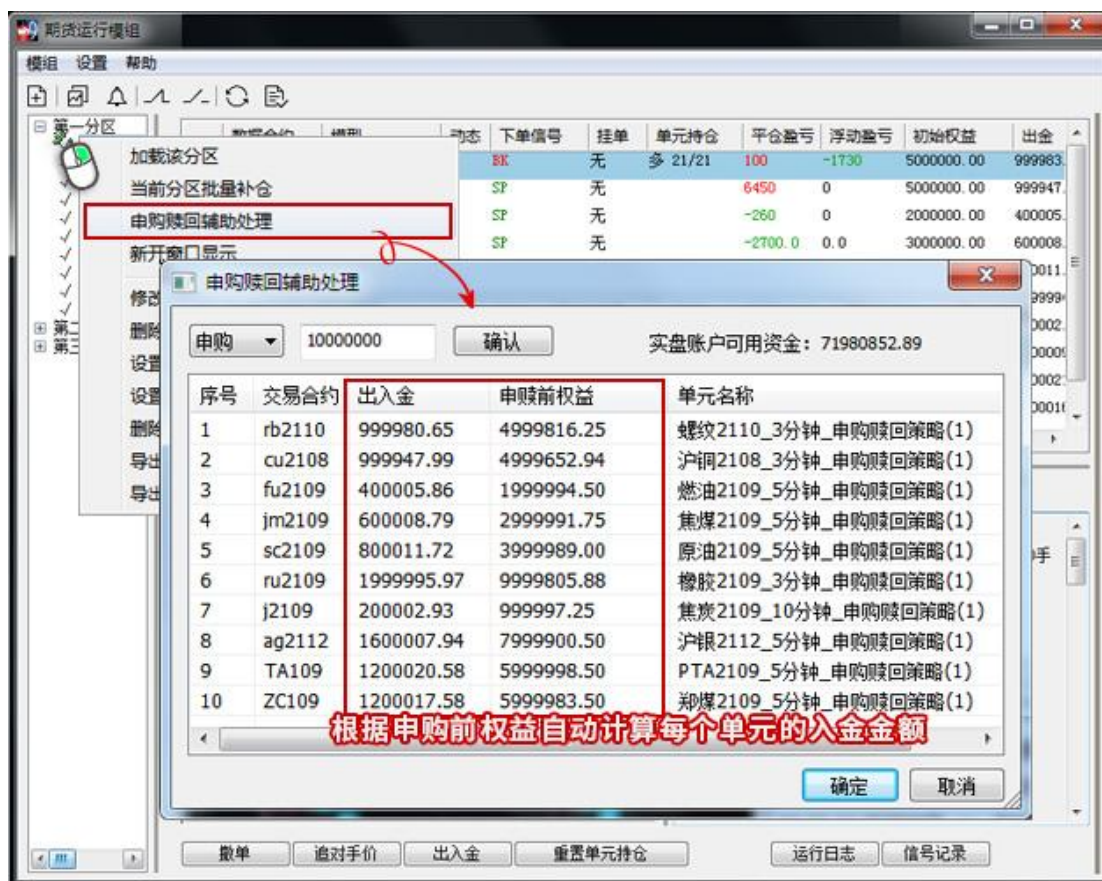
T9 的模组分区囊括一只资管产品的各个模型运行单元的情况下，每个模组分区相当于

一个资管产品。当需要对某一个资管产品申购/赎回时，只需输入申购/赎回总金额，软件会根据该分区内每个模组运行单元的权益占比，自动计算每个模组运行单元的出入金金额。同时，算法交易可通过函数获取出入金值并计算调仓手数，在需要调仓时进行智能分批下单，不仅能够减少滑点损失，还能最大程度的降低人工调仓出错的概率。

### 案例：产品申购/赎回管理举例

某产品 A 初始资金 5000 万，将资金分配给了 10 个不同的品种进行量化交易。某日客户申购该产品 1000 万，需要在下一个交易日的开盘阶段，将新入的 1000 万资金按比例分配给 10 个模组运行单元并进行调仓。

如下图，在模组分区点击鼠标右键->【申购赎回辅助处理】，可根据模组运行单元的权益占比自动计算出每个单元的入金金额。



点击“确认”按钮以后，自动对所有模组运行单元进行了入金调整，如下图。



交易模型可通过 F\_GetInOutMoney 函数取得模组当前单元本次出/入金值，并根据权益占比和持仓手数，计算出调仓手数并发出委托。

申购/赎回算法部分源码：

Vars

```

Numeric MQT; //模组权益
Numeric BPS; //多头持仓
Numeric IOM; //出入金
Global_Numeric IOMA; //记录出入金

```

... .. //省略部分源码

Begin

```

... .. //省略部分源码

//-----变量赋值-----//

If(OPFLG == 1) //变量赋值

```



```

{
    MQT = MoneyTot; //模组权益
    IOM = F_GetInOutMoney(); //出入金

    ... .. //省略部分源码

}

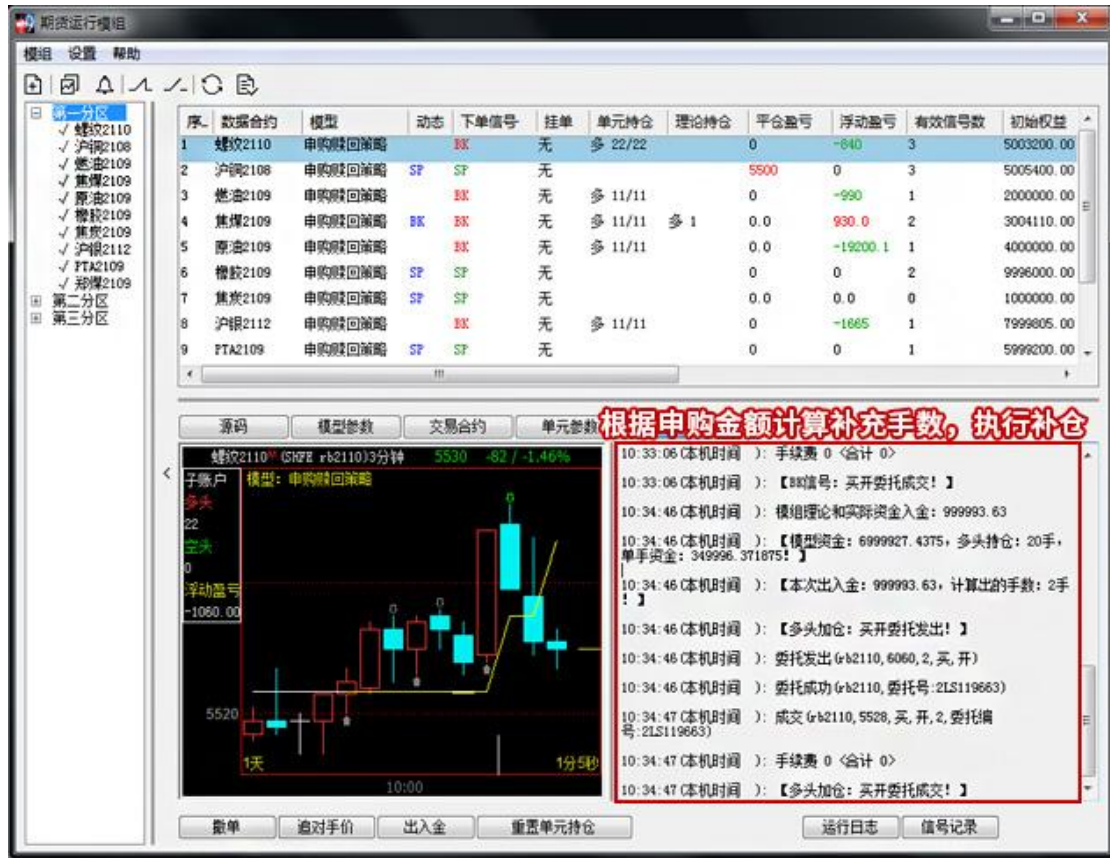
... .. //省略部分源码

//-----委托处理-----//
If(OPFLG == 1) //委托处理
{
    If(IOM != 0 && KPCFLG == 0) //如果有出入金，且未开启加减仓处理
    {
        IOMA = IOM; //记录出入金
        KPCFLG = 1; //开启加减仓处理
    }
    If(KPCFLG == 1) //如果已开启加减仓处理
    {
        If(BKFLG == 0 && BKDFLG == 0) //如果没有买开委托和买开处理
        {
            If(SPFLG0 == 0 && SPFLG1 == 0 && SPFLG == 0 && SPDFLG == 0) //如果没有卖平委托和卖平处理
            {
                If(BPS > 0) //如果有多头持仓
                {
                    NP = MQT / BPS;
                    BKCV = IIF(IOMA > 0, Floor(IOMA / NP), Ceiling(IOMA / NP, 1)); //根据入金金额计算补仓手数
                    Commentary("【模型资金: " + Text(MQT) + ", 多头持仓: " + Text(BPS) + "手, 单手资金: " + Text(NP) + "! 】");
                }
            }
        }
    }
}

```

```
Commentary("【本次出入金: " + Text(IOMA) + ", 计算出的手数: " +  
Text(BKCV) + "手! 】");  
If(BKCV > 0) //如果需要加仓  
{  
    BKDV = BKCV; //买开处理手数  
    BKDFLG = 2; //开启多头加仓买开处理  
}  
Else If(BKCV < 0) //如果需要减仓  
{  
    SPDV = Abs(BKCV); //卖平处理手数  
    SPDFLG = 2; //开启多头减仓卖平处理  
}  
}  
KPCFLG = 0; //委托处理标志归 0  
}  
}  
}  
}  
... //省略部分源码  
End
```

如下图，算法交易模型获取运行单元的申购/赎回信息，在开盘时对所有需要调仓的模组运行单元进行了补仓操作。



## 模组使用注意事项

### (1) 保持模组连续运行

只要在交易时段，软件要一直在保持运行，这样才能保证交易信号的连续性。出于节约的目的，在闭市后可以关机，但是一定要在交易重新开市前打开软件，否则就会影响连续性，造成信号和持仓对不上的情况。

量化信号能否正常执行，取决于模组中行情和交易的连续性。运行单元行情连续才能保证信号有序执行，精准下单。要想让模组代替人工执行下单，在有条件的情况下，尽量不要关机，保证电脑 24 小时不关机连续运行。

**注意：**正常关闭软件的过程中，软件会自动保存各个单元的状态。但是，如果在软件还在运行过程中直接切断电源，会导致状态丢失，导致下次启动模组时信号持仓不匹配。

### (2) 保证网络稳定

运行单元在盘中实时接收数据计算信号，网络延迟、数据丢包等问题都会直接影响到运行单元中信号的计算，对后续的执行造成影响。所以运行模组时，一定要保证网络环境的稳定性。

### (3) 避免手动干预

模组中各单元独立运行，每个单元分别管理对应策略的持仓和资金，与总账户不存在对应关系，所以手动下的单不能由模组接管。并且手动干预带入的持仓价格可能会和理论数值不符，导致盈亏的差异，从而影响运行单元信号的判断，所以在模组运行时，一定要避免手动干预。

#### 附：【不规范操作的解决办法】

	不规范操作	现象和原因	解决办法
<b>1.未连续运行</b>	非法关机 收盘前关闭软件 开盘后启动软件	1、单元持仓和理论持仓不匹配？ 2、模组中不显示单元持仓？ 3、实际持仓不能正常带入模组？  （原因：盘后正常关机，模组可以自动带入持仓。但盘中非法操作会导致模组中数据断档，单元的运行条件已经发生改变，所以无法接管之前的信号和持仓；）	如果当前运行的信号和持仓方向一致： 点击模组界面的“ <b>重置单元持仓</b> ”，把每个单元的持仓都手动带入；  如果当前的运行信号和持仓方向不一致： 在模组监控 K 线图窗口右键->“ <b>清除历史信号重新运行</b> ”，带入之前的持仓；
<b>2.网络不稳定</b>	本地网络不稳定	1、回测有信号，但是运行单元中没有信号？ 2、K 线主图的数据和运行单元里的数据相差很大？  （原因：主图回测是从服务器直接申请的完整、准确的数据计算信号。运行单元中是盘中一笔一笔接收数据计算信号，如果丢包或断网，运行单元中的数据 and 信号就会受到影响；）	在模组监控 K 线图窗口右键->“ <b>重新计算历史信号</b> ”从服务器申请准确的数据重新计算信号。运行单元后续出信号延续该信号向下执行。  操作建议：此操作只适用于临时解决运行单元信号问题。建议优化网络环境，从根源上解决问题。

<b>3.手动干预</b>	在下单版手动平掉运行单元的持仓  频繁干预单元持仓	1、单元持仓为何提示不匹配？  2、运行单元中显示的盈亏数据，怎么和实际的开平价格计算的盈亏不符？  （原因：模组由独立单元管理资金和持仓，手动开仓运行单元中接管不到；手动平掉单元的持仓，后续运行单元就只出平仓信号，不会真正执行委托；另，手动干预带入的持仓价格可能和理论不符，导致盈亏的差异，从而影响信号的判断；）	模组中不接管手动下单的持仓，如果需要运行单元接管，可参考问题1的解决方法。  操作建议：模组中要避免频繁的手动干预。如果是因为下单效果与预期不一致，建议调整编写思路，完善策略。
---------------	---------------------------------	---	--

## 附：模组的加载方式

模组运行单元的加载方法详见 [\(模组加载流程\)](#)

### 相关问题解答：

**1、模型源码中指令后面写有手数了，加载模型的“交易合约”处也填写了下单手数值，下单手数以哪个为准？**

答：源码中写的手数优先级高于加载模型设置的手数，所以如果源码中指令后面有手数，就按照源码中的手数下单，“交易合约”处设置的手数不起作用。当源码中指令后面没有手数的时候，再取加载模型“交易合约”处填写的手数值。

**2、模型加载计算慢怎么办？**

答：本地秒周期数据量过多会严重影响模型计算的速度，在主窗口 K 线图上单击鼠标右键 → **【重新计算历史信号】**，可以灵活选择加载所用的数据量，避免这一问题。

**3、为什么下单手数与信号手数不一致？**

(1) 最后信号为开仓信号；

信号手数 < 当前单元理论资金曲线计算可开手数：下单手数 = 信号手数；

信号手数 > 当前单元理论资金曲线计算可开手数：下单手数 = 可开手数。

例：信号手数为 10，下单手数却只有 5 手。

——根据运行单元分配资金计算，最大可开手为 5 手，所以即使信号手数为 10，实际下单只可以下 5 手。

(2) 最后信号为平仓信号；

信号手数 > 单元持仓：下单手数 = 单元持仓；

信号手数 < 单元持仓 : 下单手数 = 信号手数。

例: 信号手数为 10, 下单手数却只有 5 手。

——理论持仓为 5 手, 即开仓信号时只开了 5 手, 所以平仓只执行 5 手;

