



价差交易小班课

vn.py社区第7次小班课

By 陈晓优 2020-6-20



首先，做好心理准备

你一定会有很多内容听不懂的，不用怀疑是自己笨，因为
你旁边的人也听不懂，把我放在刚入行交易时也同样听不
懂



但，这就是做量化交易的本质

在未知的领域向前不断探索，每一次解决新的问题，有所
突破，最终都会是你交易盈利的来源



如果只能教会大家一样东西

我希望是：遇到某个问题时（不止在量化交易中），如何
寻找所需的资源以及解决这个困难的能力（以及决心）

课程的目标

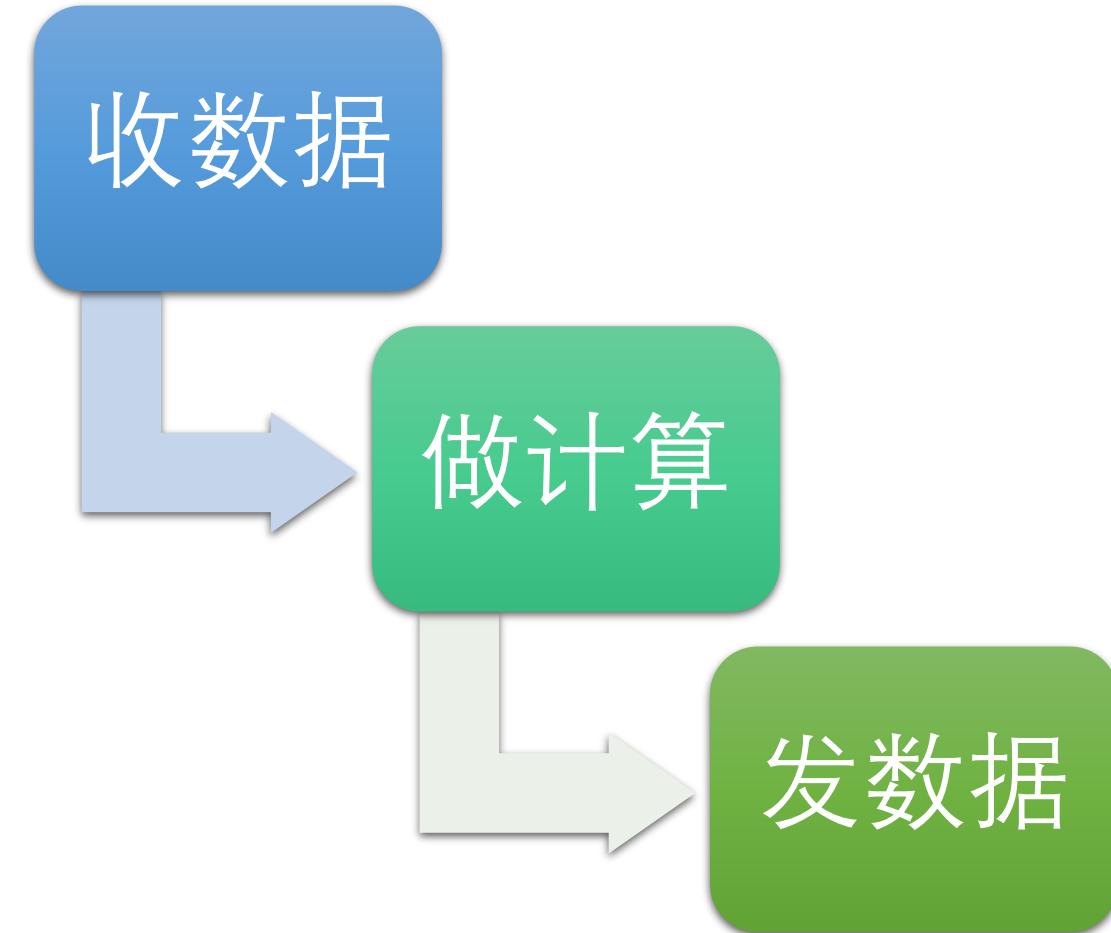
理解价差交易的优势不足

掌握价差算法的代码细节

学习寻找价差的研究方法

尝试价差进阶策略的开发

交易系统的本质



准备数据服务

国内期货

- 注册RQData试用账户

数字货币

- 注册BitMEX实盘账户
- 注册币安实盘账户



全球各大金融市场

数字货币

期货

股票

外汇

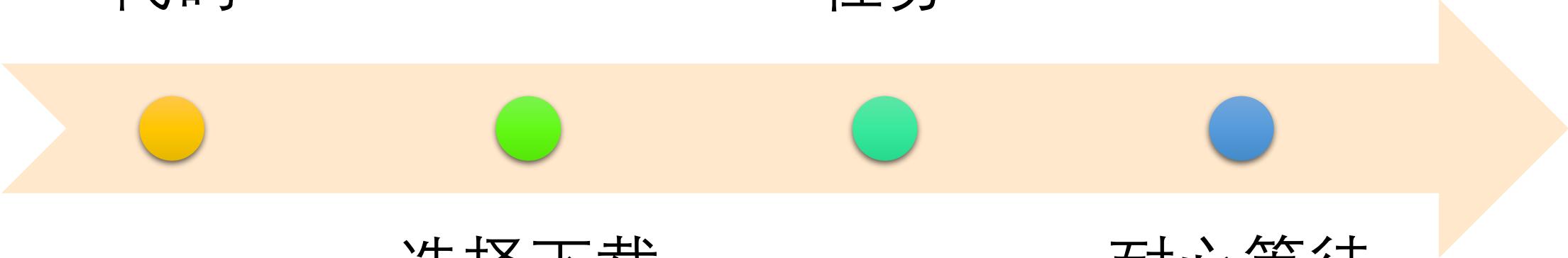
期权

债券

获取数据

确定合约
代码

开始下载
任务



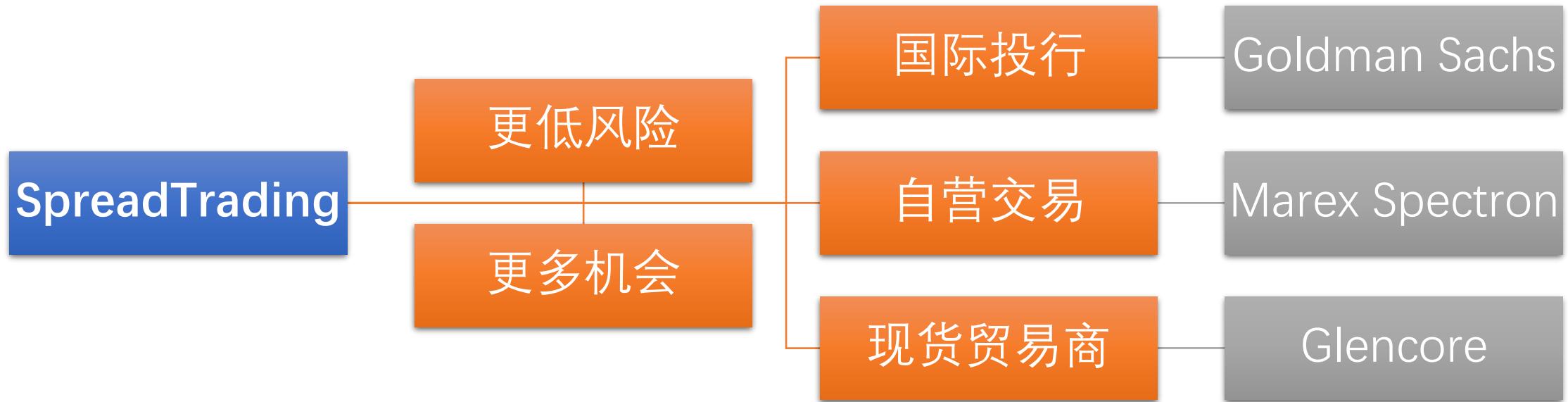
选择下载
时间段

耐心等待
完成

什么是价差交易

起源于1980年代的Mongan Stanley, 价差交易 (SpreadTrading) , 又名配对交易 (Pair Trading) , 是一种同时买入和卖出2个以上合约, 并通过两者价格间的差值 (而非价格本身) 来获利的交易方式。

大机构的主流选择



交易所 vs 本地

交易所价差

- 交易所提供
- 保证同时执行
- 举例：大商所豆粕跨期
- 优势：交易方便、执行风险低

本地价差

- 本地程序计算
- 可能出现瘸腿
- 举例：沪深300期现
- 优势：灵活定义、更多机会

保证金的优势

价差的波动
范围更小

整体价差持
仓风险更低

交易所收取
更少保证金

可以用更大
的杠杆交易

一个简单的例子

T时刻

- IF2006位于4000
- IF2007位于4010
- 远期升水较小

开仓操作

- 买入1手IF2007
- 卖出1手IF2006
- 以10做多价差

T+1时刻

- IF2006位于3900
- IF2007位于3960
- 升水恢复正常

平仓操作

- 卖出1手IF2007
- 买入1手IF2006
- 以60平仓价差

盈利情况

- IF2006: 100
- IF2007: -50
- 总盈亏: 50

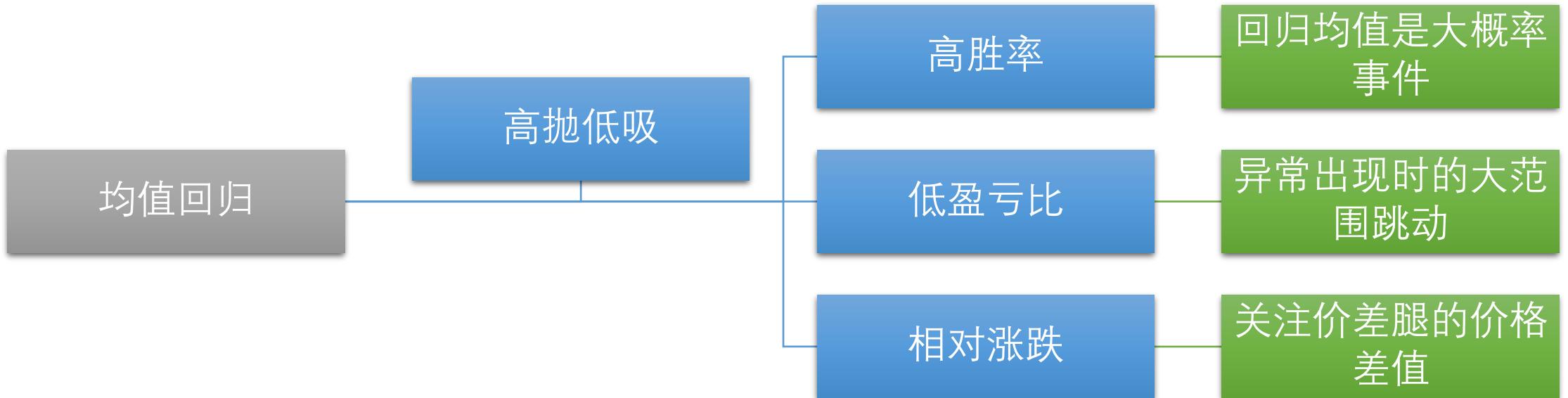
从图像上看

DOLLAR NEUTRAL PAIR TOOL

Stock Data	Long	Short
Symbol	BAC	WFC
Name	BANK OF AMERICA	WELLS FARGO & CO
Last Trade	\$15.57	\$28.69
Today's Volume 2nd Average Vol.	44,396,605 227,411,800	8,769,923 57,150,700
Market Cap (MM)	\$154,909	\$147,083
Beta to S&P	2.14	1.83
1 Year Return	30.20%	14.20%
52-Week High	\$19.86	\$34.25
52-Week Low	\$11.27	\$21.57
Dividend Ex-Date	06/02/2010	N/A
Dividend Amount	\$0.01	N/A
Next Earnings Date	07/16/2010	07/22/2010
Fundamentals		
LTM P/E	N/M	17.20X
Forward P/E	15.10X	14.30X
PEG Ratio	N/M	N/M
Dividend Yield	0.30%	0.70%
Price / Book	N/M	1.38X