——理论持仓为 10 手, 但是单元持仓为 5 手 (未初始化进来持仓/手动调仓至 5 手/账号同步至 5 手), 所以平仓只执行 5 手。

#### 4、模组单元日志中的信号触发价和信号执行价、滑点分别是什么意思?

信号触发价: 发出委托时行情的最新价

信号执行价: 信号真实委托的价格

滑点: 成交价与信号触发价之间的差值

买入滑点=成交均价-信号触发价

卖出滑点=信号触发价-成交均价

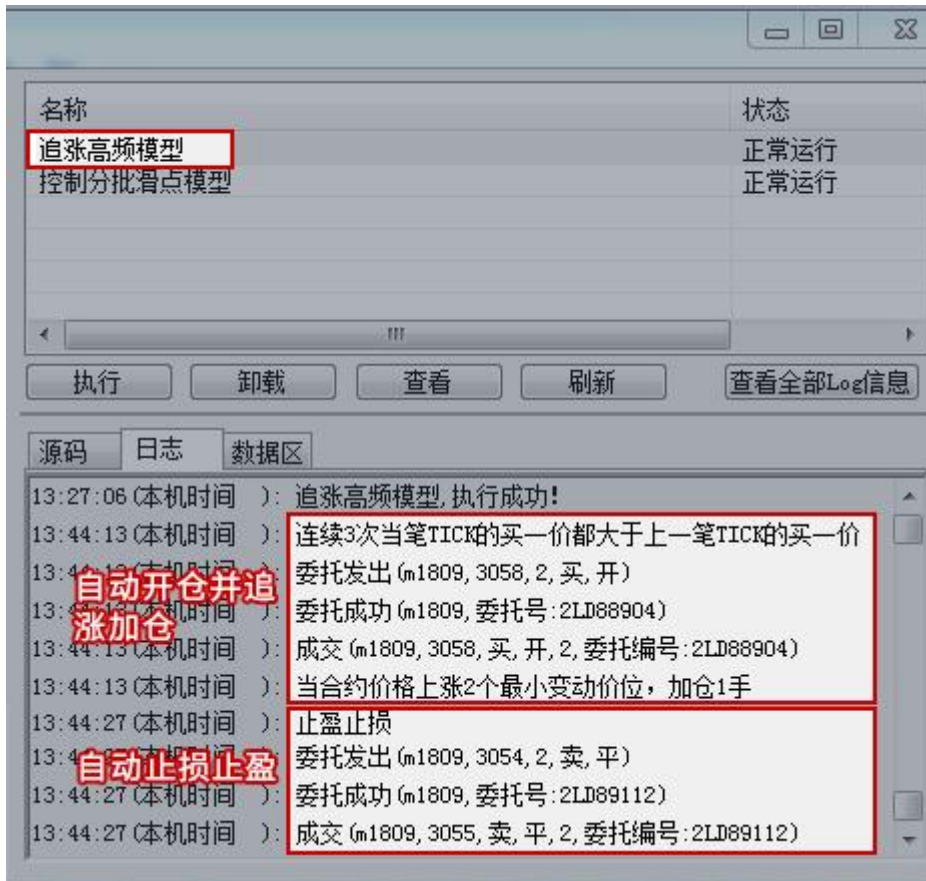
```
13:38:07 (本机时间 ): 信号4 信号触发价 < 91920 >
13:38:07 (本机时间 ): 信号4 信号执行价 < 91900 >
13:38:07 (本机时间 ): 信号4 委托发出 (ni1801, 91900, 1, 买, 开)
13:38:08 (本机时间 ): 信号4 委托成功 (ni1801, 委托号:2LS278300)
13:38:08 (本机时间 ): 信号4 成交 (ni1801, 91900, 买, 开, 1, 委托编号:2LS278300)
13:38:08 (本机时间 ): 信号4 滑点损耗 -20元 <合计滑点损耗 -40元>
```

## (二) 独立算法交易模型运行池自动交易

独立的算法交易模型通常是调用盘口数据或逐笔成交数据结合账户盈利、持仓情况进行日内高频交易策略, 不需要以 K 线图数据为计算依据, 因此是在专门的算法交易运行池中加载运行。

### 案例: 算法交易模型运行池自动交易

如下图, 是追涨高频盘口模型在算法交易运行池中的执行效果, 算法交易运行池中会记录每一笔单子的真实委托情况, 并根据模型中编写的输出语句, 输出指定字符串, 供投资者分析策略的运行效果, 掌握策略的盈利情况。



附：独立的算法模型加载方式

独立算法模型加载方法详见（[独立算法交易模型加载流程](#)）

## 第三部分 常见问题

### 一、公式编写常见问题

#### 1、T9 编写语法

[https://www.wenhua.com.cn/guide/myQuant\\_zjyfhs2.htm](https://www.wenhua.com.cn/guide/myQuant_zjyfhs2.htm)

#### 2、常用系统函数说明

[http://www.wenhua.com.cn/guide/myQuant\\_zjyfhs3.htm](http://www.wenhua.com.cn/guide/myQuant_zjyfhs3.htm)

### 3、T9 常见编写和机制说明

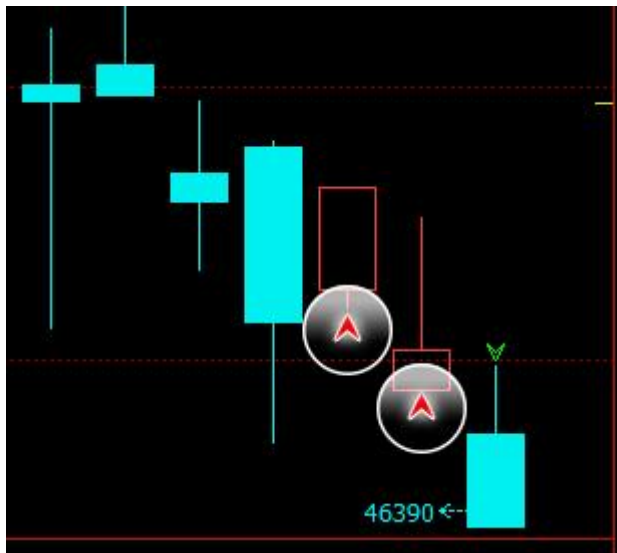
[http://www.wenhua.com.cn/guide/myQuant\\_zjyfs6.htm](http://www.wenhua.com.cn/guide/myQuant_zjyfs6.htm)

### 4、T9 的信号指令机制：

- ① 股票不支持反手指令执行
- ② 信号不支持锁仓

### 5、什么是加减仓模型

答：加减仓模型，允许连续出开仓信号或者连续出平仓信号，可实现加仓、减仓。如下图。



### 6、加减仓模型编写时需要注意哪些问题

答：A、加仓模型中，加仓语句需要判断是否是第一次开仓

方法：可利用判断当前是否有持仓或判断上一个信号是否是相同信号的方法确定是否是第一次开仓。如，加仓条件中增加判断 `MarketPosition == 1;`

B、减仓模型中，减仓语句需要判断当前是否有可平持仓

方法：可利用 `MarketPosition` 这样的函数来判断持仓情况。如，平仓条件中增加判断 `MarketPosition != 0;`



## 7、为什么我的加减仓模型不加仓

答：模型中写有 AddTime 函数，才可以实现加仓。

例：

Setting

```
AddTimes:3;
```

Begin

```
If(IsUp)
{
    BK(3, 0);
}
If(IsDown)
{
    SP(0);
}
```

End

## 8、如何快速查找源码中函数的解释

答：双击函数名称后，在选中的函数名称上单击鼠标右键，即可查看该函数的说明，如下图。



## 9、宽语言源码中不同颜色代表什么意思

答：系统函数（含关键字），蓝色字体

自定义函数或数字，红色字体

操作符，紫色字体

变量或参数，黑色字体，如下图。

```
1  Vars
2  Numeric TmpValue(0);
3  NumericSeries ADValue(0);
4  Begin
5  If(High > Low)
6      TmpValue = ((Close - Low) - (High - Close)) / (High - Low) * Vol;
7  If(CurrentBar == 0)
8  {
9      ADValue = TmpValue;
10 }Else
11 {
12     ADValue = ADValue[1] + TmpValue;
13 }
14 PlotNumeric("AD", ADValue);
15 End
```

## 10、引用合约、引用周期函数的限制情况

答：一个模型中#ImPort、#Call、#Call\_Plus、#Call\_Other 总的语句个数不能超过 6 个；

一个 T9 软件同时使用的模型中#ImPort、#Call、#Call\_Plus、#Call\_Other 总的数量不能超过 120 个。

## 11、如何实现 k 线走完发委托

答：T9 软件所采用的信号机制为出信号立即下单，K 线走完复核的方式。要实现 K 线走完发委托的运行方式，可以通过 Ref(条件, 1); 的形式来实现，即上一根 K 线满足条件，当根 K 线生成后立即发委托。

例：

```
Begin
    if(Ref(isup, 1))
        BPK;
    if(Ref(isdown, 1))
        SPK;
end
```

## 12、如何实现单向指令量化，即公式条件单

答：编写只有一个开仓指令的模型，加载到运行模组中，运行方式选择忽略回测运行，即可变相实现开仓指令只发出一次，后续不再发出信号的运行方式。

例：

```
Vars
    Numeric Ma5;
    Numeric Ma10;
Begin
    Ma5=Ma(Close, 5); //5 周期均线
    Ma10=Ma(Close, 10); //10 周期均线
    If(CrossUp(Ma5, Ma10)) //5 周期上穿 10 周期均线，做多
    {
        BK;
    }
End
```

## 13、T9 模型如何实现类似 T8 的自动换月移仓

T8 的模型可以实现自动换月移仓，T9 的模型中也是使用 TRADE\_OTHER 函数来实现需要写在 Setting 字段下，具体参考函数说明。

格式：

```
Setting
    Trade_Other:Auto;
```

## 14、T8 中的画线语句，在 T9 中如何实现

例如：LL:LLV(LOW, N), DOT;

T9 中的写法：

```
Params
Numeric N(20);
Vars
```

```
Numeric LL;  
  
Begin  
  
LL = LLV(Low, N);  
  
PlotLine("LL", LL, Red, Dot);  
  
End
```

### 15、T8 中仅显示数值不画线的句子，如何转换

例如：LL:LLV(L, N), RGB(99, 138, 173), NOTEXT;

T9 中的写法：

```
Params  
  
Numeric N(20);  
  
Vars  
  
Numeric LL;  
  
Begin  
  
LL = LLV(Low, N);  
  
PlotLine("LL", LL, RGB(99, 138, 173), NoDraw);  
  
End
```

### 16、NumericSeries 和 NumericArray 两种类型有什么区别

答：NumericSeries 是数值型序列变量，每一根 K 线都有对应一个数据存储，例如收盘价，每根 K 线都存储一个收盘价。

NumericArray 是数值型数组，例如一个大小为 10 的数组，就是将 10 个数字存储到这个数组的 10 个位置。

一般思路使用 NumericSeries 即可，NumericArray 主要用于网格等思路存储网格开仓标识。

### 17、想把模型中的参数设置为小数

答：将参数写为整数，在模型中对参数除以相应倍数即可。

例如：

Params

```
Numeric N(5);
```

Begin

```
N/100*Close;
```

End

18、布尔型的参数、函数、变量类型数据如何改写到 T9 软件中

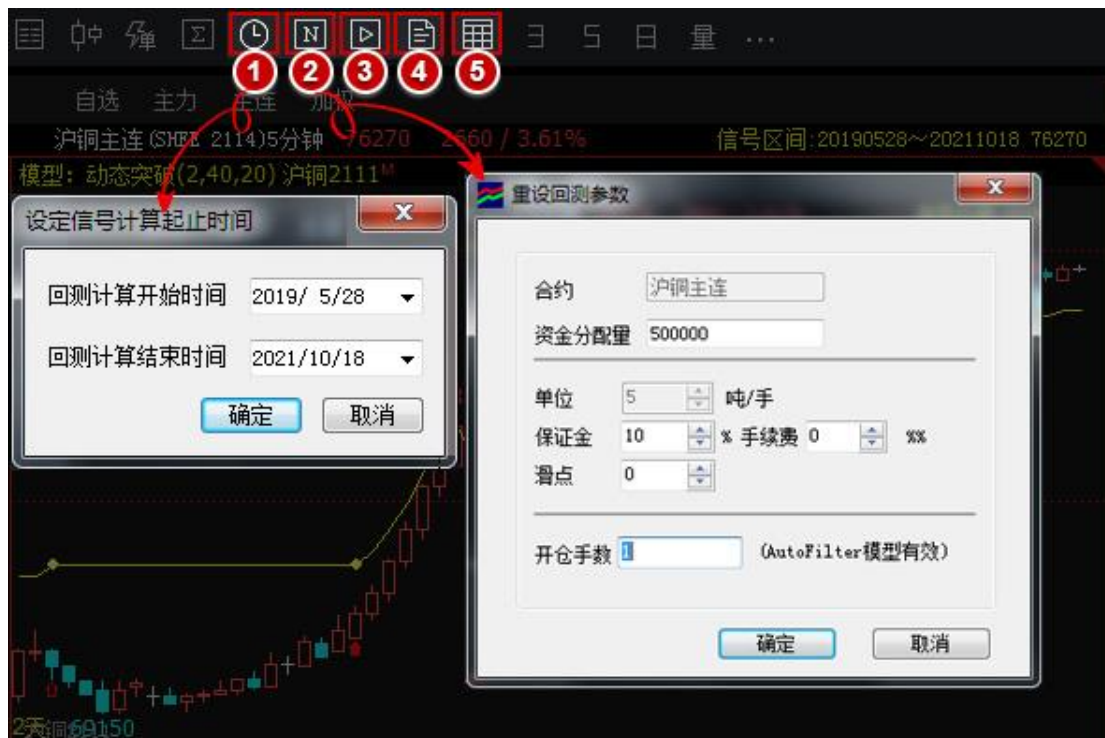
答：其他软件上的布尔型数据，在 T9 上可以用数值型数据替代，TRUE 是 1，FALSE 是 0。






19、T9 中 REF(CLOSE, 1) 是否完全等同于 CLOSE[1]

答：是的，两种写法主要是为了兼容不同软件的写法，表达的含义是一样的。

## 二、回测常见问题

1、如何修改回测时初始资金等相关参数



- ①  : 设定信号计算起止时间。
- ②  : 设置回测参数，如初始资金，开仓手数，保证金，手续费等。
- ③  : 装入到模组后台运行，将 k 线图上回测的合约和模型加载到模组中运行。
- ④  : 查看回测报告
- ⑤  : 查看信号明细

## 2、如何实现逐笔 tick 测试模型

答：模型中写入 MultSig 函数时，支持逐笔回测。

## 3、滑点是什么意思，设置多少合适

答：滑点是指委托价格与成交价格之间的差值。

回测参数中设置滑点，目的是为了回测效果最大限度拟合实盘的成交情况，具体设置多大的滑点要根据交易的合约和委托方式调整。

例如，以对价方式发委托，可以设置 1-2 个滑点，使回测结果尽可能的贴合品种行情特征，更接近实盘效果。

#### 4、为什么已经补充了数据，但是选择信号计算起始时间不能选择到补充的时间

答：回测之前需要将**数据合约**和**交易合约**的数据都补充一致再回测。

例如，模型中含有 Trade\_Other:Auto；函数指定了交易合约再加载到加权合约上测试时，数据合约是加权合约，交易合约是主连，补充了加权合约数据之后，还需要同步补充主连合约的数据再进行回测，如下图操作。



#### 5、为什么无法补充全部的数据，在补充数据中只能看到最近几个月

答：这种情况是因为回测合约选择的是具体月份合约，所以只能补充到自合约上市以来的数据。如果需要使用连续的历史数据进行回测，建议使用加权合约。

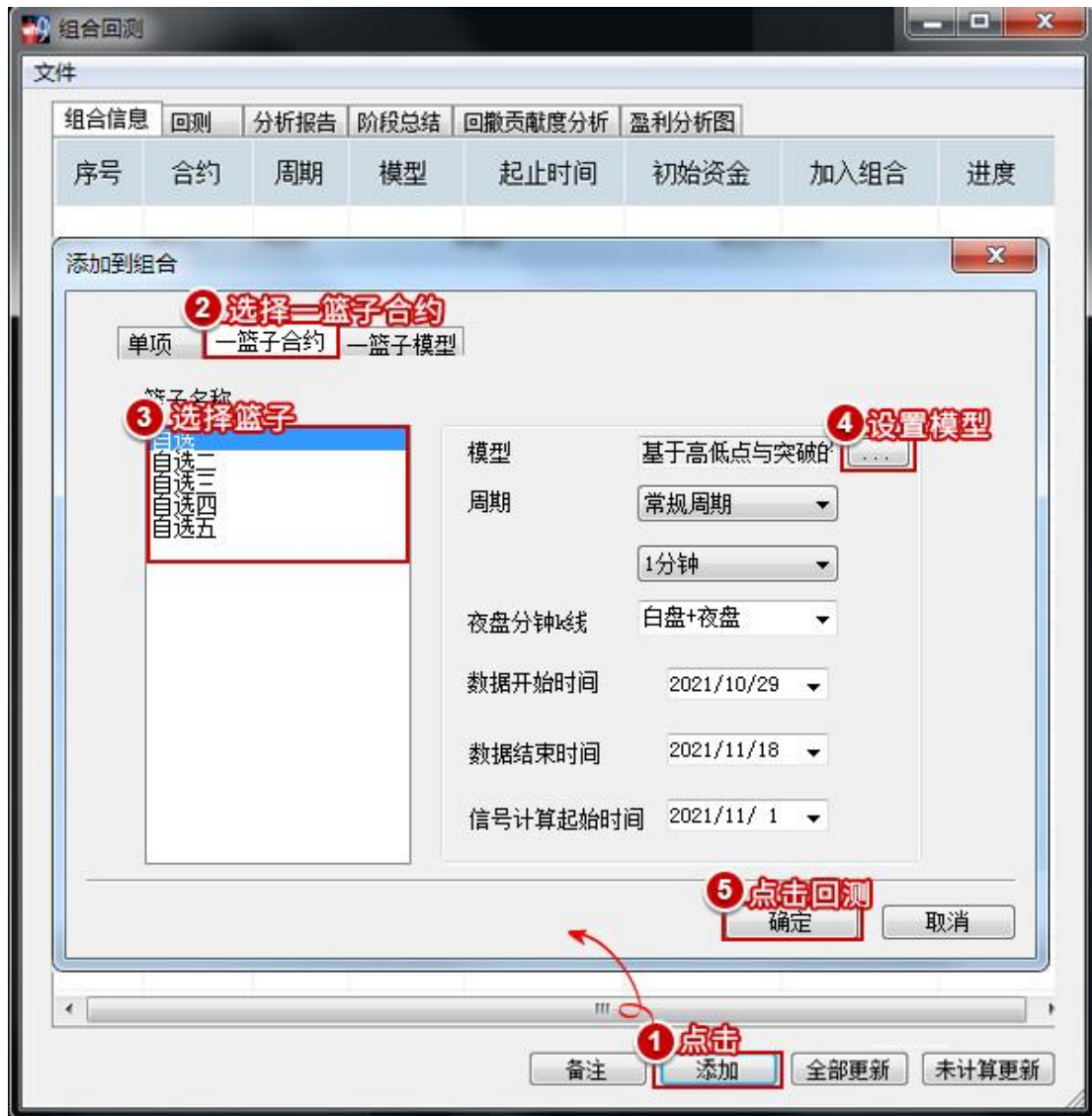
如：沪铜 1805 是 2017 年 5 月上市交易，所以只能申请到 2017 年 5 月上市以来的数据，而沪铜加权合约则支持 1996 年品种上市以来全部数据。

#### 6、如何实现快速将同一模型加载至不同合约进行测试？

答：利用组合回测→回测一篮子合约可实现该想法。

先将要回测的合约加入到自选篮子中。如下图，在组合回测界面点击右下角【添加】按钮选择【一篮子合约】，在弹出的窗口中选择要加载的自选篮子，选好回测的模型、周期等参数，

点击【确定】按钮，即可用同一模型对不同合约进行测试。

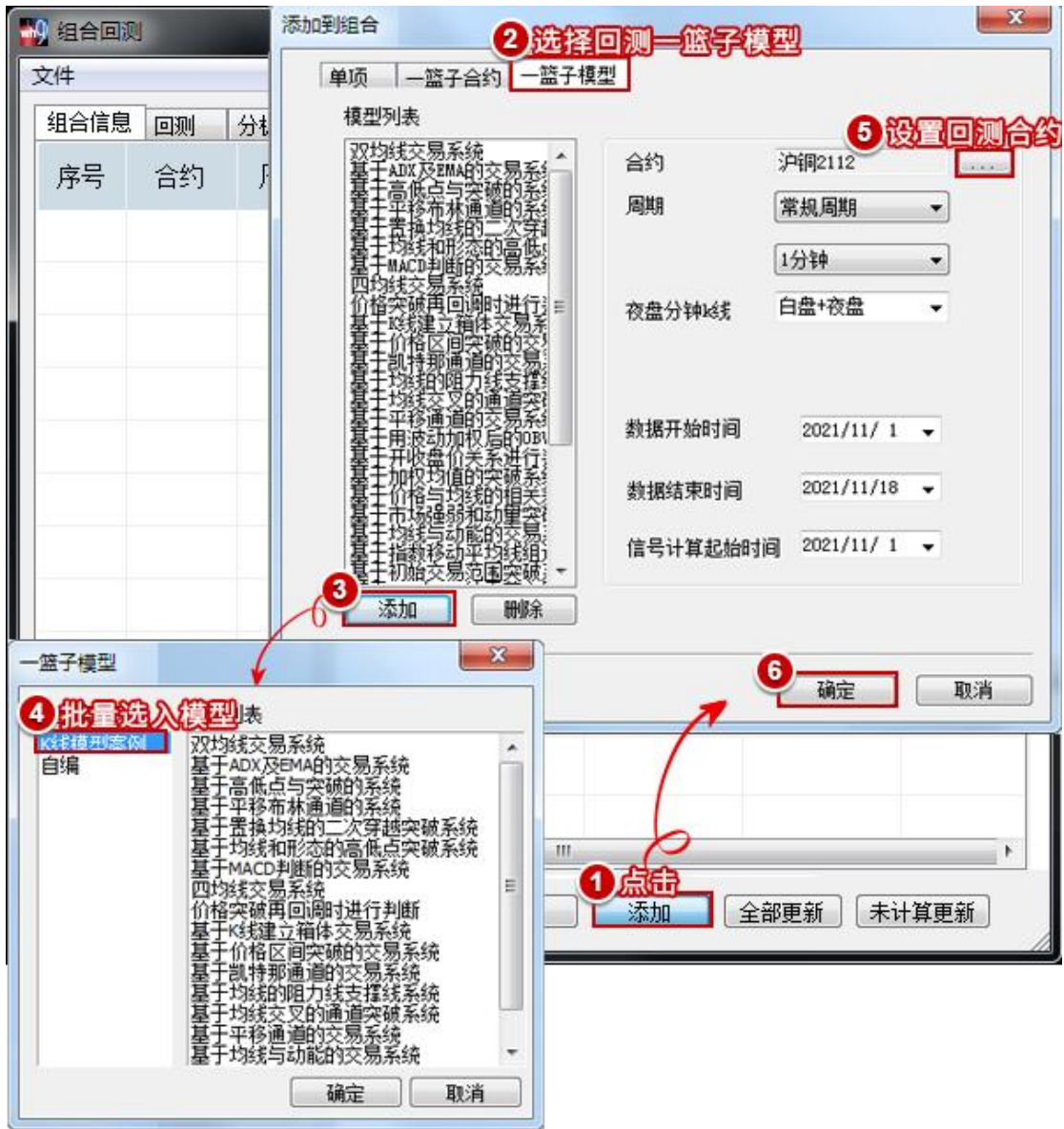


### 7、如何实现快速将同一合约同一周期加载不同模型进行测试？

答：利用组合回测->回测一篮子模型功能可实现该想法。

如下图，在组合测试界面点击右下角【添加】按钮，选择【一篮子模型】，在弹出的窗口中批量选入要加载的模型。在弹出窗口中设置一些选项，点击【确定】，可实现快速对同一合约同一周期加载不同模型进行测试。





## 8、期货主连链回测机制

<https://www.wenhua.com.cn/popwin/wh8zhulianlian.htm>

## 9、为什么我的模型不能进行参数优化

答：含有参数项的模型才可以进行参数优化（如下图左），如果在模型中指定了交易参数，则不能优化（如下图右）。

```
//可进行参数优化
```

```
Params
```

```
    Numeric N1(30);
```

```
Vars
```

```
    Numeric Stdd;
```

```
Begin
```

```
    Stdd = Std(Close, N1);
```

```
//不能参数优化
```

```
Vars
```

```
    Numeric Stdd;
```

```
Begin
```

```
    Stdd = Std(Close, 30);
```

### 三、策略运行常见问题

#### 1、T9 软件需要的电脑配置

答：（1）内存配置需 $\geq 8G$ ；

（2）CPU 要达到四线程及以上配置；

（3）操作系统 win7、win10、win2008 的 64 位版本；

（4）建议使用电信、联通宽带，其它宽带会有卡顿问题。

#### 2、股票持仓 T+0 增强交易机制说明

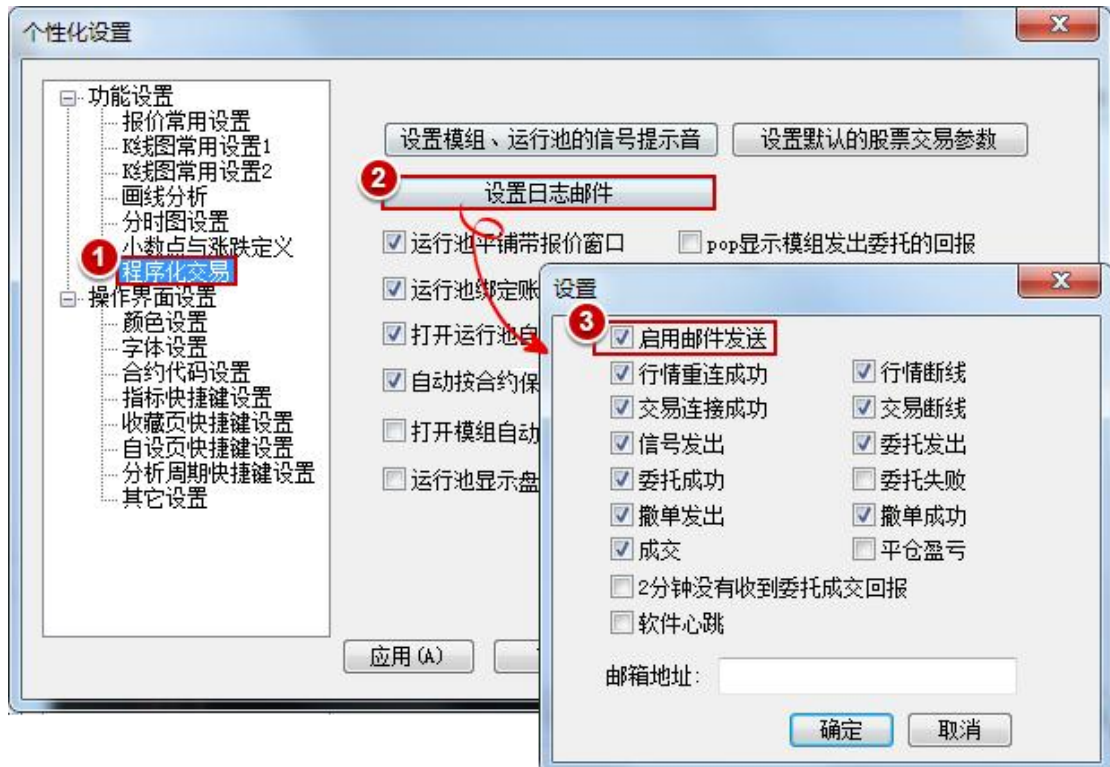
<https://www.wenhua.com.cn/popwin/T+0jiaoyichi.htm>

#### 3、期货合约运行模组机制说明

<http://www.wenhua.com.cn/popwin/mozuyxjz2.html>

#### 4、如果我不在电脑前，但想知道模型的运行情况怎么办

答：可以使用【日志邮件】功能，该功能可以帮助您接收到模型运行动态的邮件，人不在电脑前一样可以监控自己的程序。在软件右上方【系统工具】->【个性化设置】中按下图的操作步骤勾选启用后，盒子和模组的交易动态会直接发送至绑定的邮箱中。



### 5、模组里可以加载多少个运行单元？

答：模组的运行单元数量上限与 CPU 线程数挂钩，每一个线程允许 32 个运行单元，总数不超过 256。

### 6、模型修改后，让模组按照新模型源码执行，如何操作

模型编写平台上点击【更新】→【立即更新模组的公式】，即可按照新的源码执行模组。



注：

①模型修改并更新后，运行单元会清空历史信号，并按照新的公式来执行。但是会保留原有的理论资金和实际资金。

②模型更新后，运行单元第一个有效信号必须是开仓信号

### 7、T9 上如何加载期权量化策略

答：可将期权定价策略、期权市场波动套利策略等复杂的期权交易思路编写成算法交易模型，加载到算法交易运行池中后台运行下单。

### 8、T9 上如何加载股票 T+0 策略

答：如下图所示，点击右上角【量化】菜单->【股票持仓 T+0 增强交易】，填入交易池名称、资金、证券、模型和底仓限额等信息建立股票 T+0 交易池。



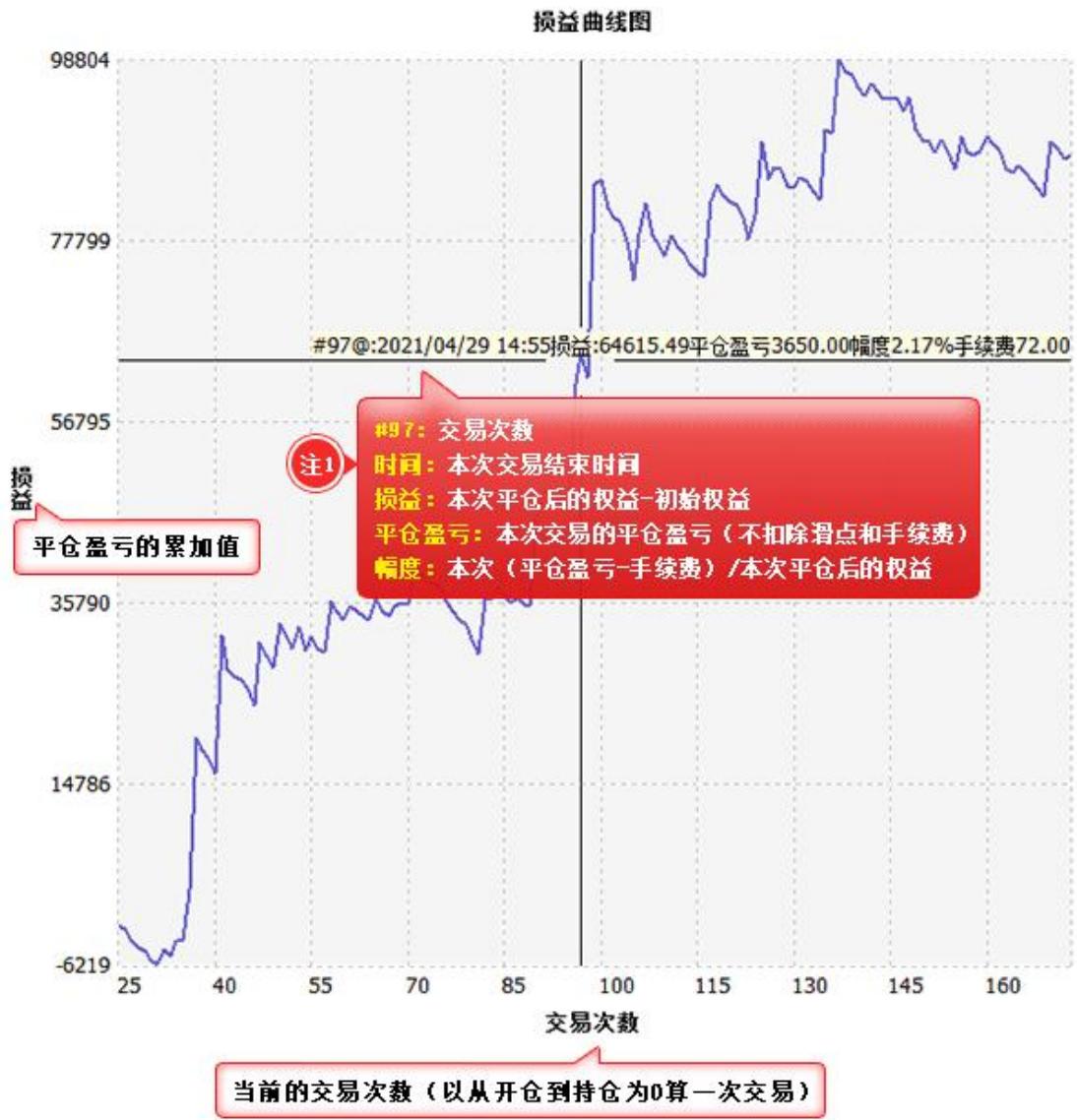
加载后，同一个交易池中的证券共用一份资金运行，一个运行单元调仓时释放的资金可用于其他运行单元买回证券底仓操作，如下图所示。