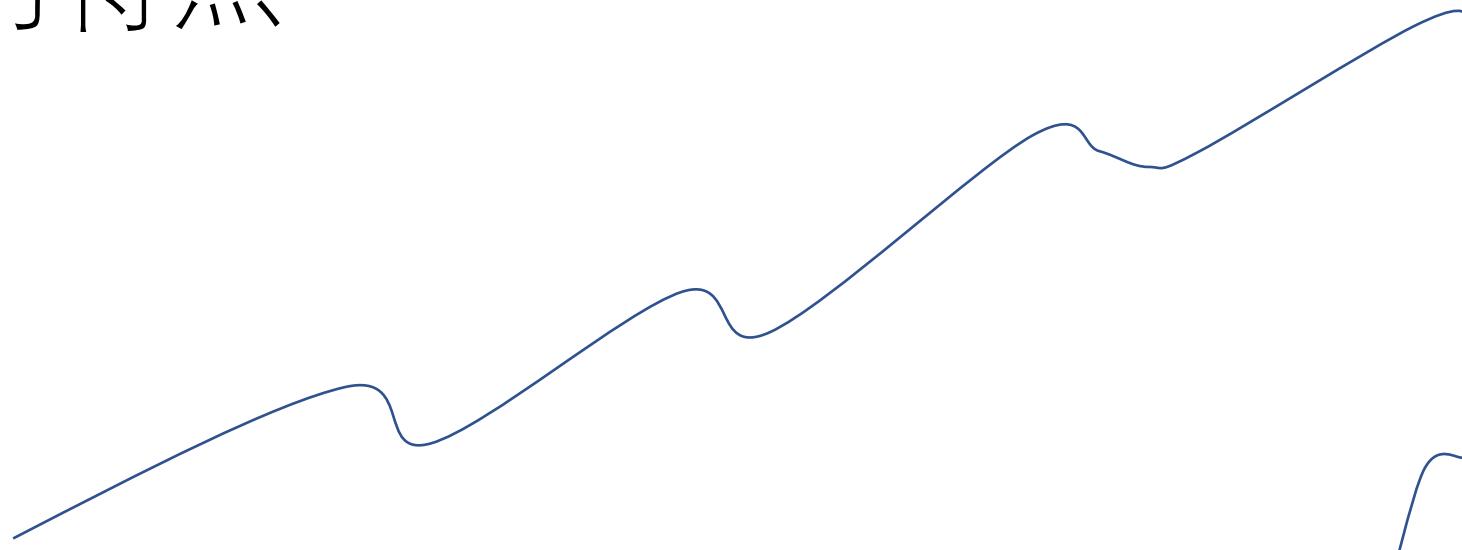


价差交易的特点

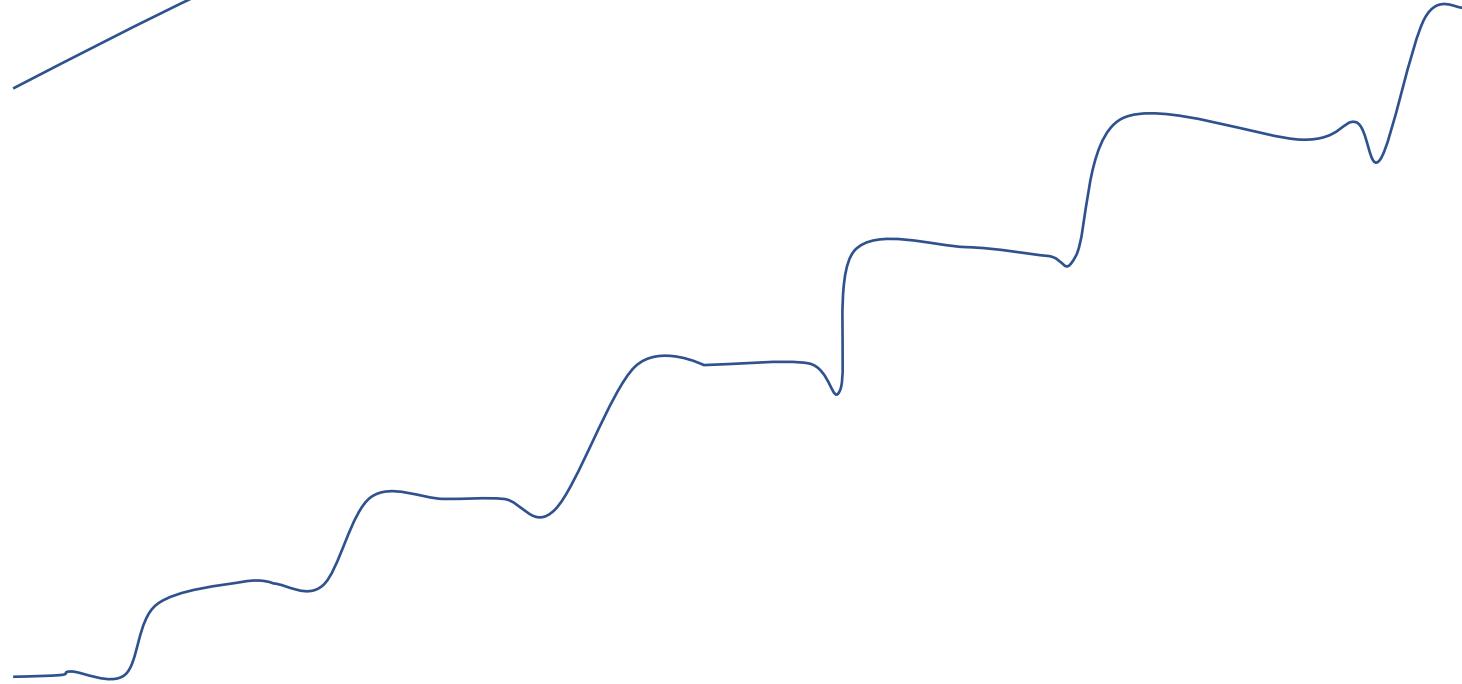


曲线的特点

价差交易



CTA策略



CTA策略的特点

趋势跟踪

追涨杀跌

低胜率

高盈亏比

绝对涨跌

什么是价差

描述性的定义

由2个或者2个以上的合约构成的组合

组合中多和空方向都分别至少有一个合约

合约之间必须存在某种关系：统计面或基本面

组合关系



沪深300期现套利



价格乘数



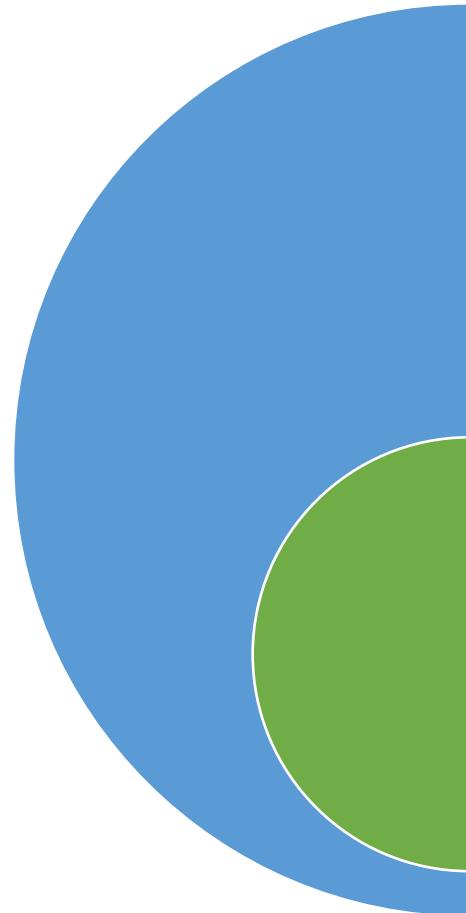
期货腿

现货腿

- IF2006
- 价格: 3998
- 价格乘数: -1

- 510300
- 价格: 4.011
- 价格乘数: 1000

交易乘数

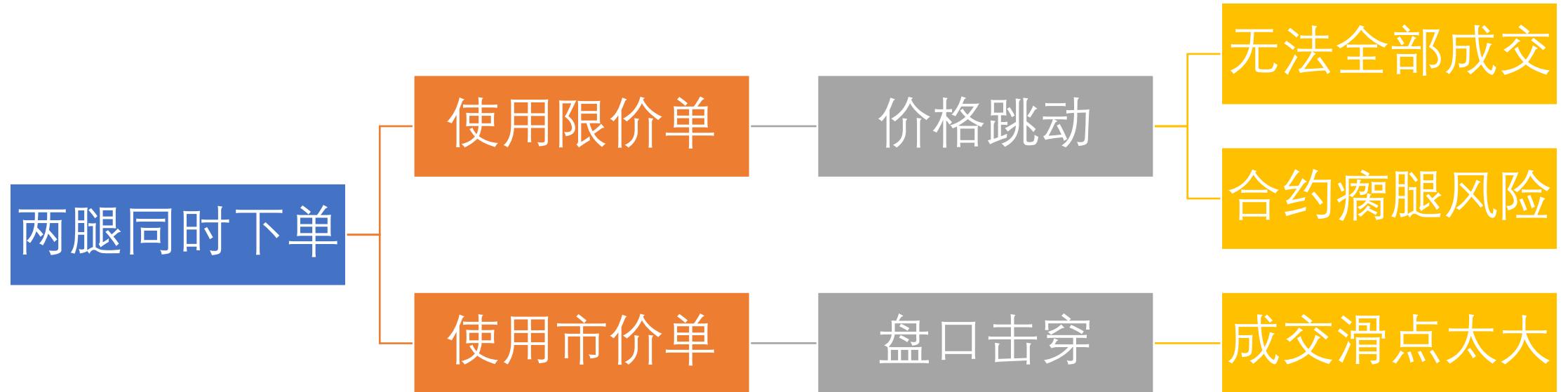


期货腿

现货腿

- IF2006
 - 价格: 3998
 - 合约大小: 300
 - 交易乘数: -1
-
- 510300
 - 价格: 4.011
 - 合约大小: 1
 - 交易乘数: 300000

核心问题



更好的方案

价差价格

- 等条件



发主动腿

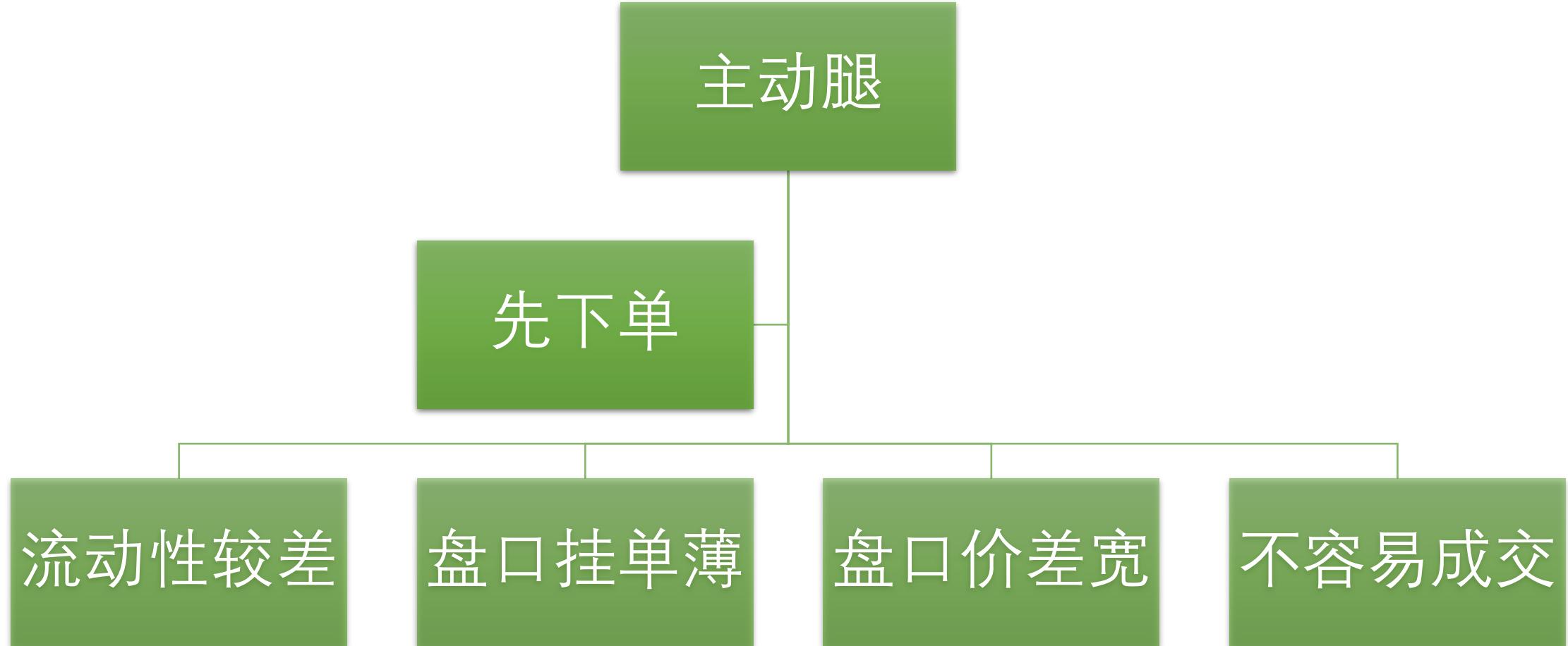
- 先成交



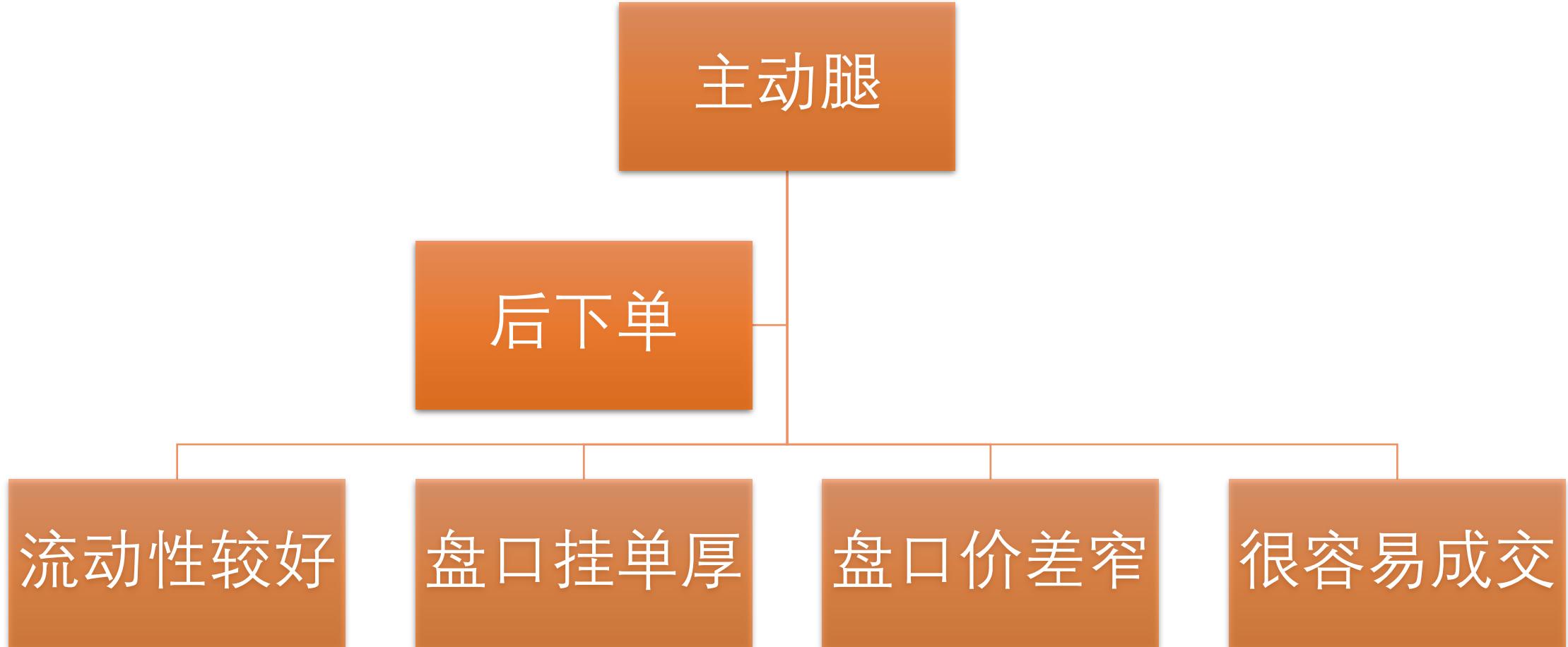
发被动腿

- 再对冲

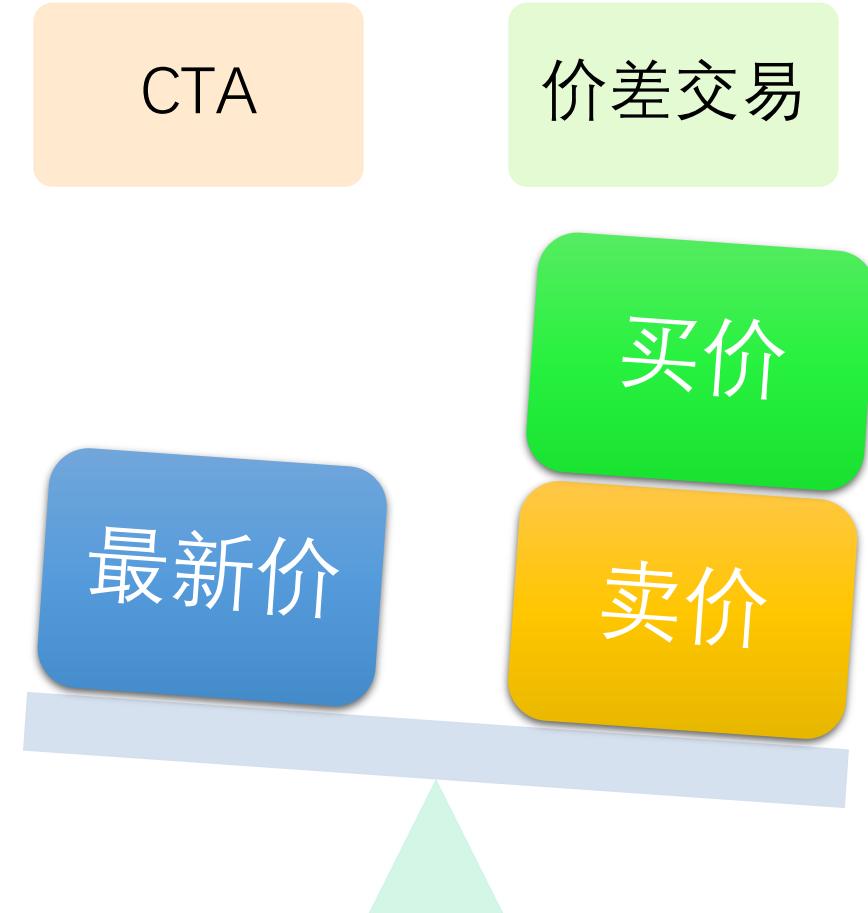
主动腿



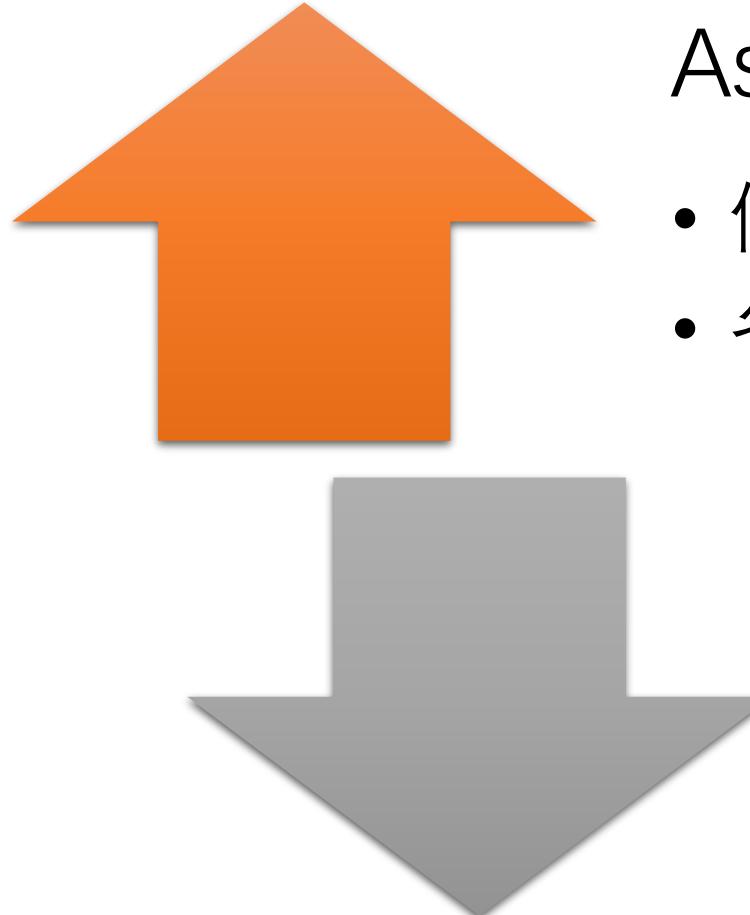
被动腿



价差盘口



盘口的计算



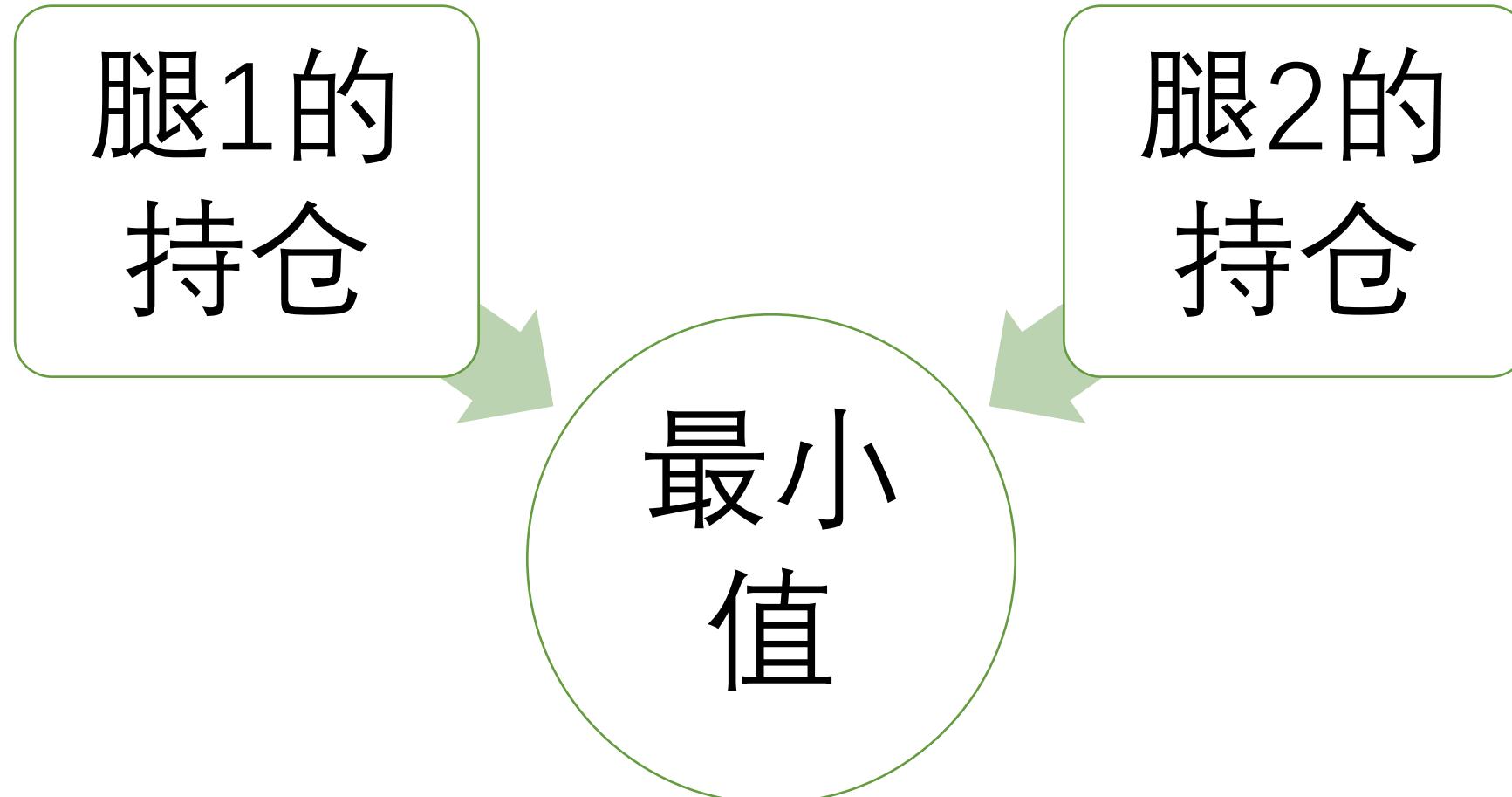
Ask 卖价

- 做多价差所需付出的成本
- 各条腿均立即以对价成交

Bid 买价

- 做空价差所需付出的成本
- 各条腿均立即以对价成交

价差持仓



用数字来举例 - 1

期货腿

IF2006

买价：3998

卖价：3998.2

价格乘数：-1

现货腿

510300

买价：4.011

卖价：4.015

价格乘数：1000

价差组合

300价差

买价：17

卖价：12.8

用数字来举例 - 2

期货腿

IF2006

持仓: -3

价格乘数: -1

现货腿

510300

持仓: 120万

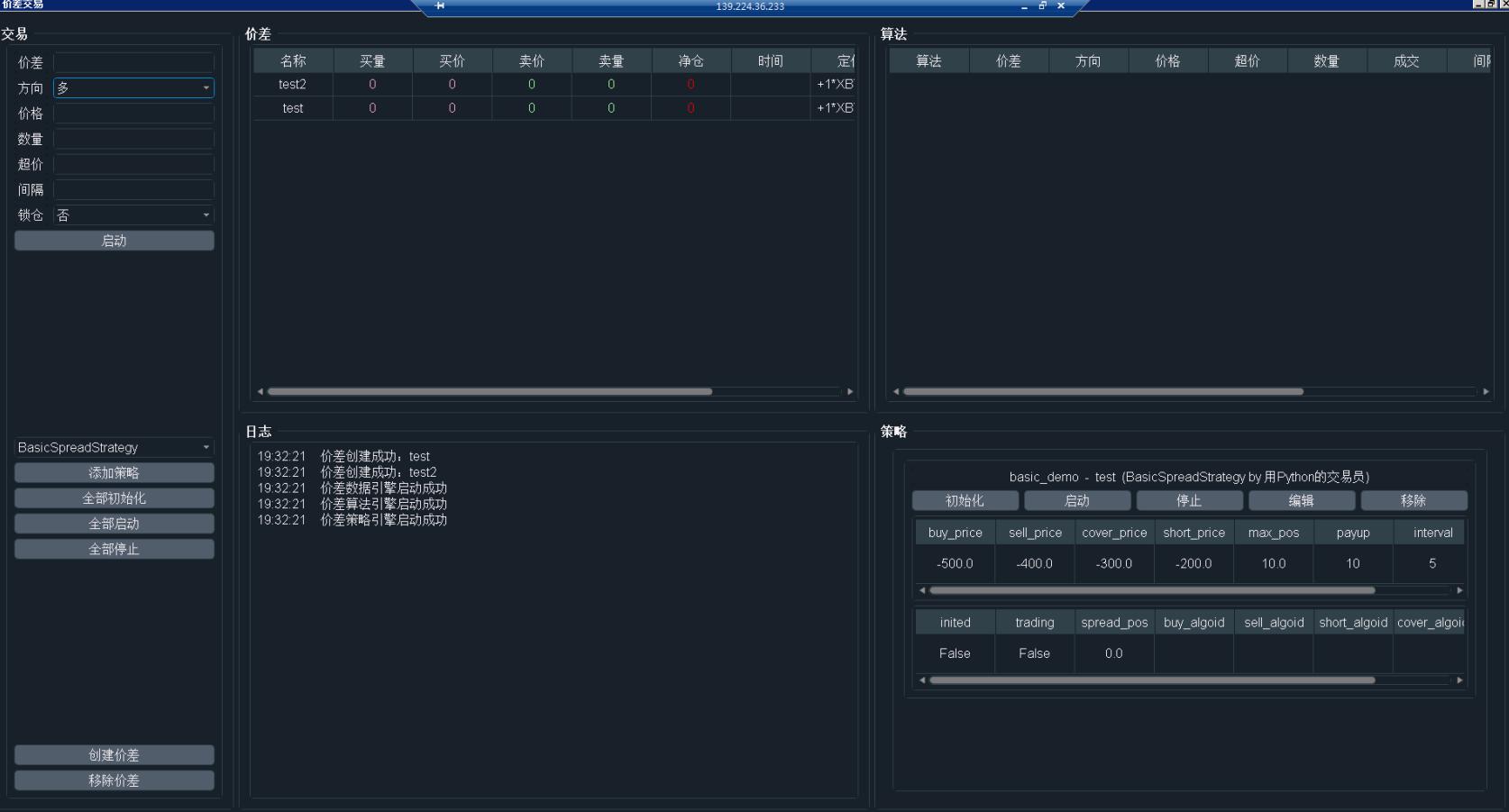
交易乘数: 30万

价差组合

300价差

持仓: 3

SpreadTradingApp



The image shows a screenshot of the SpreadTradingApp interface, which consists of two main windows:

- 价差交易 (Spread Trading) Window:** This window contains several input fields and a table:
 - 交易 (Transaction):** Fields for Price Difference, Direction (多), Price, Quantity, Margin, Stop Price, and Lock.
 - 价差 (Spread):** A table with columns: 名称 (Name), 买量 (Buy Volume), 买价 (Buy Price), 卖价 (Sell Price), 卖量 (Sell Volume), 净仓 (Net Position), Time, and 定时 (Timer). It lists two entries: "test2" and "test".
- 策略 (Strategy) Window:** This window contains a table and a control bar:
 - 策略 (Strategy):** A table with columns: 算法 (Algorithm), 价差 (Spread), 方向 (Direction), 价格 (Price), 超价 (Margin), 数量 (Quantity), 成交 (Trade), and 间隔 (Interval). It lists two entries: "basic_demo - test" and "BasicSpreadStrategy by 用Python的交易员".
 - 控制栏 (Control Bar):** Buttons for 初始化 (Initialize), 启动 (Start), 停止 (Stop), 编辑 (Edit), and 移除 (Remove).
 - 日志 (Log) Panel:** Displays log messages from 19:32:21, including the creation of spreads and strategies, and the start of the spread algorithm.