## 附录：回测报告的各项统计图表说明

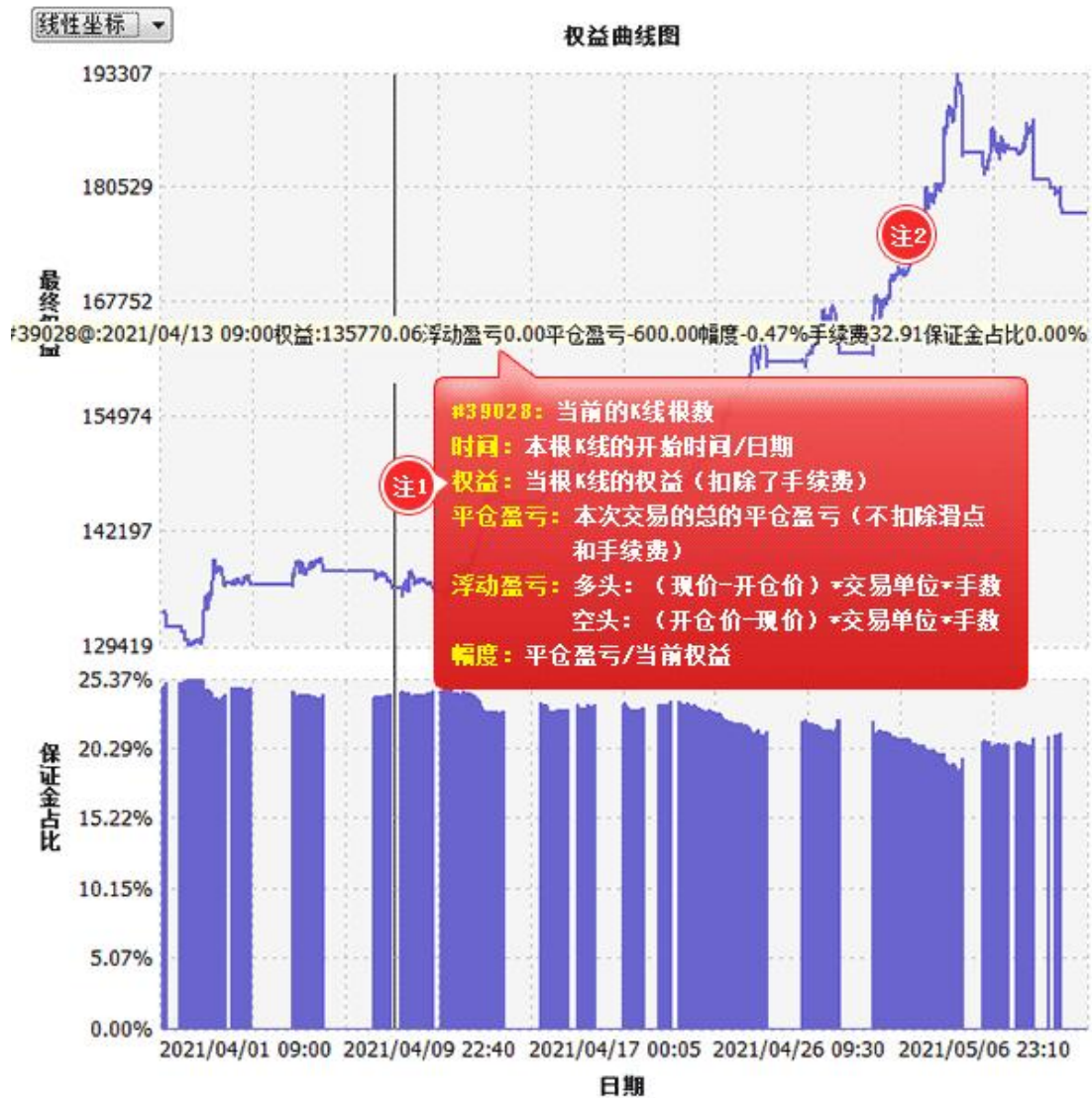
### 收益/风险

(1) 损益曲线图：策略的累计盈亏曲线，反映一段时间内平仓盈亏累加值的波动，每完整的交易一次计算一次损益值



**注1** 本次交易结束时间为最后的平仓信号所在K线的起始时间。

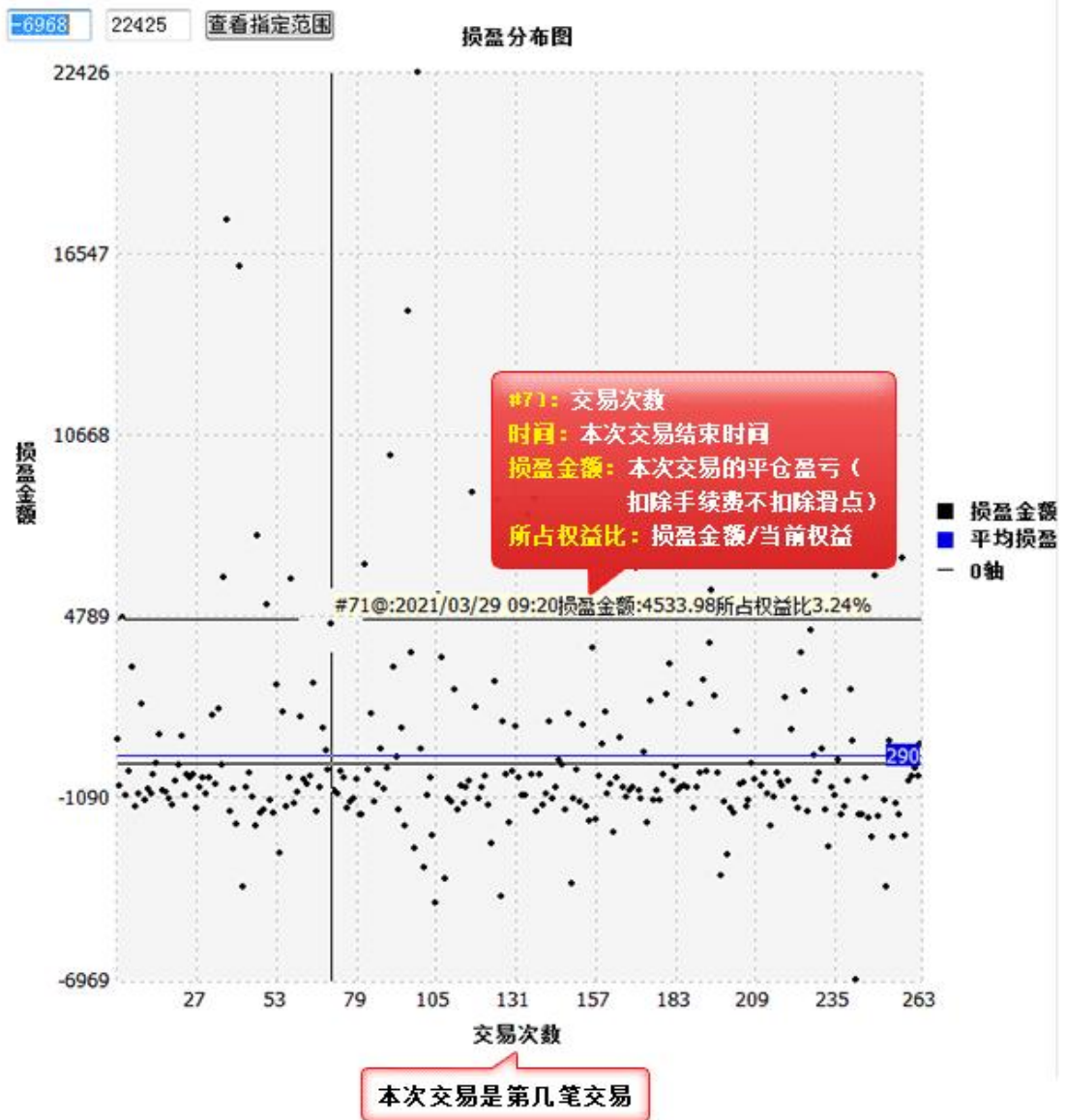
(2) 权益曲线图: 反映某一个时点的权益波动情况, 每根K线都动态的统计一次权益数值