三层结构

价差

- 组合的价格以及持仓
- 每条腿可以属于多个价差

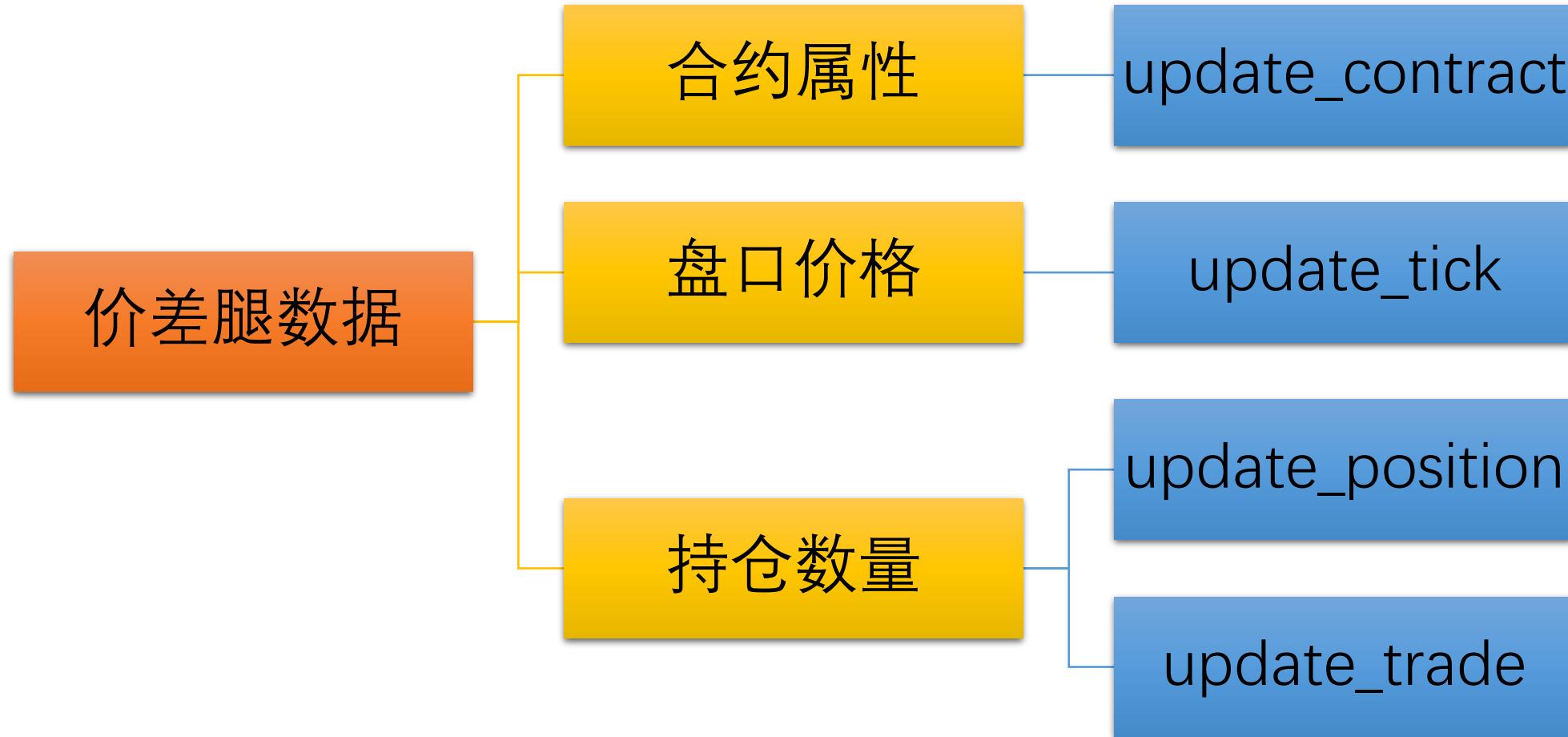
算法

- 负责价差交易的执行
- 一条主动腿，多条被动腿

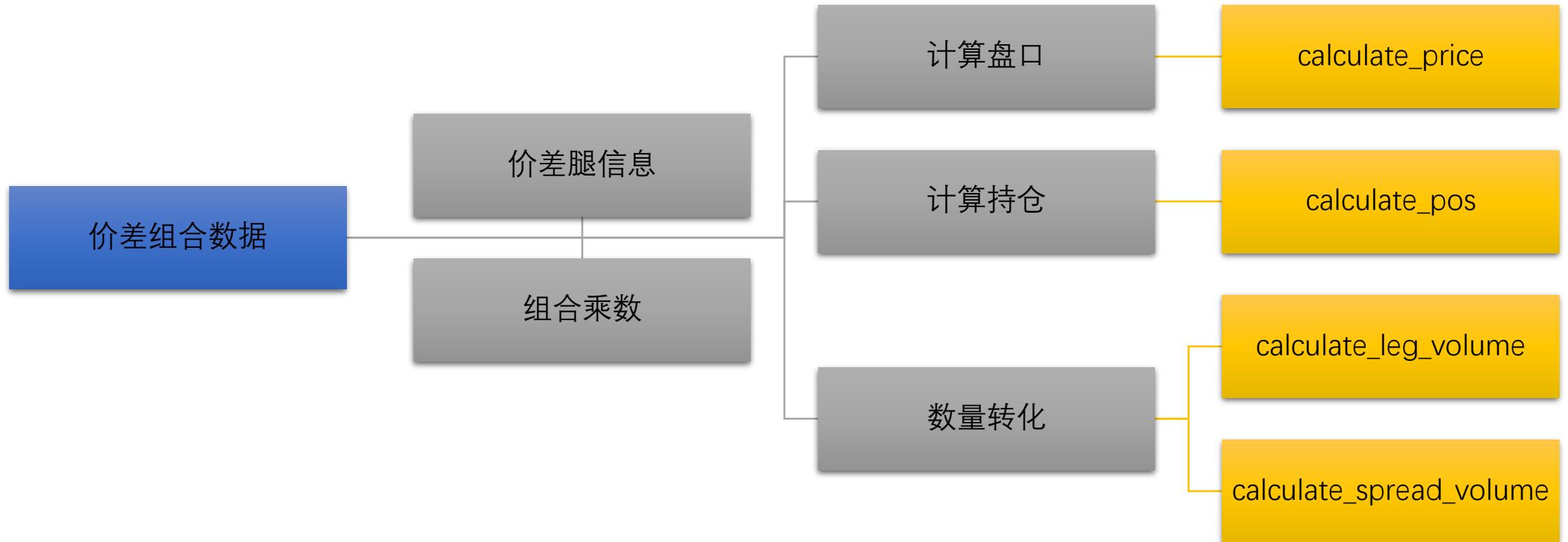
策略

- 负责价差算法的调度
- 包含Alpha捕捉的交易逻辑

LegData



SpreadData



开发模板

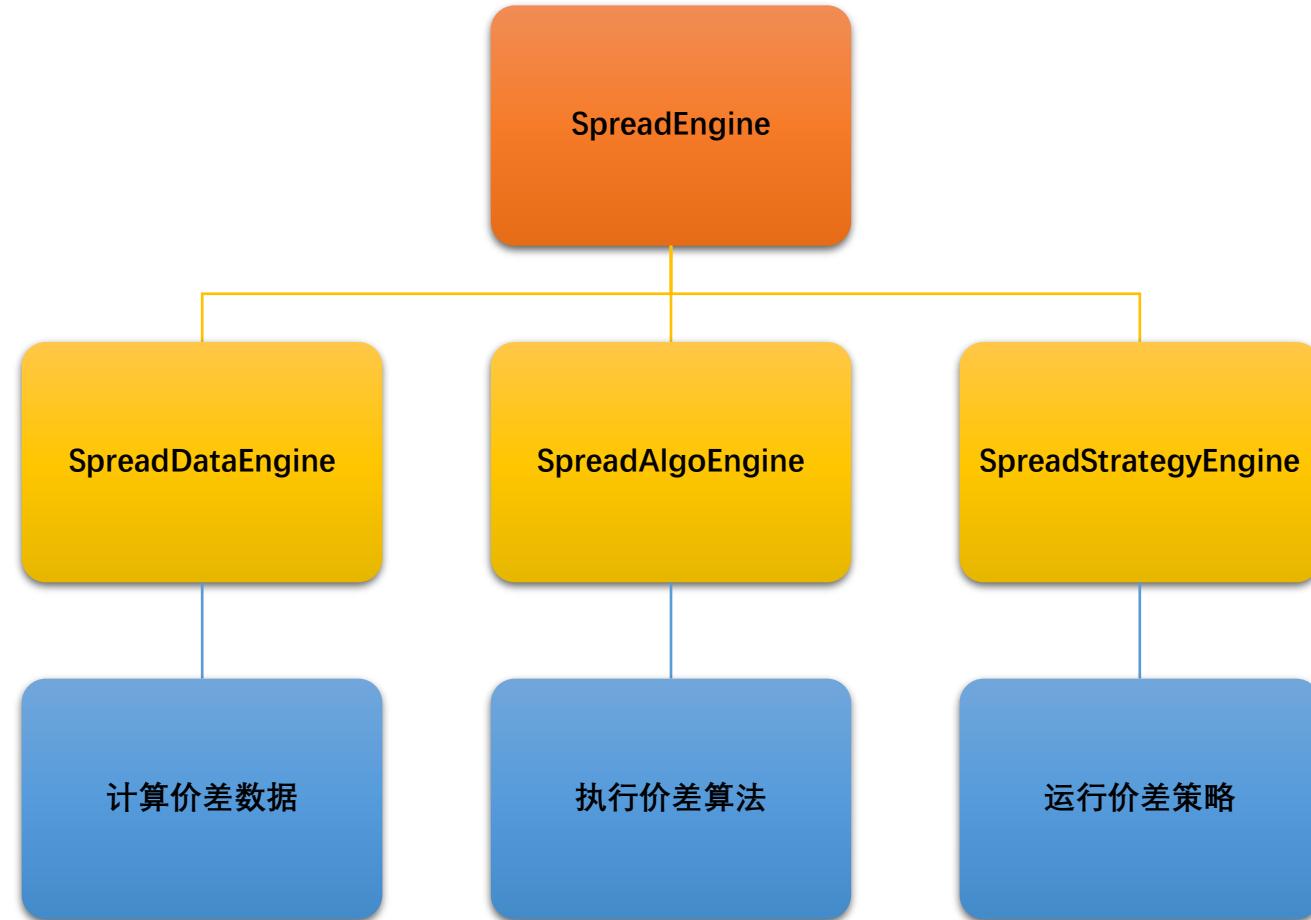
算法

- SpreadAlgoTemplate

策略

- SpreadStrategyTemplate

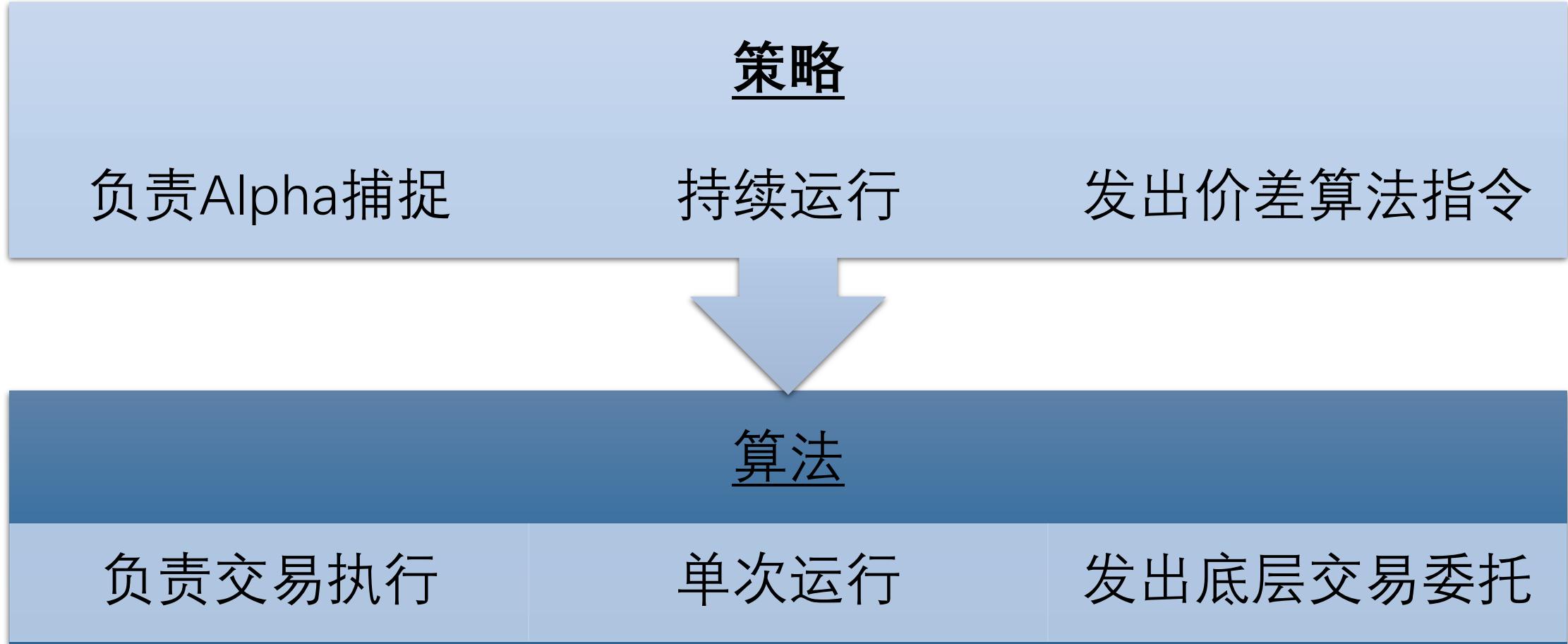
四大引擎



什么是价差算法

- 将价差交易简化为普通委托，封装掉所有主动腿下单和被动腿对冲的细节。

策略 vs 算法



算法模板

名称

- SpreadAlgoTemplate

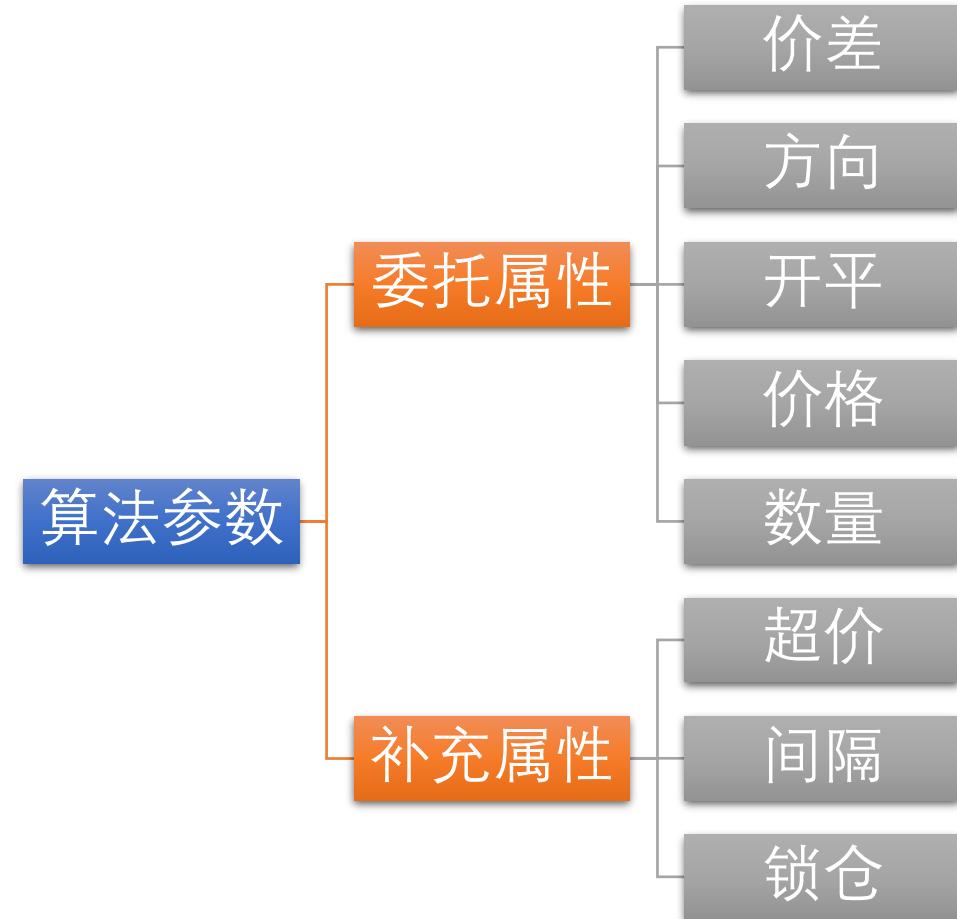
功能

- 提供价差算法的标准开发功能接口

位置

- vnpy.app.spread_trading.template

启动算法的参数



状态检查函数

is_active

- 当前算法是否结束

check_order_finished

- 检查委托是否全部结束

check_hedge_finished

- 检查被动腿和主动腿是否匹配

check_algo_finished

- 检查算法数量已经全部成交

calculate_traded

- 计算已成交价差数量

引擎管理函数

stop

- 停止算法

update_tick

- 更新行情数据

update_trade

- 更新成交数据

update_order

- 更新委托数据

update_timer

- 更新定时器事件

回调函数 (virtual)

on_tick

- 收到行情推送

on_trade

- 收到成交推送

on_order

- 收到委托推送

on_interval

- 收到定时间隔更新

主动函数

委托相关

- send_long_order
- send_short_order

撤单相关

- cancel_leg_order
- cancel_all_order

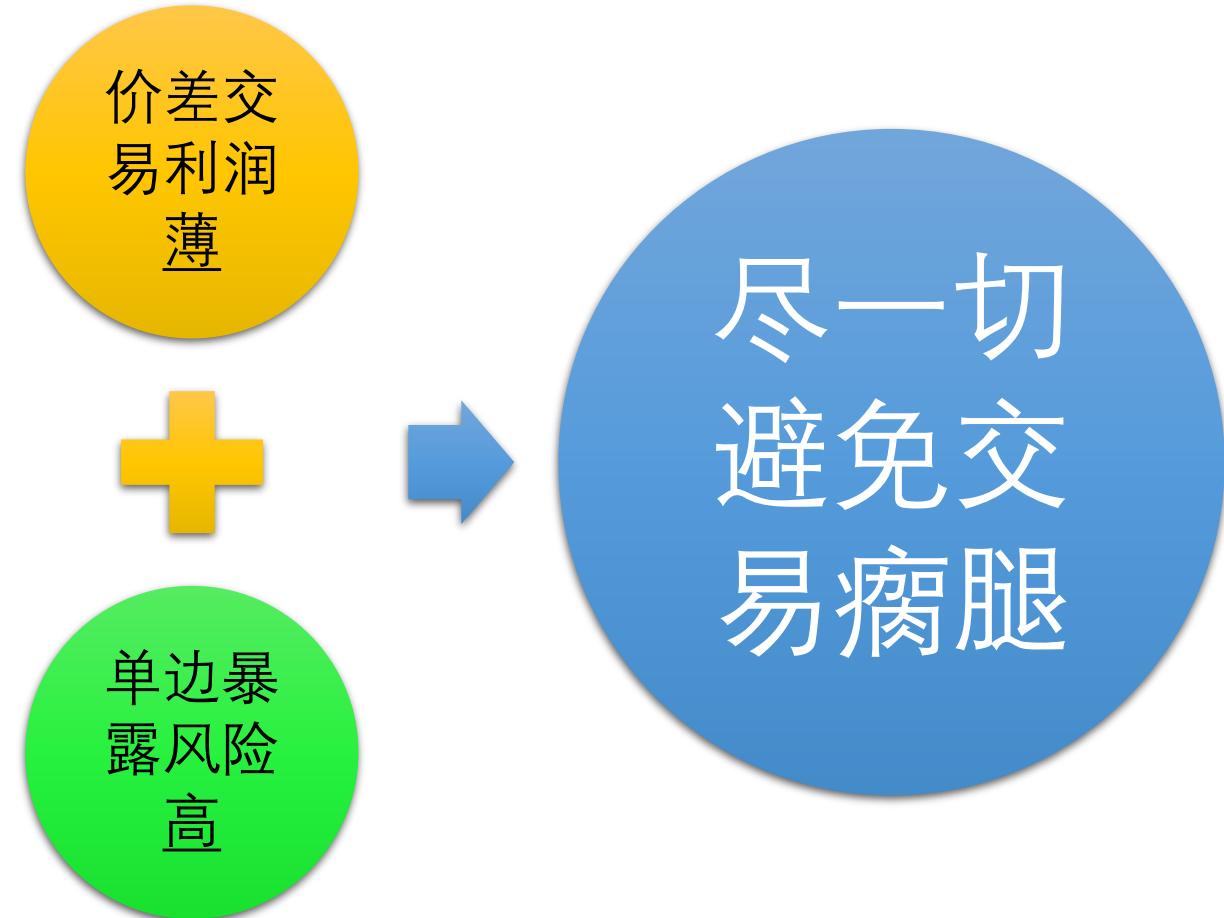
查询相关

- get_tick
- get_contract

周边功能

- put_event
- write_log

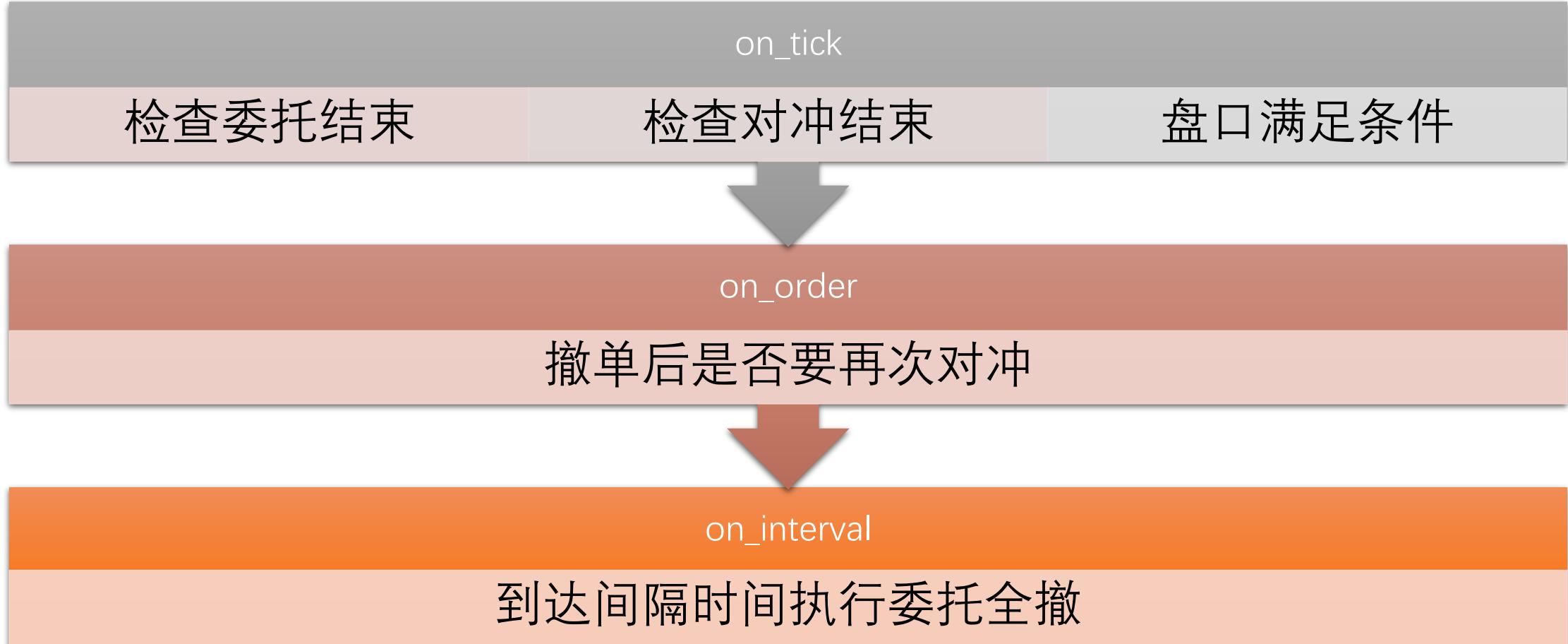
最小风险原则



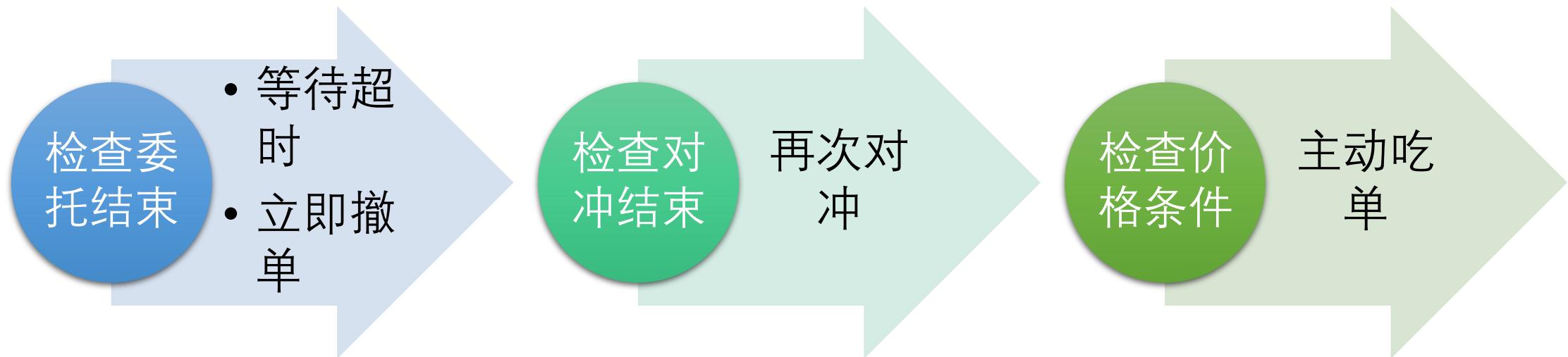
SpreadTakerAlgo



接收数据



算法状态机



主动函数

send_leg_order

- 基于仓位变化量下单（正负号判断方向）

take_active_leg

- 吃掉主动腿的盘口挂单
- 委托数量的限制：盘口挂单量、剩余数量的最小值

hedge_passive_legs

- 根据主动腿的成交量，计算被动腿需要成交的数量
- 遍历对冲各条被动腿（盘口吃单）

超时限制

主动腿吃单无法成交

被动腿对冲无法成交

**超时
全撤**

数量的转换



单腿 -> 价差



价差 -> 单腿



单腿到期差

- 根据主动腿成交量，计算对应的价差数量

价差到单腿

- 根据上一步的价差数量，计算需要对冲的被动腿数量
- 根据剩余价差委托数量，计算主动腿要吃单的数量

Taker的不足

主动腿见价下单

- 所有腿均需要付出盘口买卖价差的滑点成本
- 等待主动腿对价满足条件，需要更长的时间

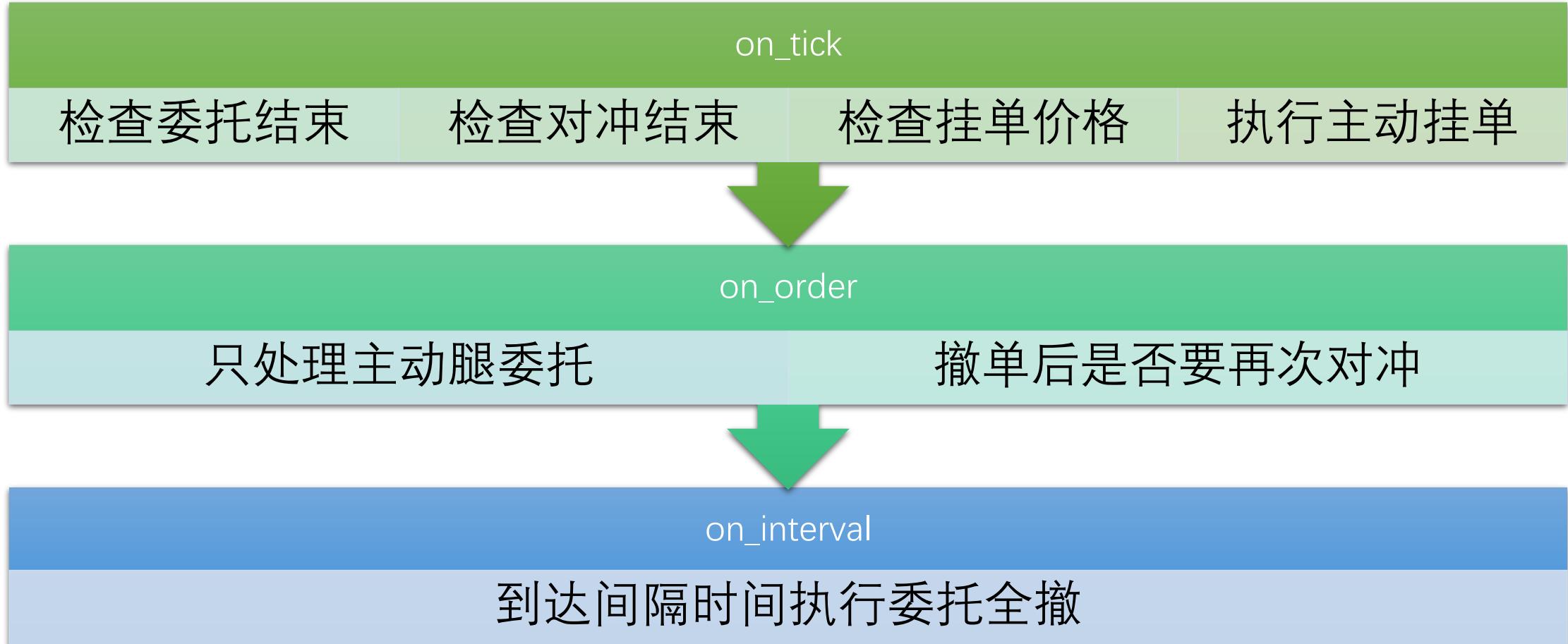
SpreadMakerAlgo

主动挂单

基于被动腿盘口，计算主动腿最差成交价

主动挂单做市，目标是能赚得盘口价差

原理相似



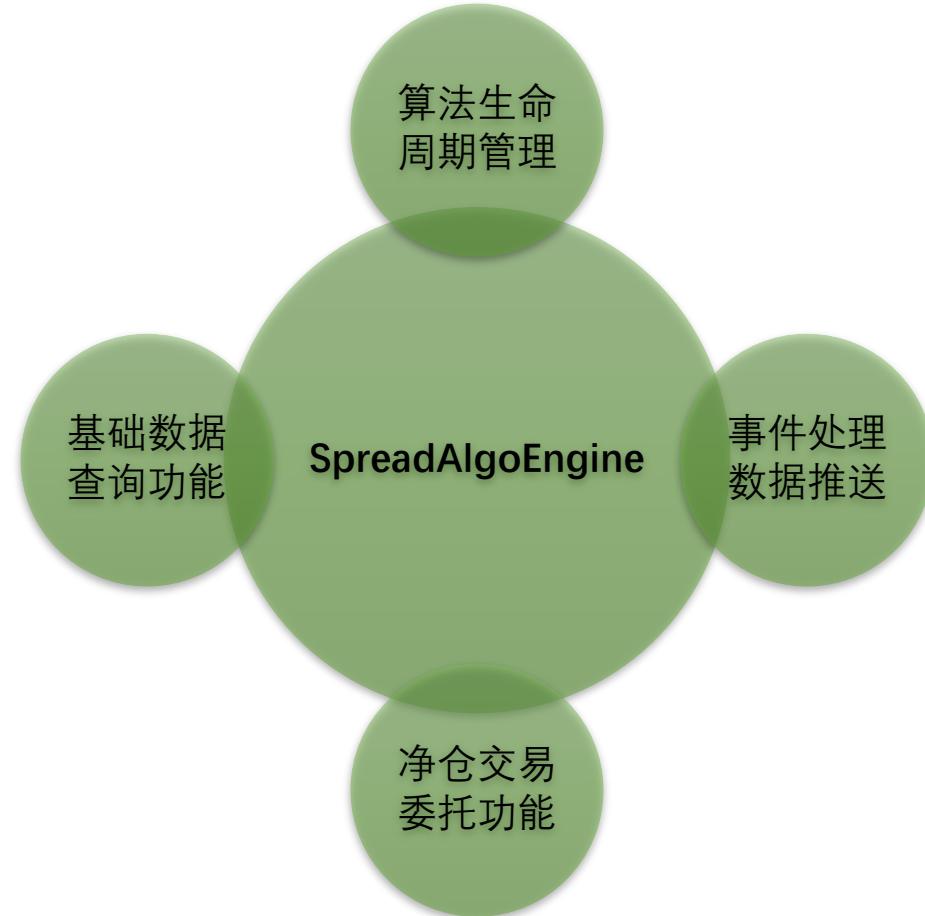
挂撤单管理



容错空间



算法引擎控制



策略模板

名称

- SpreadStrategyTemplate

功能

- 提供价差策略的标准开发功能接口

位置

- vnpy.app.spread_trading.template

回调函数

策略状态

on_init

on_start

on_stop

价差相关

on_spread_data

on_spread_pos

on_spread_algo

委托相关

on_order

on_trade

算法函数

start_long_algo/start_short_algo

- 启动一个新的价差交易算法
- 参数：代码vt_symbol、价格price、数量volume、超价payup、间隔interval、锁仓lock
- 返回：算法编号algoid