注1 权益=初始权益值 + 平仓盈亏 + 浮动盈亏 - 手续费

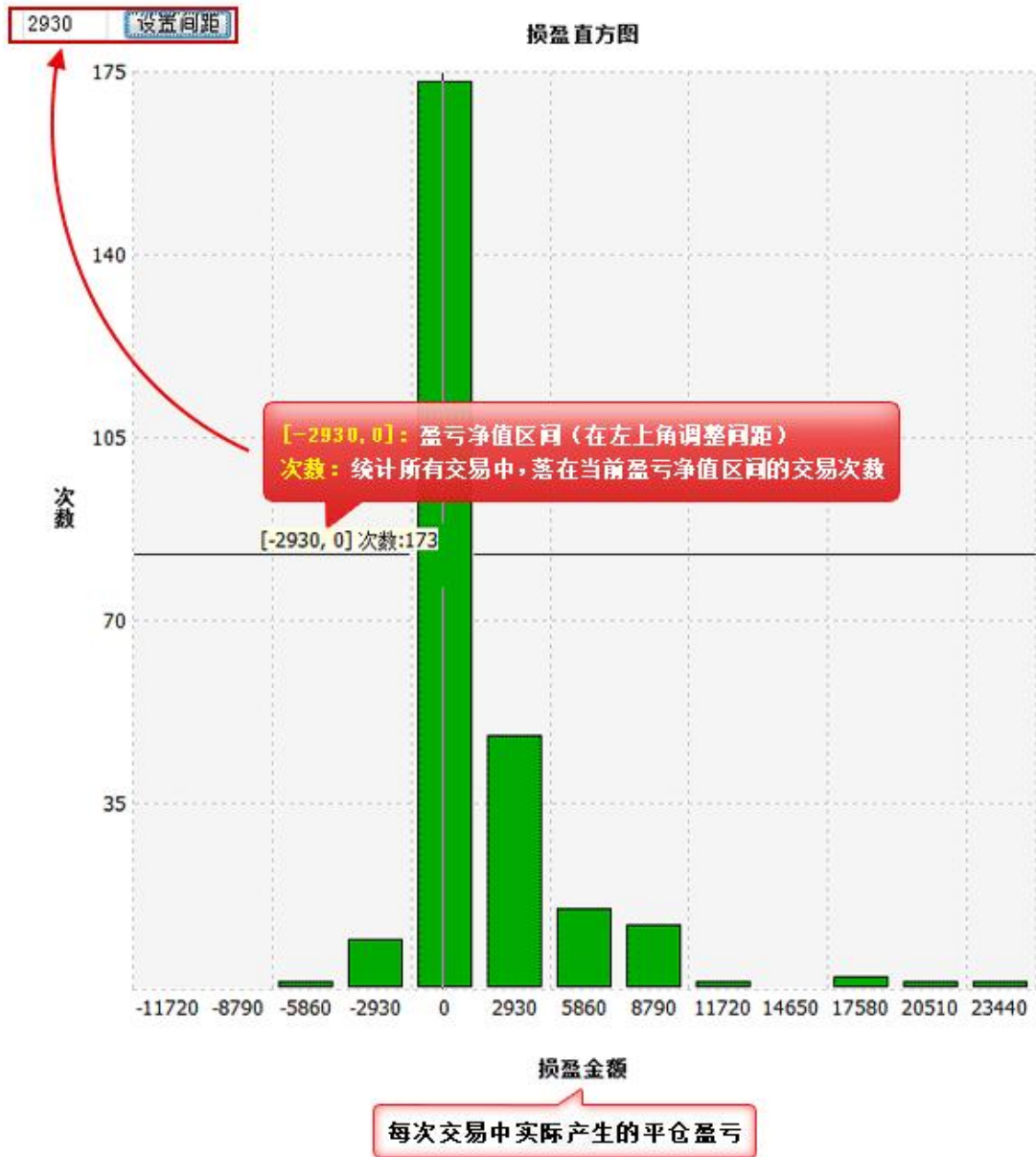
注2 如果一根k线上有多个信号, 取最后一个信号的权益值。

(3) 损盈分布图: 根据距离平均盈亏的离散程度分析交易的整体盈利情况

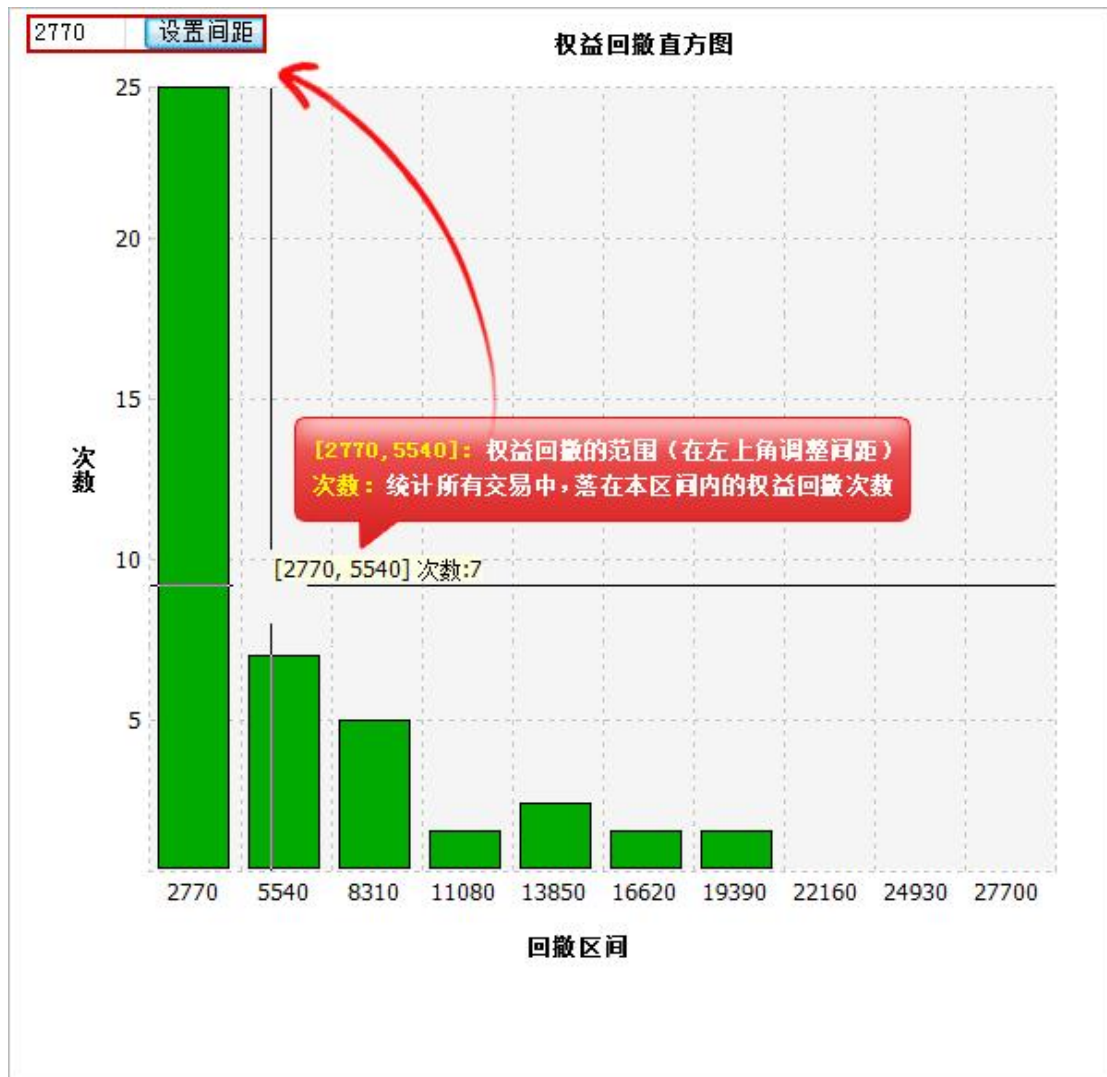


(4) 损益直方图: 统计落入盈亏净值区间内的次数, 对比盈利和亏损能力

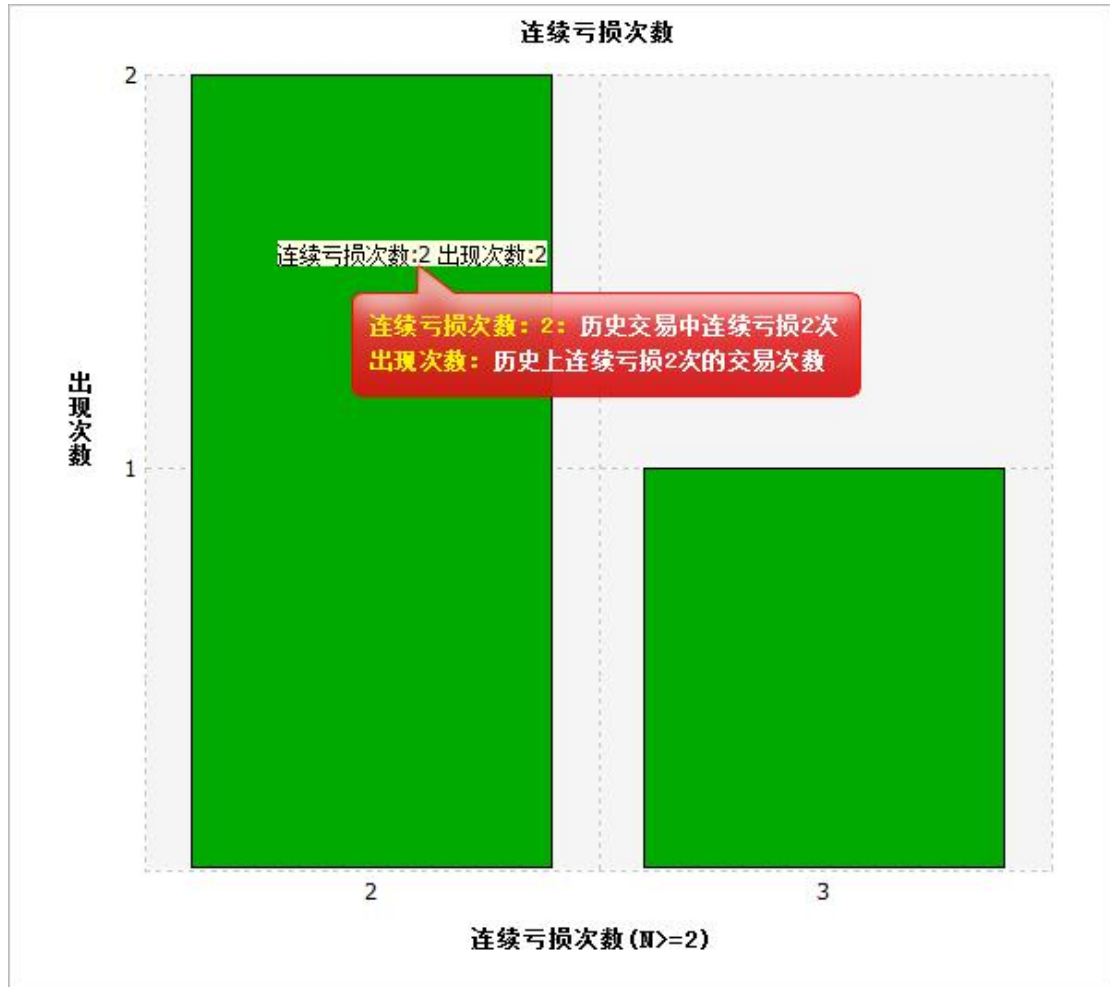




(5) 权益回撤直方图: 统计落入某一回撤区间的次数, 反映权益回撤的金额分布

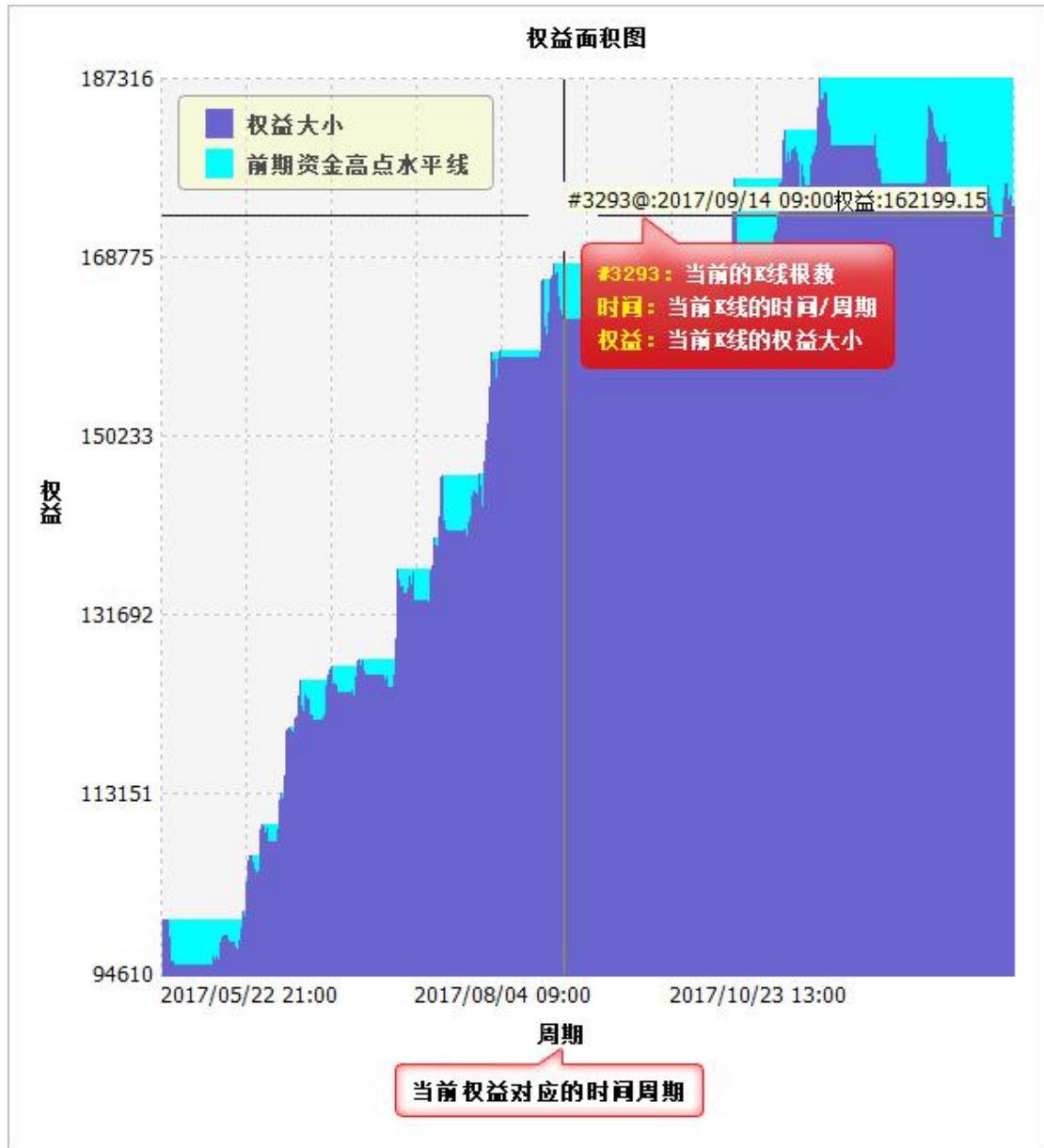


(6) 连续亏损次数: 统计连续亏损的次数, 投资者可结合自身对风险的承受能力综合考量策略的稳定性



### 浮动盈亏

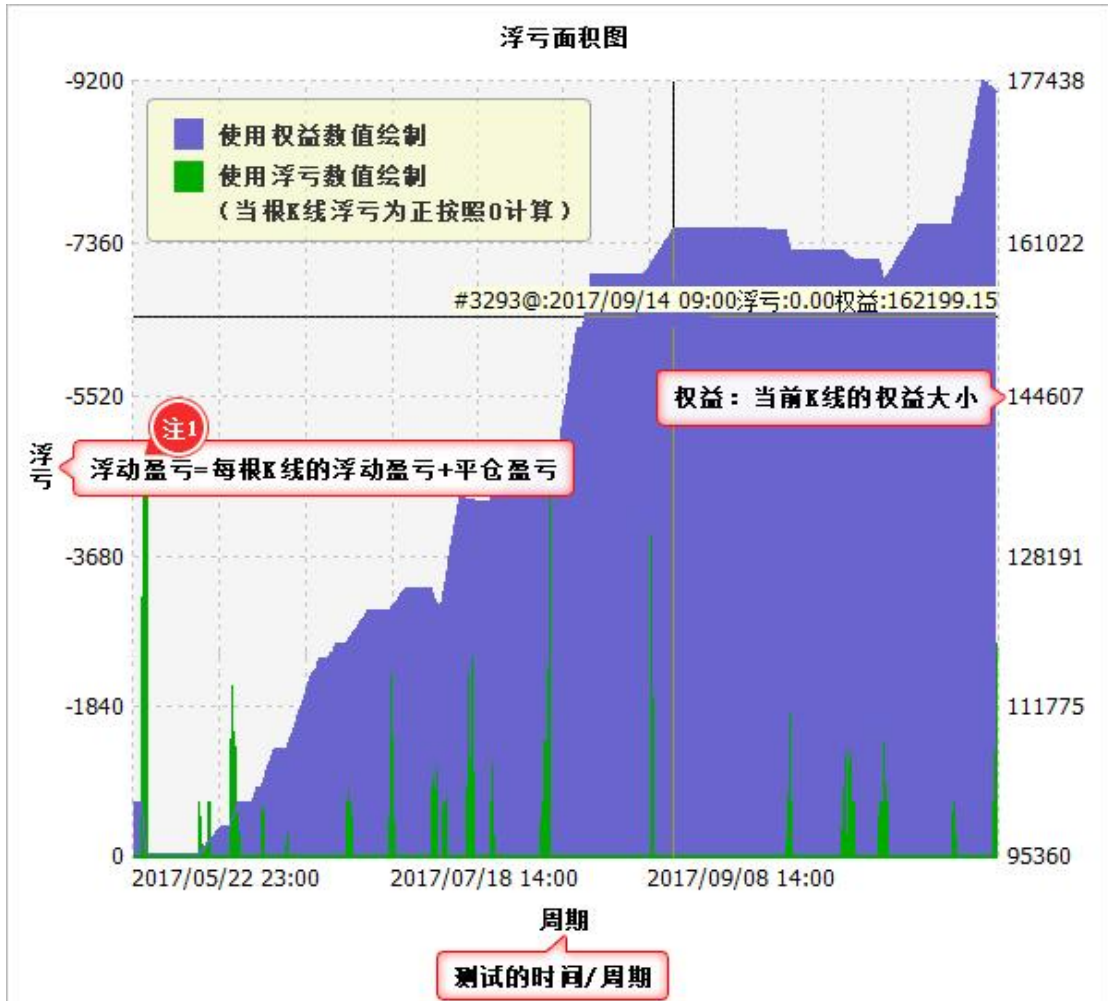
- (1) 权益面积图: 对比权益的高点位置和权益波动, 直观的查看权益回撤的幅度



(2) 浮盈面积图: 对比权益面积图, 分析浮盈波动情况



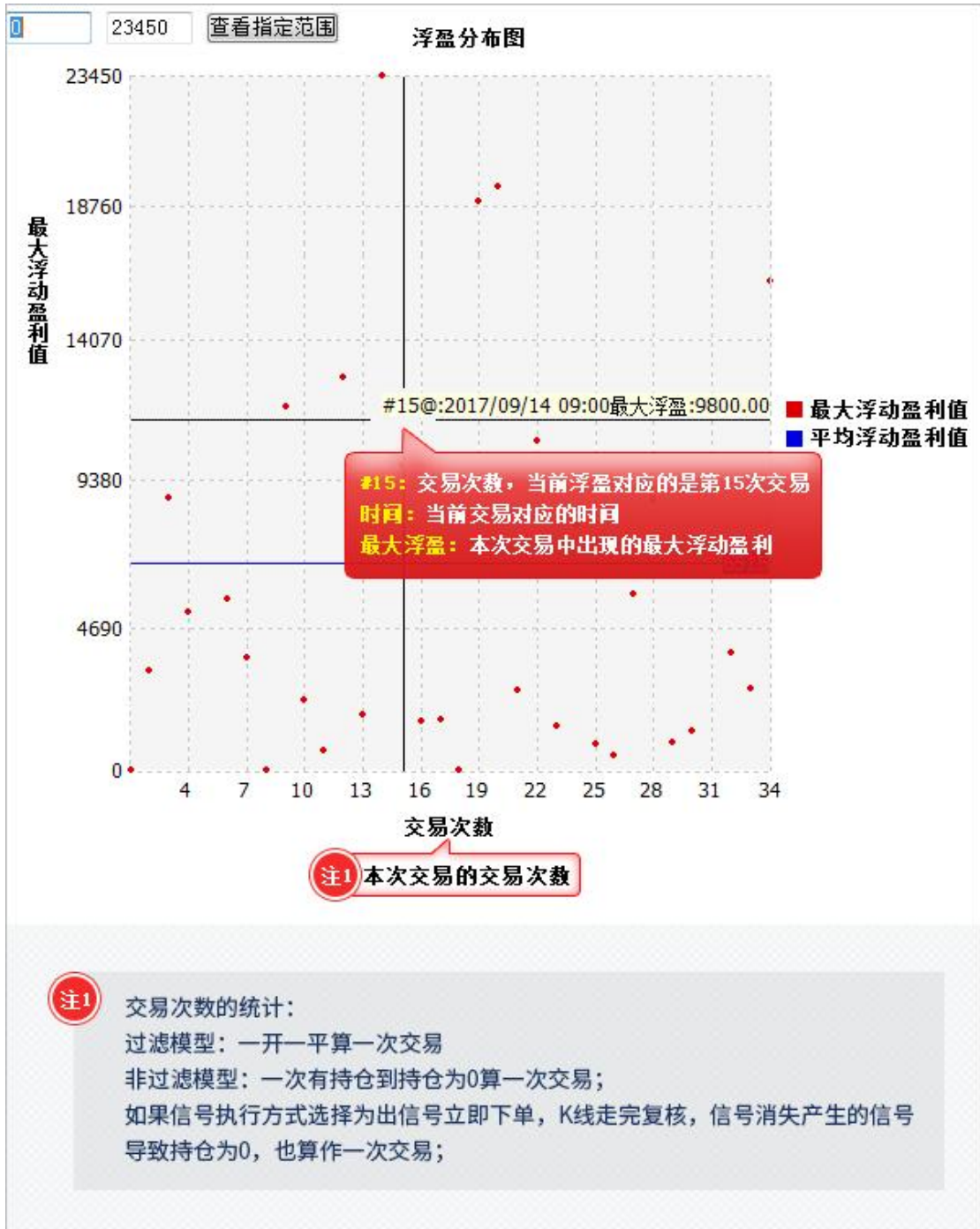
(3) 浮亏面积图: 对比权益面积图, 分析浮亏波动变化



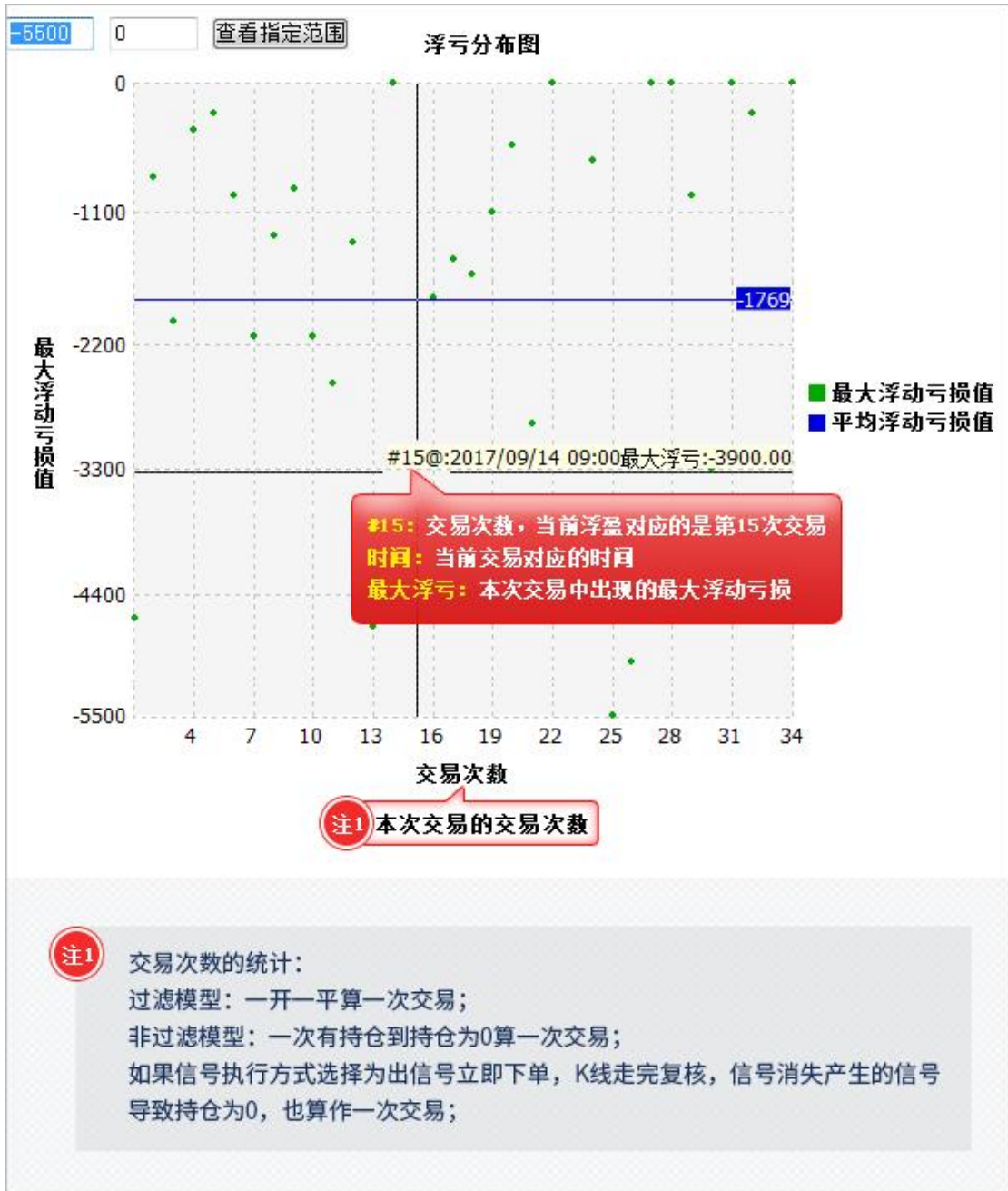
**注1** 浮动盈亏计算：（以开多为例）

- 1.K线走完确认信号下单：
  - 开仓信号当根K线：当根没有浮动盈亏；
  - 开仓信号后的K线：浮动盈亏=（收盘价-开仓均价）\*手数；
  - 平仓信号当根K线：平仓后持仓为0，浮动盈亏=平仓盈亏；
  - 平仓后仍有持仓，浮动盈亏=平仓盈亏+当根剩余持仓的浮动盈亏；
- 2.出信号立即下单，K线走完复核：
  - 开仓信号当根K线固定的信号，未发生信号消失：浮动盈亏=（收盘价-指令价）\*手数；
  - 当根K线发生过信号消失：浮动盈亏=平仓盈亏；