stop_algo

- 停止一个运行中的算法（注意有可能会瘸腿）
- 参数：算法编号algoid

stop_all_algos

- 停止所有活动状态的算法

委托函数

buy/sell/short/cover

- 发出底层交易委托
- 参数：代码vt_symbol、价格price、数量volume、锁仓lock
- 返回：委托号vt_orderid

cancel_order

- 撤销某个特定的委托
- 参数：委托号vt_orderid

cancel_all_orders

- 撤销全部活动状态的委托

控制函数

get_spread_tick 获取价差数据转化的TickData

可用于BarGenerator合成K线

get_leg_tick 获取特定合约的TickData

get_leg_pos 获取某条腿的持仓情况，用于处理瘸腿后的细粒度调整

参数：合约代码vt_symbol，方向direction

get_spread_pos 获取价差的净持仓

put_event 界面更新

write_log 输出日志信息

寻找价差配对

基于经济学中的客
观关系

- 价差本身可能存在趋势

基于统计学上的分
析结果

- 可能哉入伪回归的坑中

协整关系 (Cointegration)



平稳序列



常见价差类型



比特币跨交易所

	腿1	腿2
代码	火币 BTC/USDT	币安 BTC/USDT
合约大小	1	1
价格乘数	1	-1
交易乘数	1	-1
委托顺序	主动腿	被动腿
合约模式	正向	正向

比特币期现

	腿1	腿2
代码	OKEX BTC/USDT	OKEX BTCUSD-20200925
合约大小	1	100
价格乘数	1	-1
交易乘数	1	-1
委托顺序	被动腿	主动腿
合约模式	正向	反向

比特币永续费率

	腿1	腿2
代码	火币 BTC/USDT	BITMEX XBTUSD
合约大小	1	1
价格乘数	1	-1
交易乘数	1	-1
委托顺序	被动腿	主动腿
合约模式	正向	反向

股指期货跨期

	腿1	腿2
代码	IF2006	IF2007
合约大小	300	300
价格乘数	1	-1
交易乘数	1	-1
委托顺序	被动腿	主动腿
合约模式	正向	正向

300指数期现

	腿1	腿2
代码	510050	IF2006
合约大小	1	300
价格乘数	1000	-1
交易乘数	3000000	-1
委托顺序	主动腿	被动腿
合约模式	正向	正向

虚拟钢厂价差

	腿1	腿2	腿3
代码	rb2009	i2009	j2009
合约大小	10	100	100
价格乘数	2	-3	-1
交易乘数	20	-3	-1
委托顺序	主动腿	被动腿	被动腿
合约模式	正向	正向	正向

黄金矿商价差

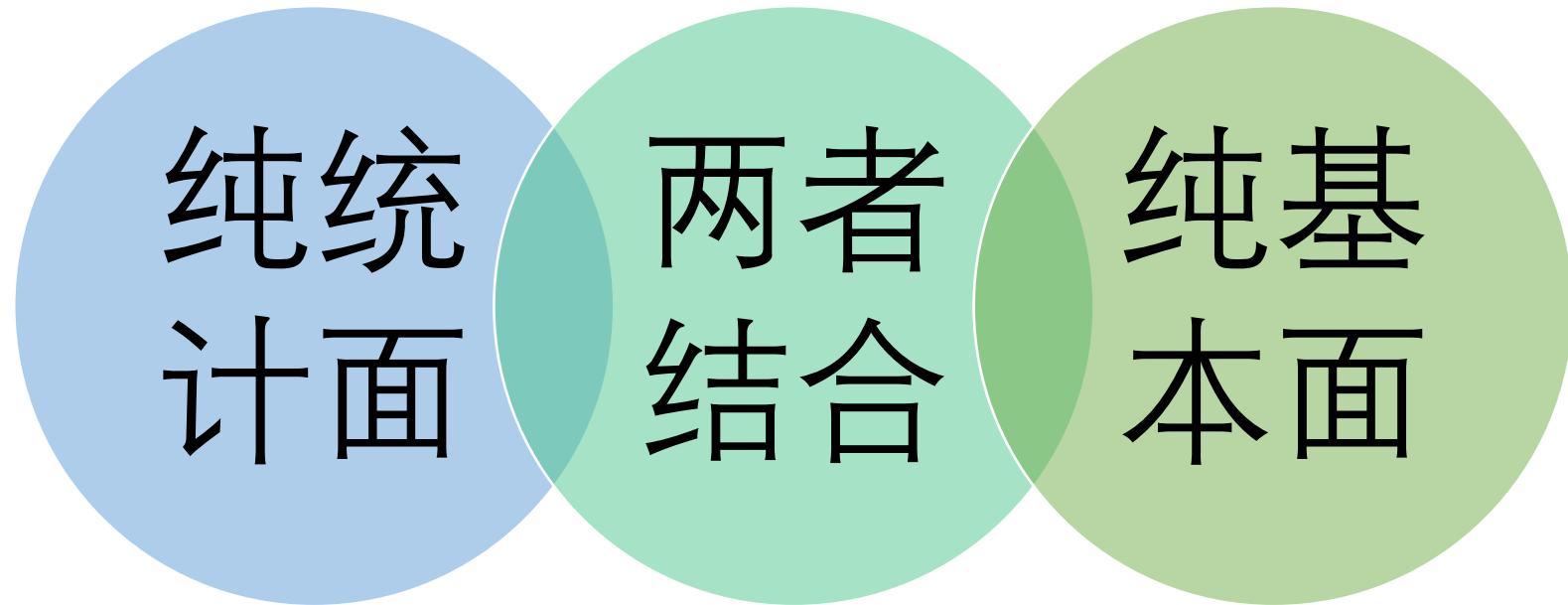
	腿1	腿2
代码	GLD	GDX
合约大小	1	1
价格乘数	1	-1
交易乘数	1	-1
委托顺序	被动腿	主动腿
合约模式	正向	正向