(4) 浮盈分布图：根据距离平均浮盈的离散程度分析交易的浮盈极值和整体盈利状态

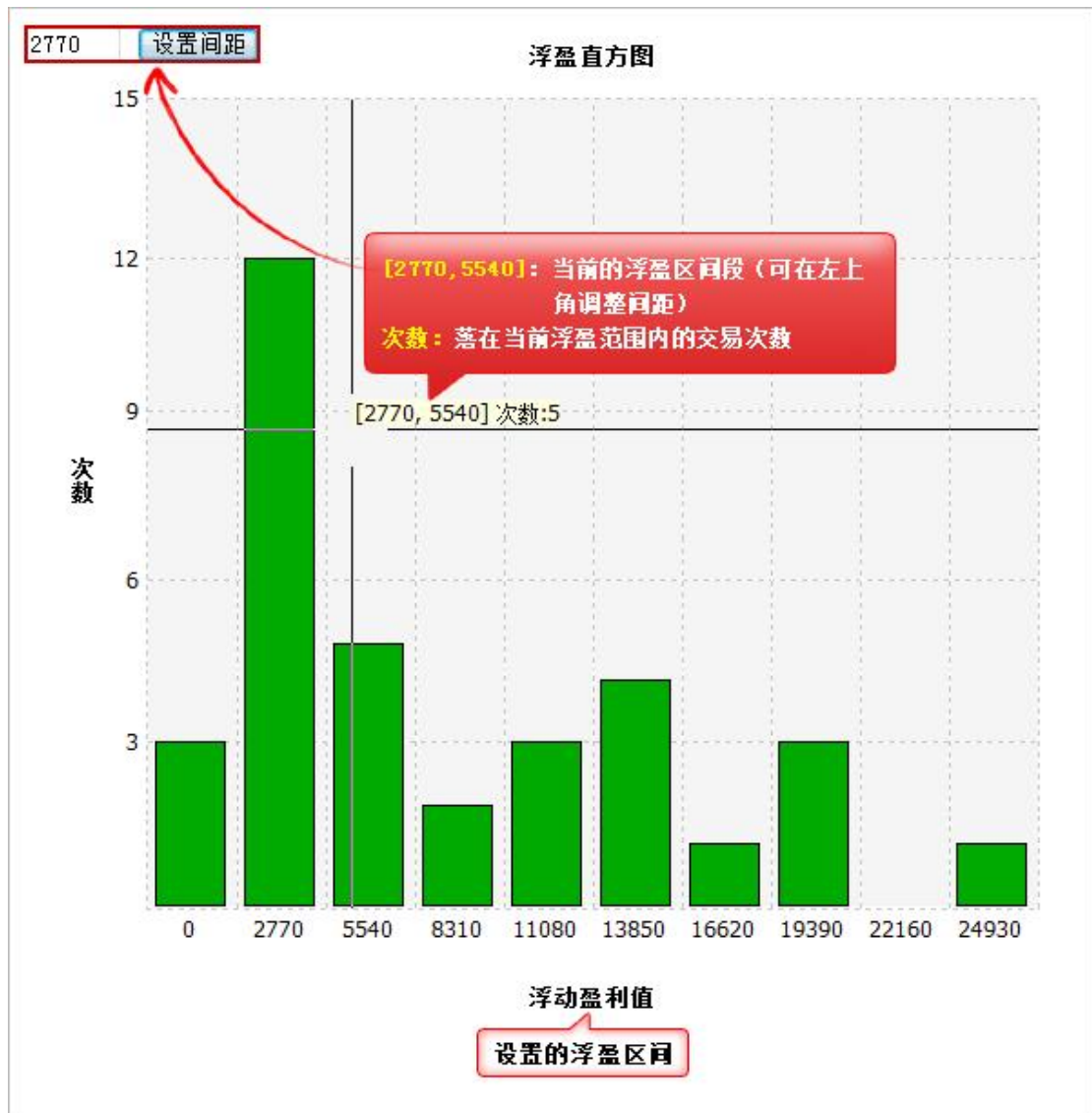


(5) 浮亏分布图: 根据距离平均浮动亏损值的离散程度分析浮亏的极值和整体亏损情况

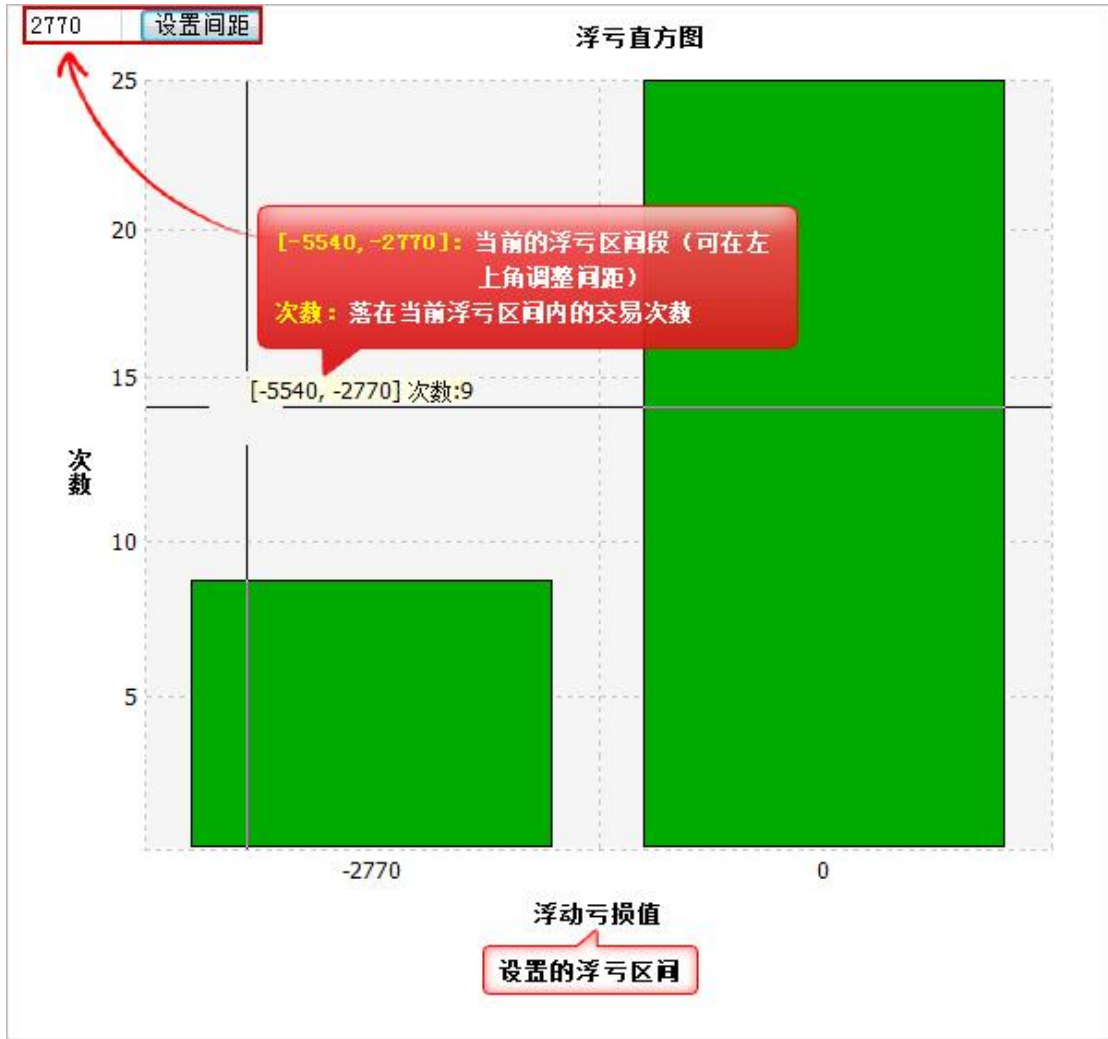


(6) 浮盈直方图：统计落在指定浮盈区间内的交易次数，分析策略的盈利能力



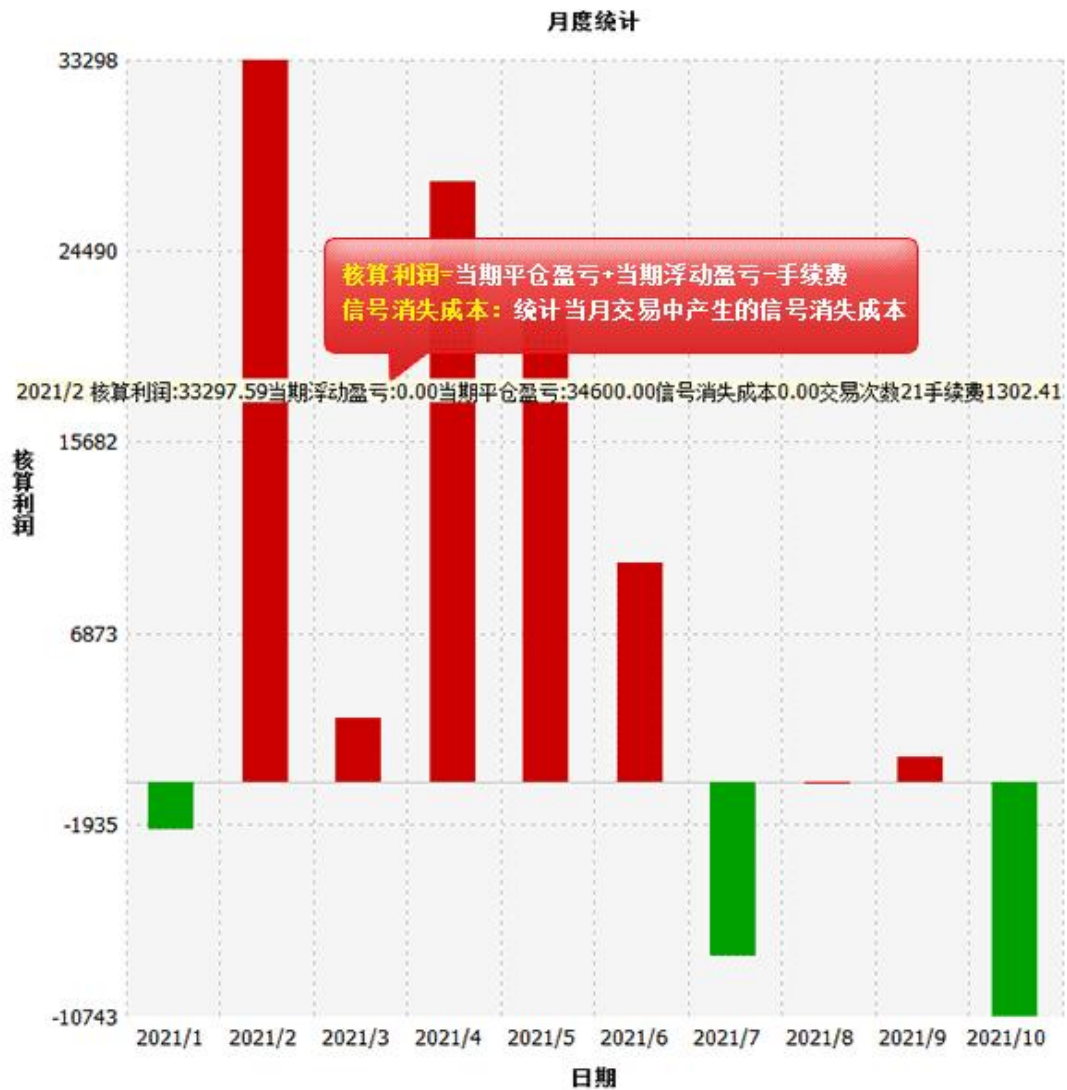


(7) 浮亏直方图: 统计落在固定浮亏区间内的交易次数, 分析策略的亏损程度



### 阶段分析

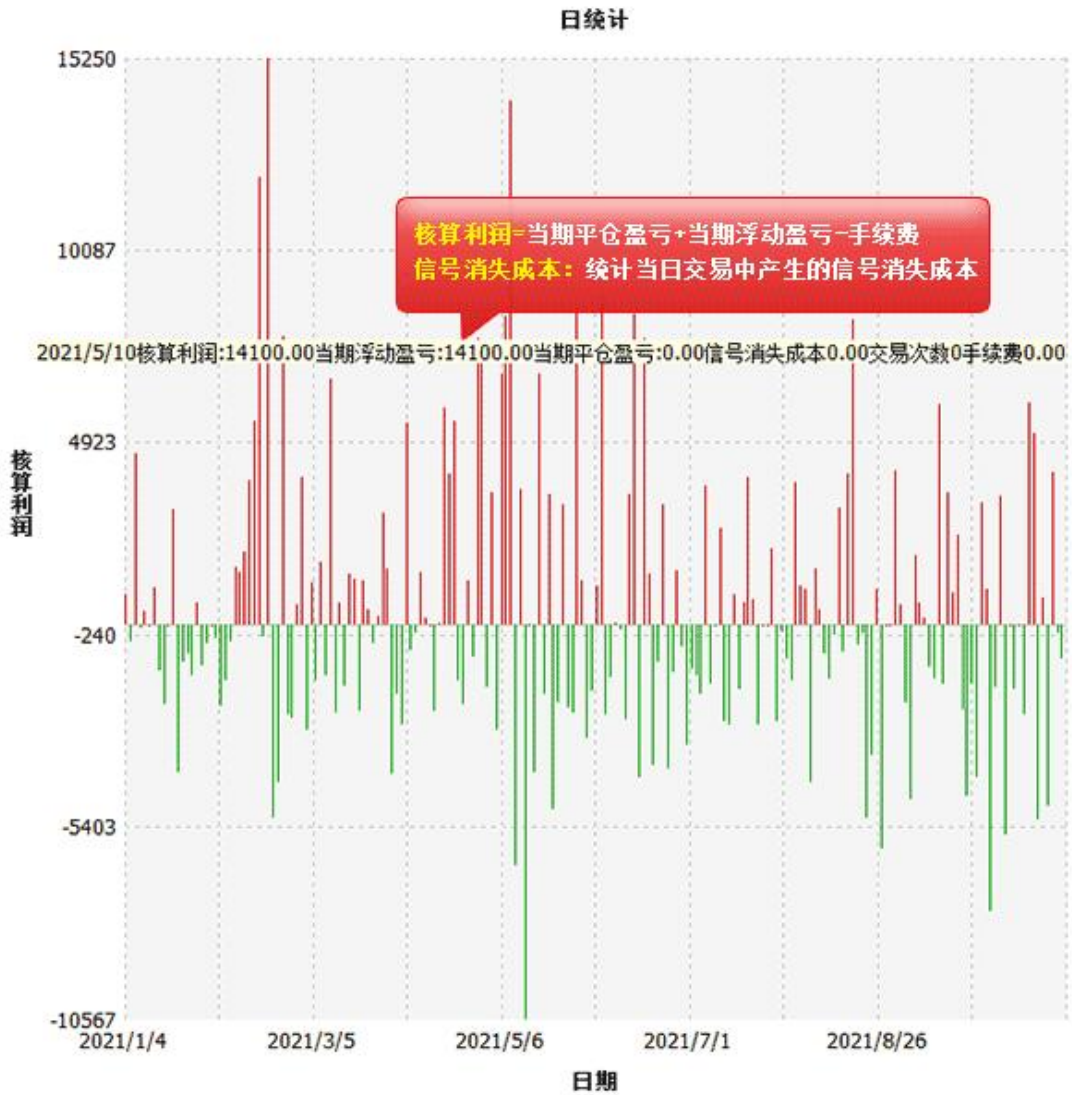
(1) 月度统计: 统计当月交易的盈亏和成本大小



**当期平仓盈亏：**当月起始无持仓，根据交易明细累加当月发生的平仓盈亏；当月起始有持仓，计算平仓盈亏时，起始持仓根据上月收盘价计算得到的开仓价格来计算平仓盈亏，月内的开仓则根据交易明细正常计算当期平仓盈亏；

**当期浮动盈亏：**当月收盘无持仓，当期浮动盈亏为 0；  
 当月收盘有持仓，根据当时收盘价为平仓价格计算当期浮动盈亏；

(2) 日统计：统计当日的交易的盈亏和成本大小

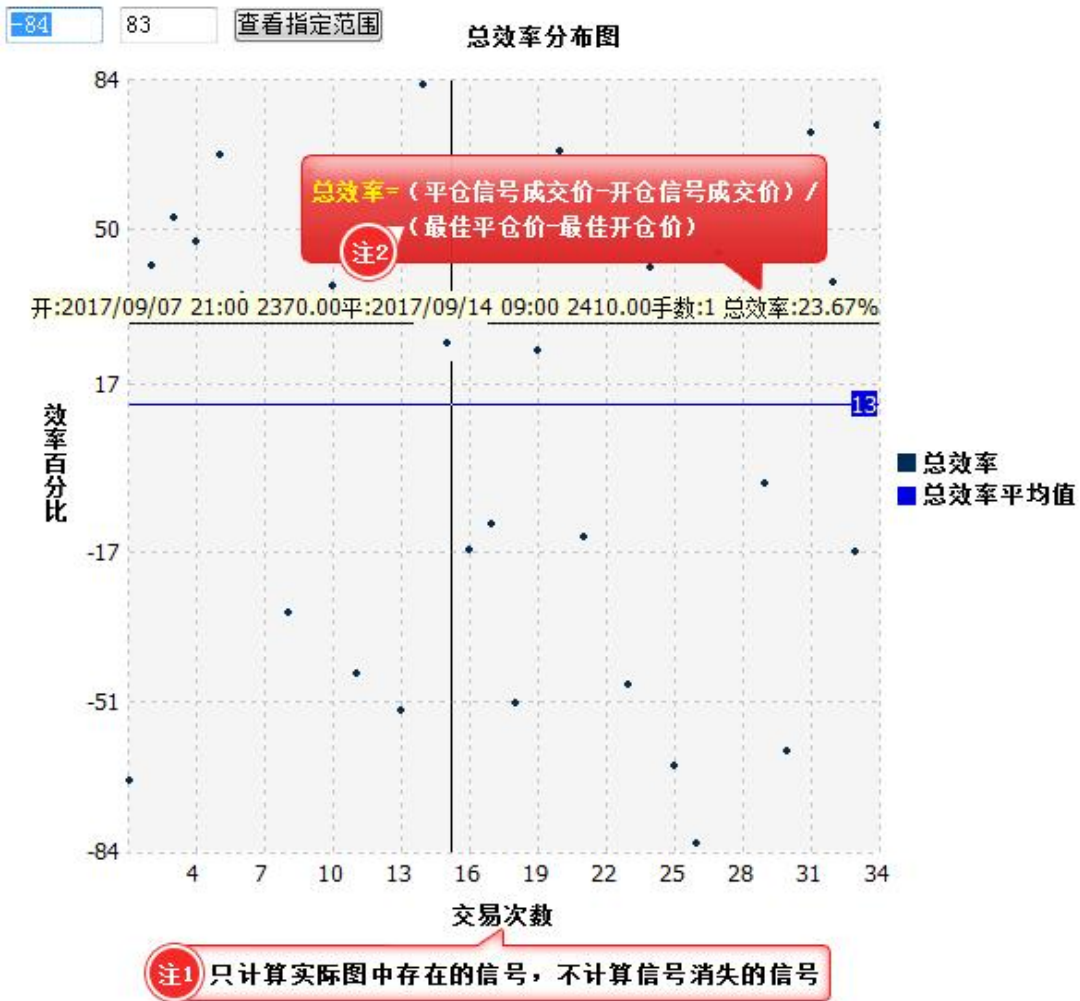


**当期平仓盈亏:** 当日起始无持仓, 根据交易明细累加当日发生的平仓盈亏; 当日起始有持仓, 计算平仓盈亏时, 起始持仓根据昨日收盘价计算得到的开仓价格来计算平仓盈亏, 日内的开仓则根据交易明细正常计算当期平仓盈亏;

**当期浮动盈亏:** 当日收盘无持仓, 当期浮动盈亏为 0;  
当日收盘有持仓, 根据当时收盘价为平仓价格计算当期浮动盈亏;

### 效率分析

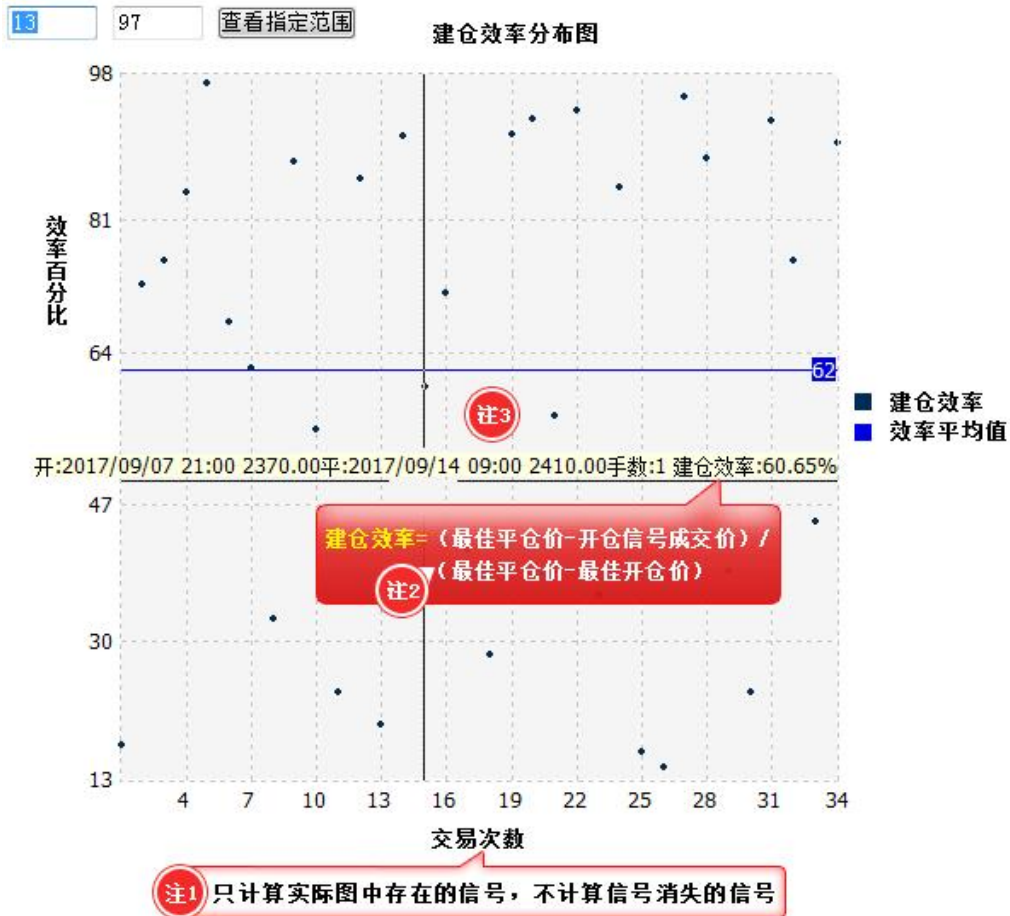
(1) 总效率分布图: 以做多为例, 买在最低点卖在最高点视效率为 100%, 按交易价格折算每次交易的效率



注1 过滤模型：开平对应，一开一平算作一次交易。  
非过滤模型：平仓信号次数对应的开仓信号次数计算交易次数。例如：  
一个平仓信号对应着一个开仓信号的仓位，则此次平仓计算为1次交易，  
一个平仓信号对应着两个开仓信号的仓位，则此次平仓计算为2次交易；

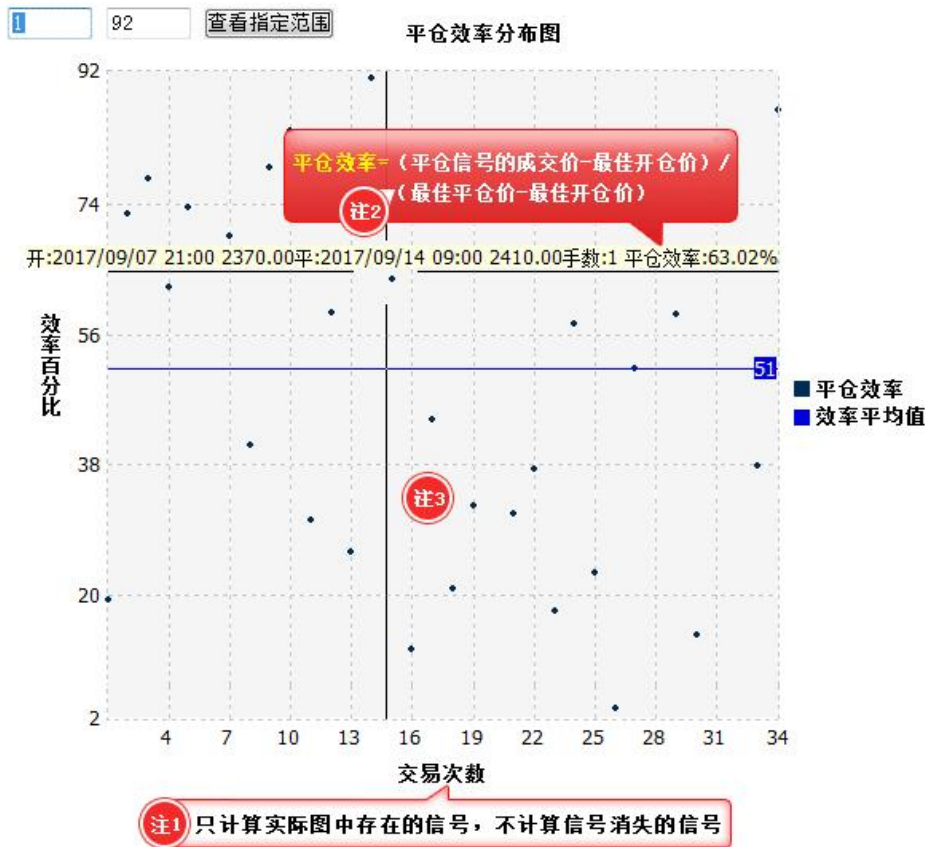
注2 最佳平仓价、开仓价：（以做多为例）  
最佳平仓价位 本次交易开仓到下次开仓间的最高价（包含开、平仓当根K线）  
最佳开仓价位 上次交易平仓到本次平仓间的最低价（包含开、平仓当根K线）

(2) 建仓效率分布图：以做多为例，买在最低点即为最佳入场点，效率为 100%，再按实际入场点折算每次建仓的效率。



- 注1** 过滤模型：开平对应，一开一平算作一次交易。  
非过滤模型：平仓信号次数对应的开仓信号次数计算交易次数。例如：  
一个平仓信号对应着一个开仓信号的仓位，则此次平仓计算为1次交易，  
一个平仓信号对应着两个开仓信号的仓位，则此次平仓计算为2次交易；
- 注2** 最佳平仓价、开仓价：（以做多为例）  
最佳平仓价位 本次交易开仓到下次开仓间的最高价（包含开、平仓当根K线）  
最佳开仓价位 上次交易平仓到本次平仓间的最低价（包含开、平仓当根K线）
- 注3** 该图表不支持信号执行方式选择为出信号立即下单，不进行复核使用，切换到图表时有弹出框提示。

(3) 平仓效率分布图：以做多为例，卖在最高点即为最佳出场点，效率为 100%，再按实际出场点折算每次平仓的效率。



**注1** 过滤模型：开平对应，一开一平算作一次交易。  
非过滤模型：平仓信号次数对应的开仓信号次数计算交易次数。例如：  
一个平仓信号对应着一个开仓信号的仓位，则此次平仓计算为1次交易，  
一个平仓信号对应着两个开仓信号的仓位，则此次平仓计算为2次交易；

**注2** 最佳平仓价、开仓价：（以做多为例）  
最佳平仓价位 本次交易开仓到下次开仓间的最高价（包含开、平仓当根K线）  
最佳开仓价位 上次交易平仓到本次平仓间的最低价（包含开、平仓当根K线）

**注3** 该图表不支持信号执行方式选择为出信号立即下单，不进行复核使用，切换到图表时有弹出框提示。

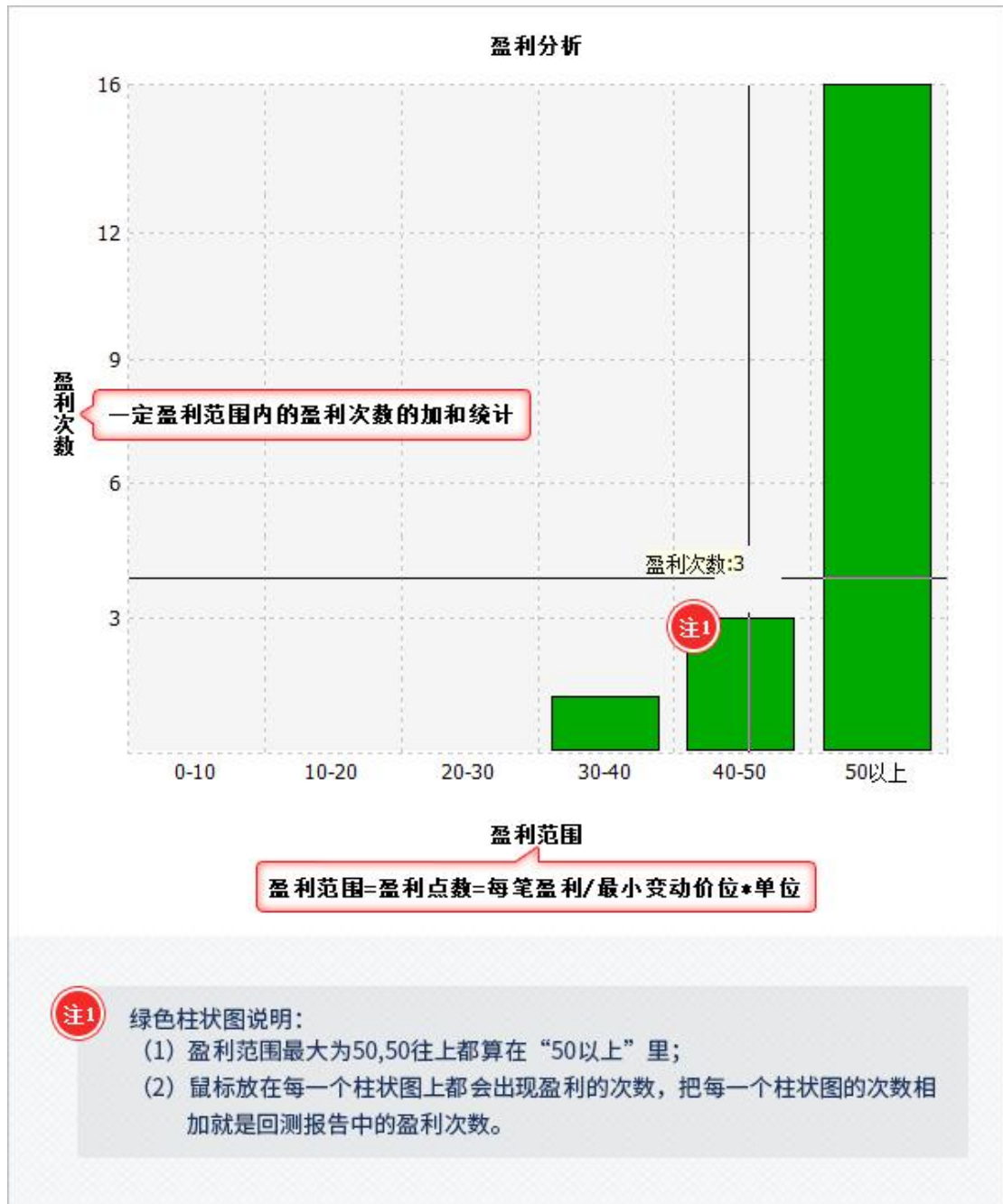
## 盈亏分析

(1) 胜率分析：统计不同盈利百分比下的胜率分布

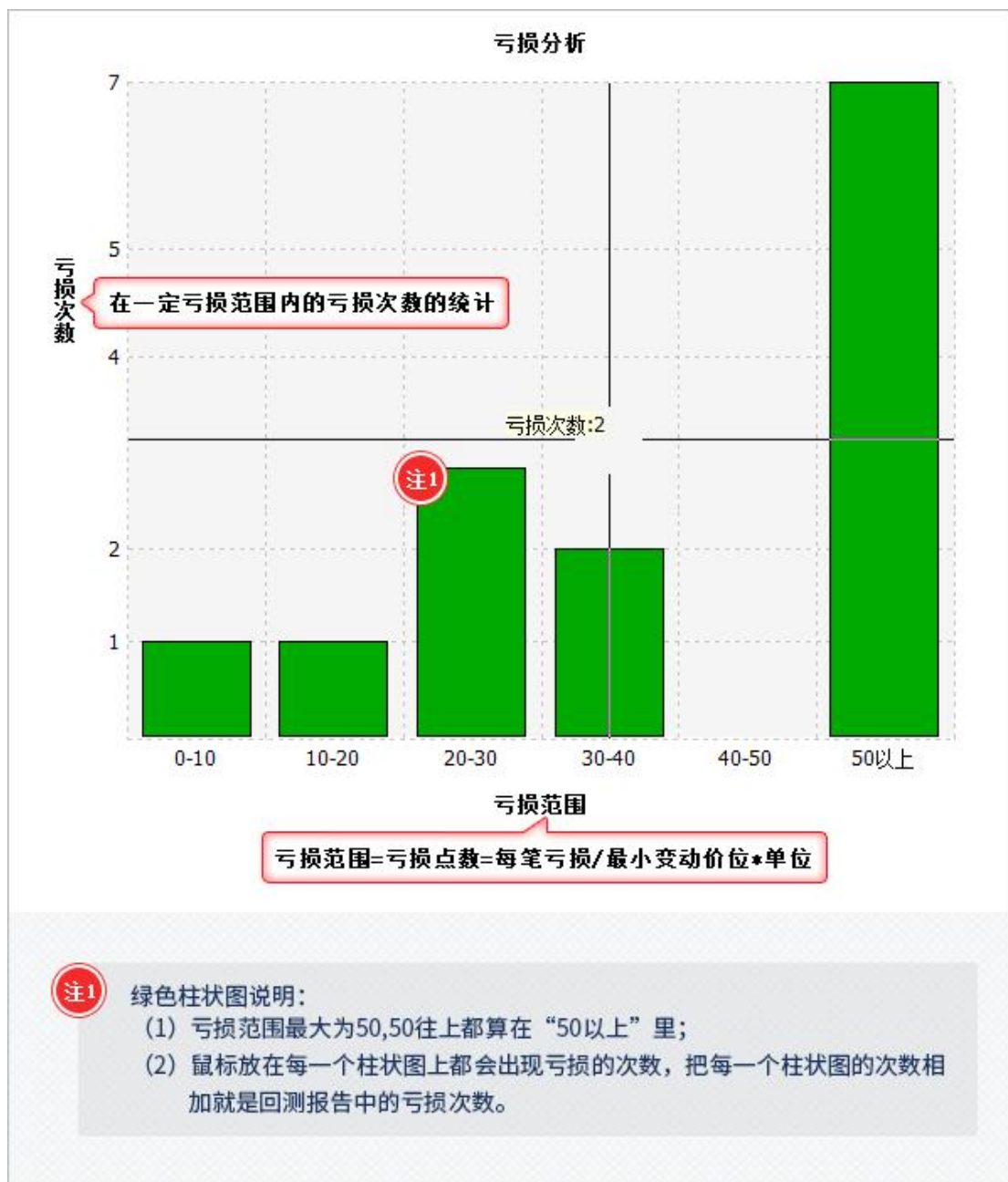


(2) 盈利分析：统计一定盈利范围内的盈利次数

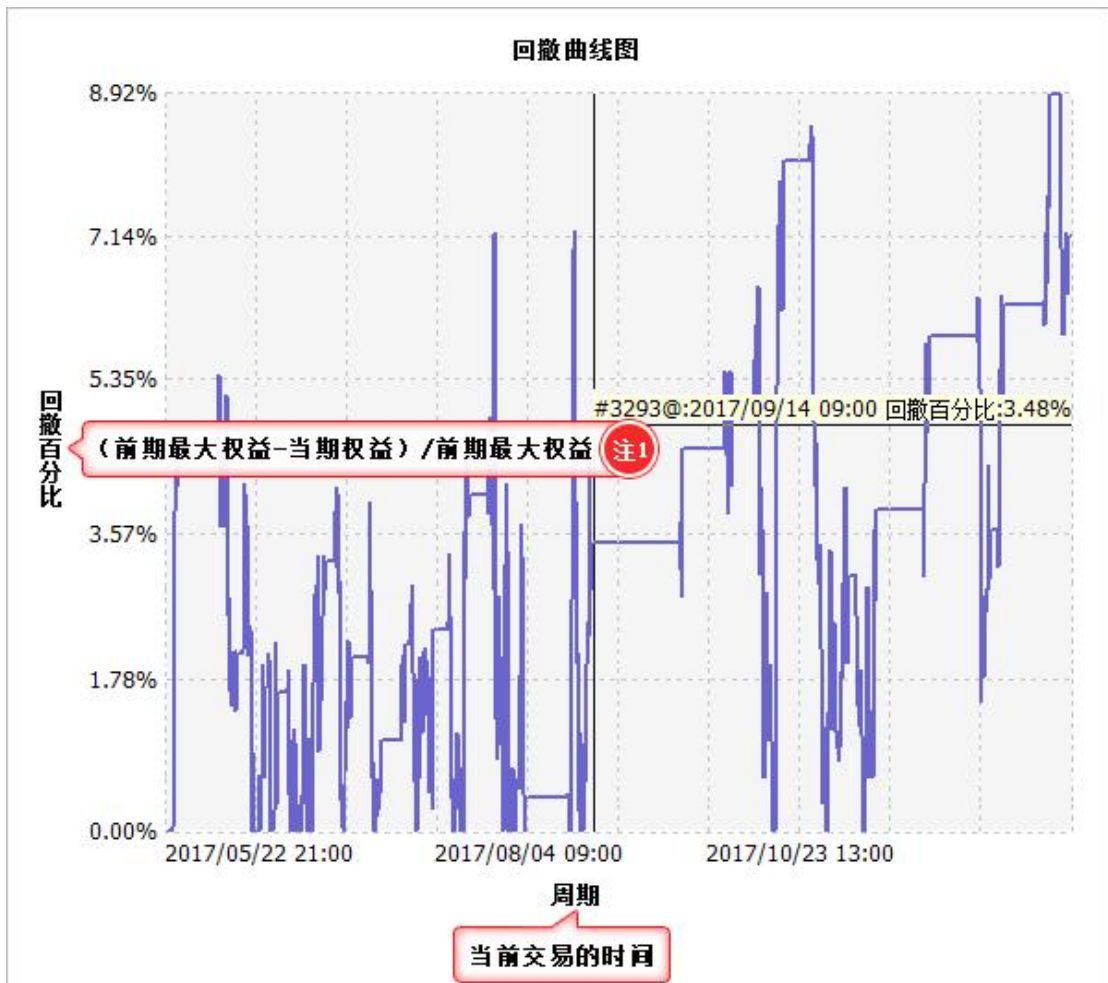




(3) 亏损分析：统计一定亏损范围内的亏损次数



(4) 回撤分析：按回撤比例分析权益回撤幅度



**注1** 回撤百分比说明：回撤百分比是按回撤比例的最大值来计算的，而不是以回撤的最大金额计算。  
例如：初始资金10w元，先发生了一个2w的回撤，交易一段时间后，总金额增长到了40w，又发生了一个4w的回撤。如果以回撤金额来计算，是后面这个4w的回撤大，但是以回撤百分比来计算，则是2w的回撤大。