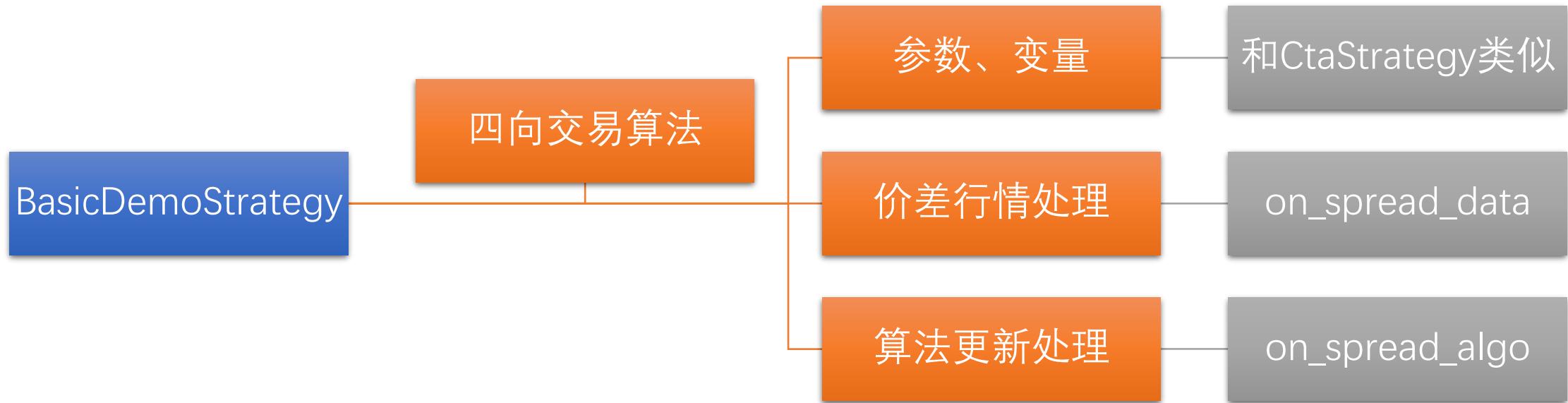
Pair Trading Lab

- 针对美股的配对交易研究平台
- 用户名: vnpy
- 密码: Vnpy@1234
- 到期日: 2020/7/16

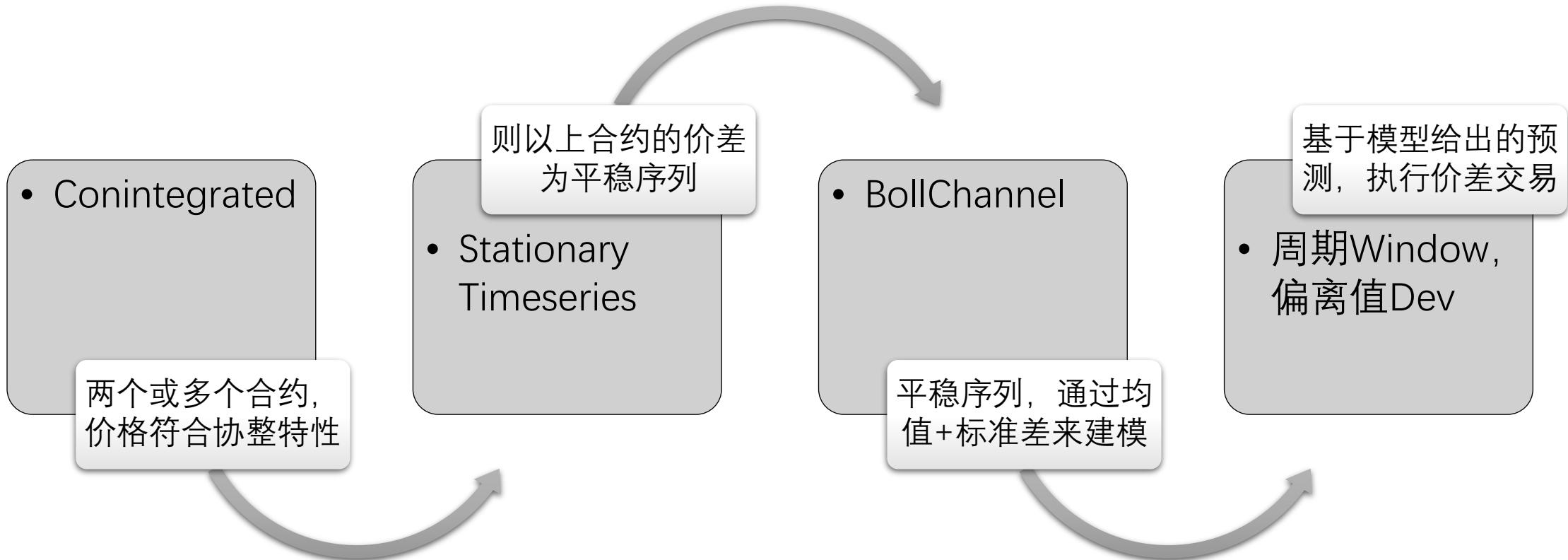
价差交易策略的分类



固定范围



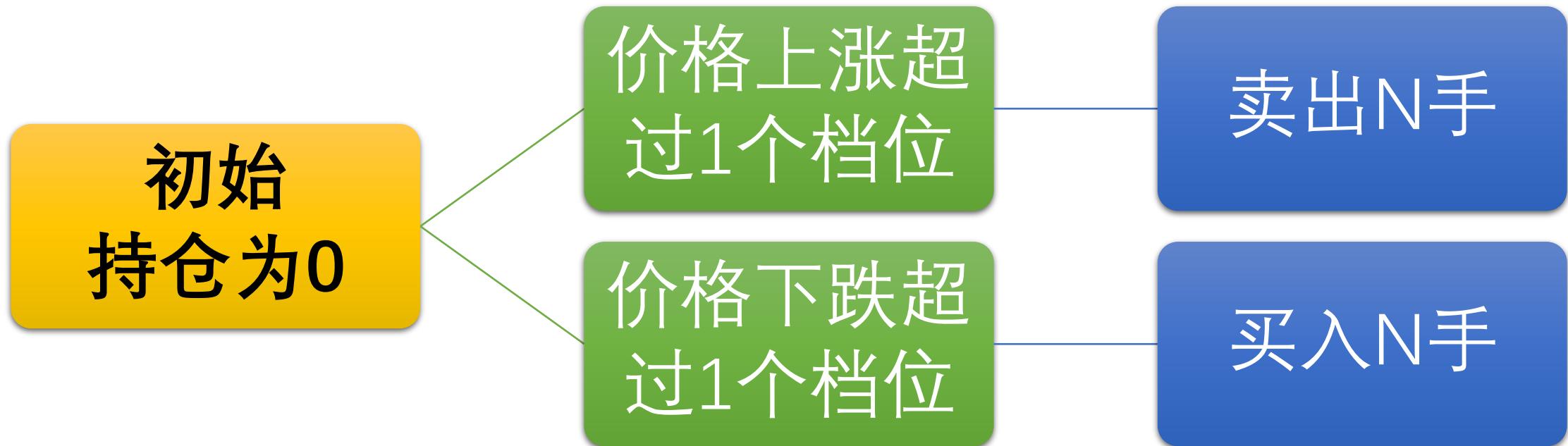
统计套利



什么是网格

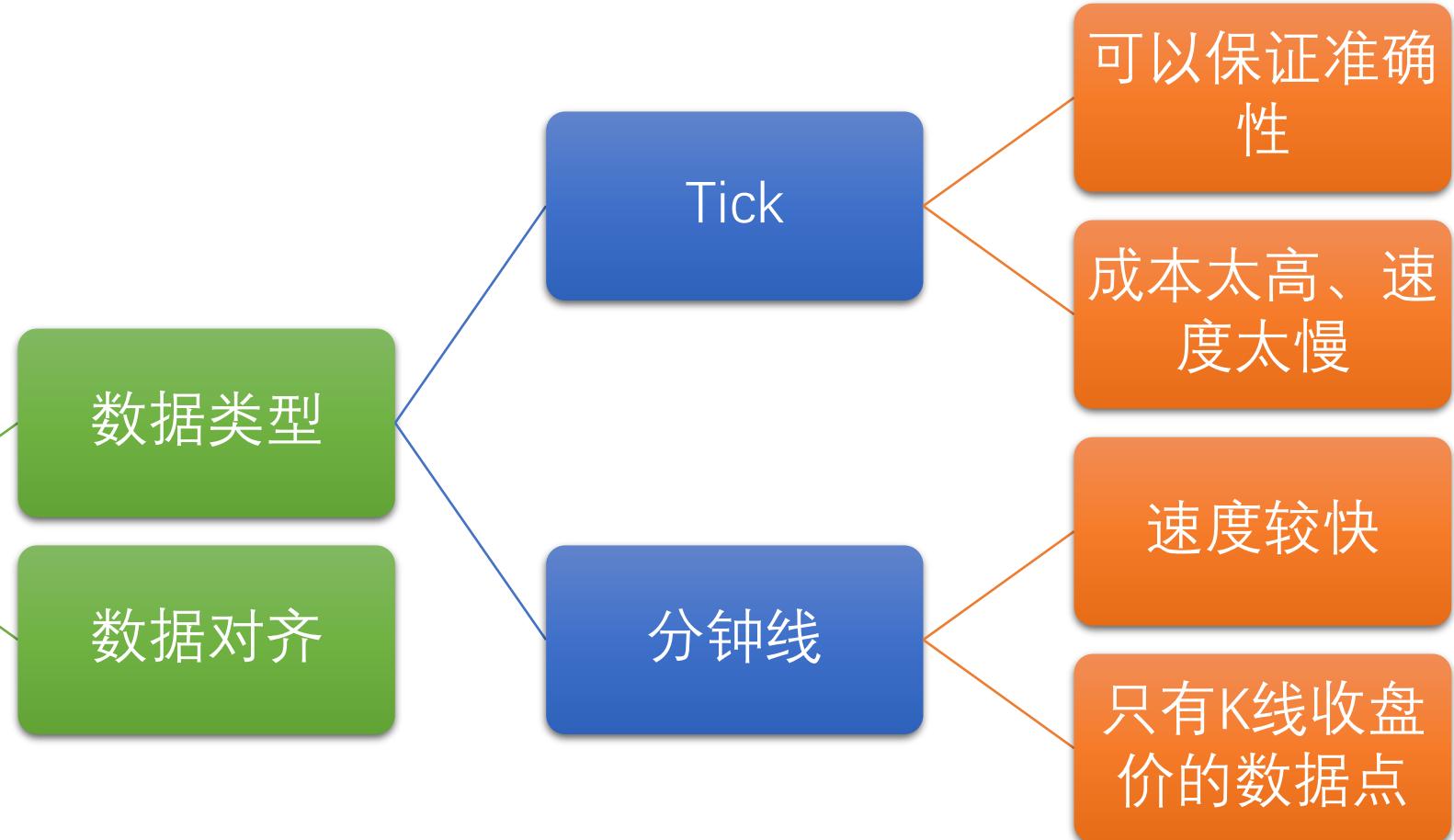


网格的交易逻辑

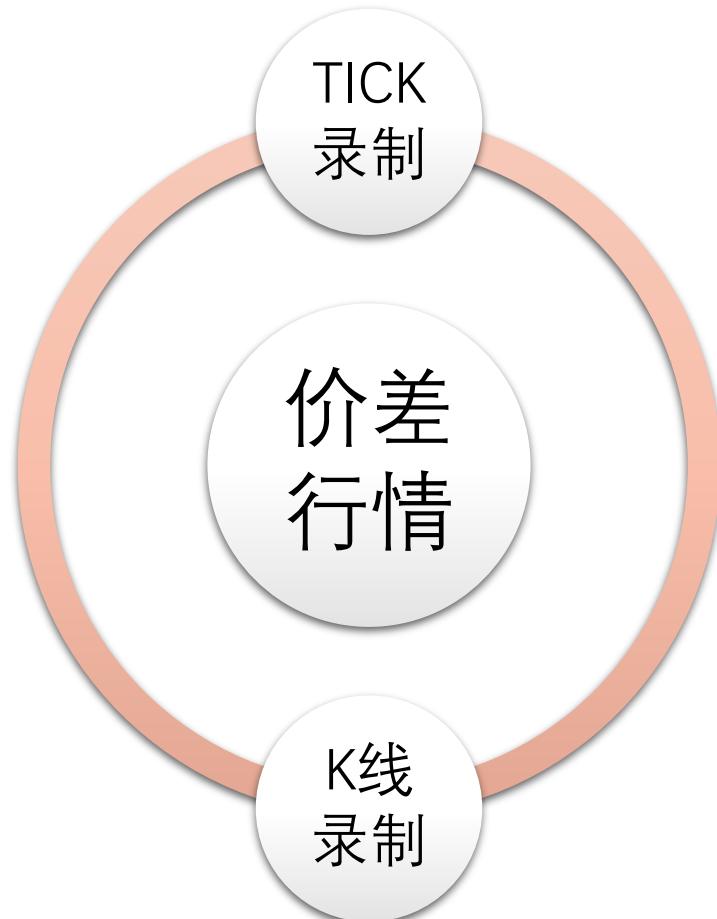


价差回测的难点

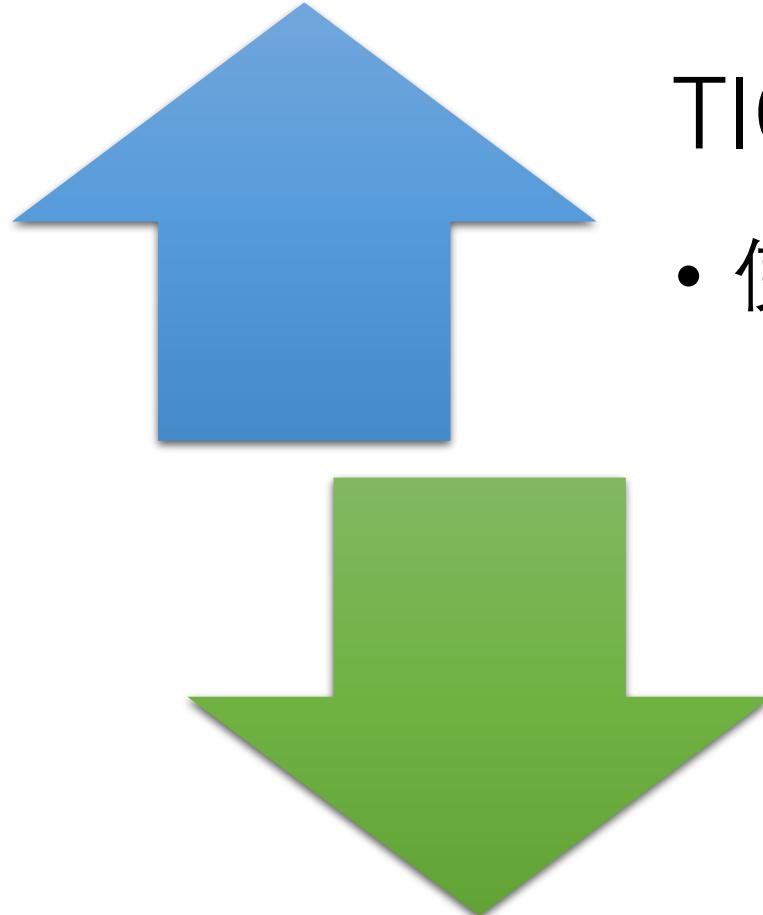
价差历史数据
的生成是难点



录制价差数据



两种模式



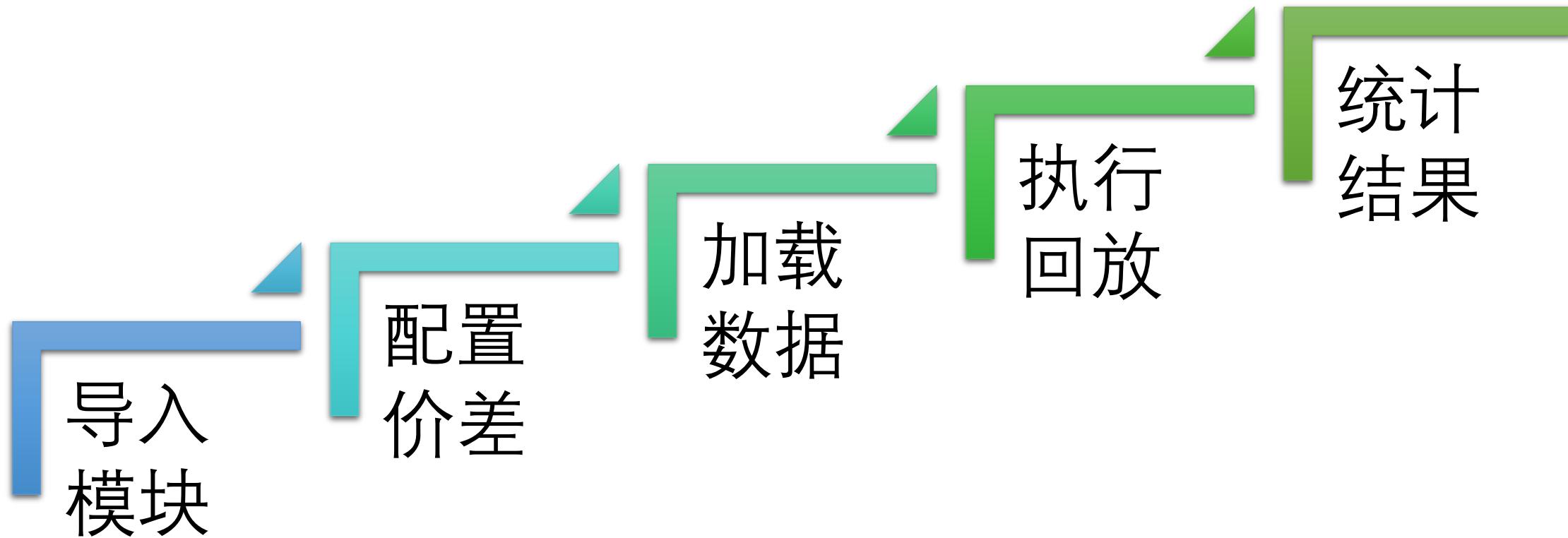
TICK模式

- 使用录制的价差TICK

K线模式

- 使用各腿K线
(1分钟)

Jupyter操作

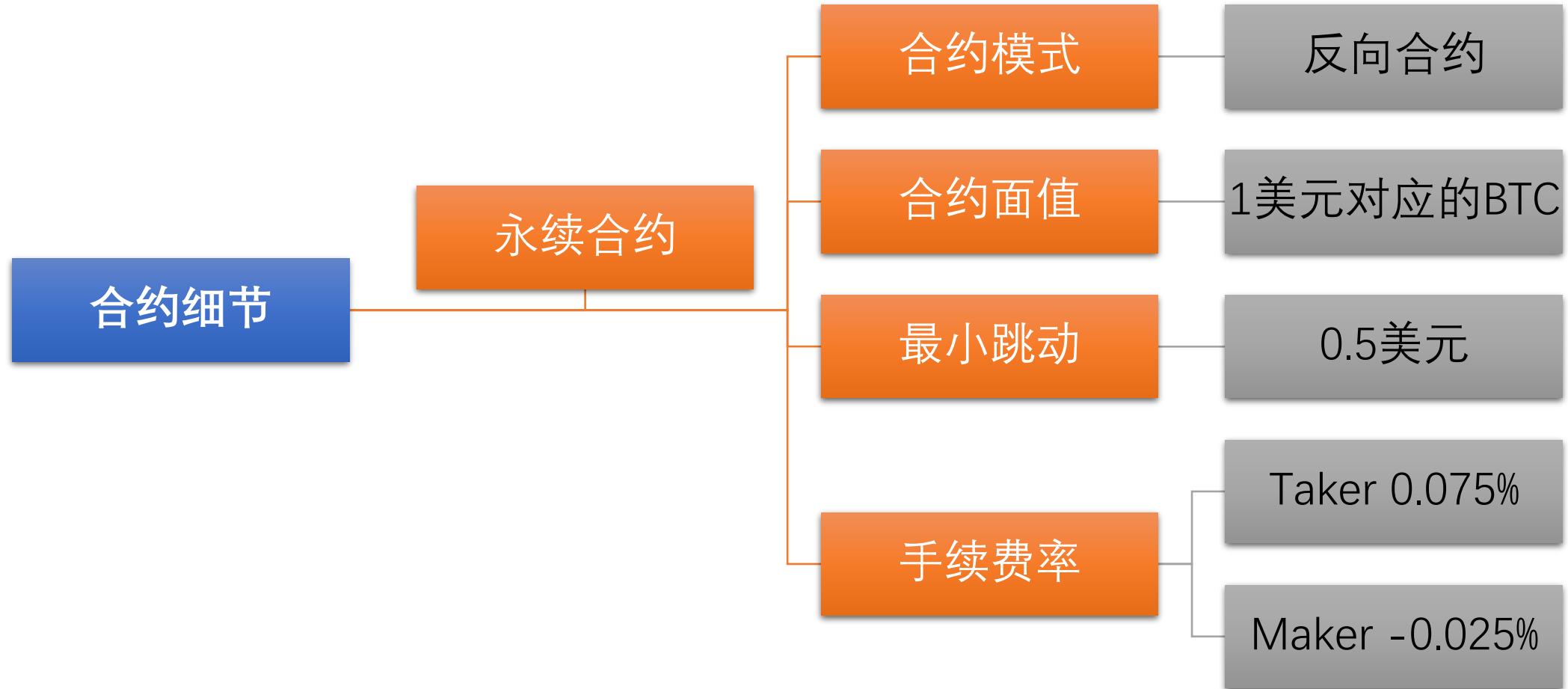




什么是反向合约？

以数字货币本身而非法币来结算盈亏的衍生品合约（期货、永续）

BITMEX的XBTUSD



正向 VS 反向

	CME BTC期货	BitMEX XBTUSD永续
合约类型	正向合约	反向合约
合约面值	5个BTC	1美金对应的BTC
计价货币	USD	USD
盈亏结算	USD	BTC
保证金	USD	BTC

币本位计价

法币本位 (CME)

- BTC价值 = 5 BTC * 持有合约数量
- USD价值 = 5 BTC * 合约价格 * 持有合约数量

币本位 (BitMEX)

- BTC价值 = 1 / 合约价格 * 1 USD * 持有合约数量
- USD价值 = 1 USD * 持有合约数量

以币的数量为准

正向合约腿



无需做任何处理

反向合约腿



合约数量基于市价
转化为币数量

从合约数量到币数量

calculate_inverse_volume

- 合约数量 * 合约大小 / 当前价格

需要做转化的地方

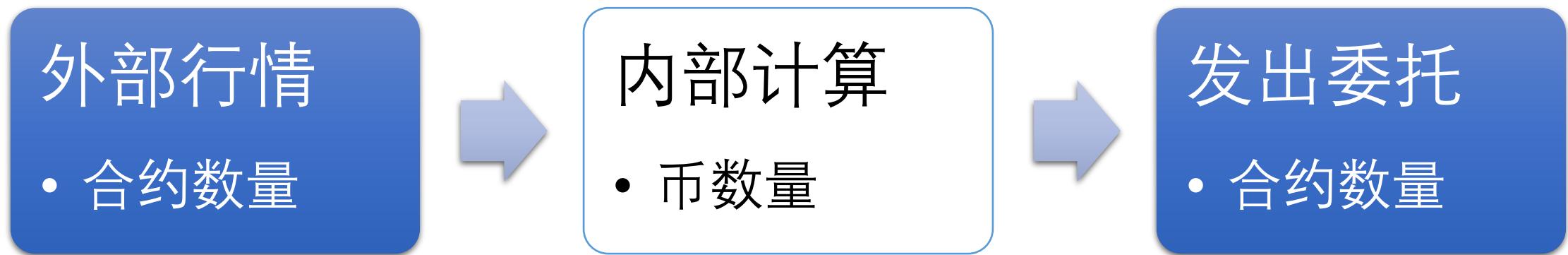
盘口挂单量

- 使用挂单价格

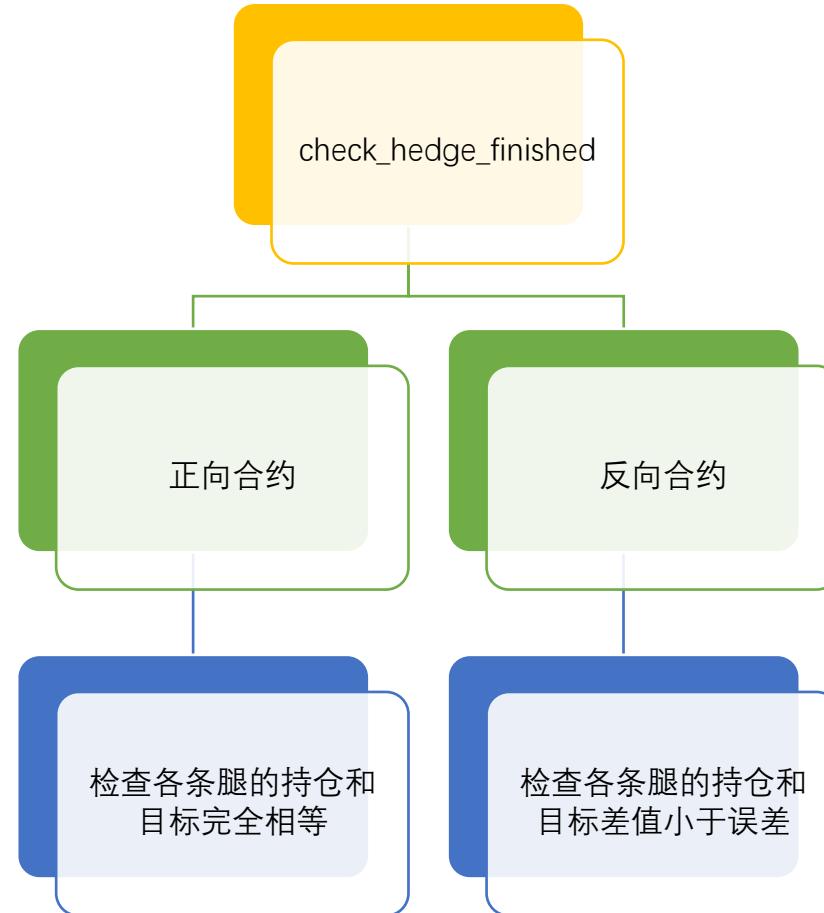
价差持仓量

- 使用最新成交价

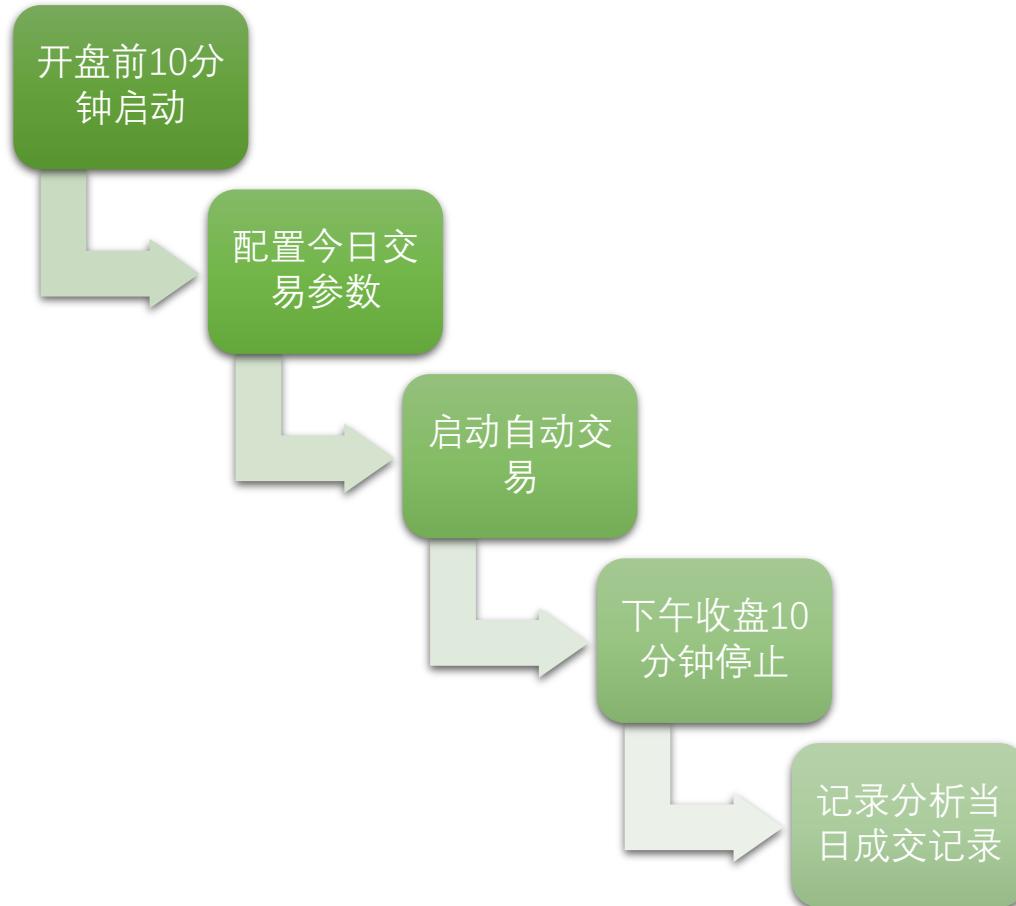
下单转化



控制函数的优化



策略每日生命周期



策略部署运行

在strategies目录创建策略文件，并完成策略代码开发

启动VN Trader加载SpreadTrading模块

添加一个新的策略实例，并配置参数

初始化策略（后续将会支持历史数据载入）

启动策略，开始交易（别忘记要先创建策略交易的价差）

盘中异常分类



交易所挂了

经纪商柜台挂了

电脑硬件异常

网络连接抖动

交易平台异常

策略程序错误

推荐处置流程



预防永远在第一位

充分测试

- 代码检查：保证代码质量
- 回测检验：程序代码正确
- 仿真交易：交易逻辑正确

托管环境

- 托管服务器：专业金融机房运维
- 云服务器：云服务商机房运维

Let it crash

- 针对风险不可控类环境：数字货币
- 交易进程写文件或端口定时更新状态
- 守护进程检查状态，一旦异常立即重启交易进程
- 守护进程保证状态恢复和及时邮件通知



vn.py社区公众号：[vnpy-community](#)

各种入门教程、进阶经验、公告发布等等