

法律声明

本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。



关注 **小象学院**

《进阶！量化交易实战：迭代式的量化策略研发》第3期

第7课：期货趋势型策略开发
主讲：宋战江

内容介绍

趋势型量化策略必备的期货知识

期货量化工具介绍

趋势型策略原理

用TB开发商品期货趋势型策略

多空双向 T+0 杠杆交易

趋势型量化策略必备的期货知识

典型的交易标的

- ☐ 股票
- ☐ 期货
- ☐ 债券
- ☐ 外汇
- ☐ 期权
- ☐ 数字货币
- ☐



期货交易的基本特征

- ☐ 合约标准化
- ☐ 交易集中化
- ☐ 双向交易和对冲机制
- ☐ 杠杆机制
- ☐ 每日无负债结算制度
- ☐ T+0交易

避免断更,请加微信501863613

交易所名称	交易品种
上海期货交易所	黄金、白银、铜、铝、锌、橡胶、 燃油、螺纹钢、线材、石油、沥青、铅
郑州商品交易所	小麦、棉花、白糖、PTA、绿豆、红小 豆、花生、菜油籽、玻璃
大连商品交易所	大豆、豆粕、玉米、豆油、啤酒、焦 炭、棕榈油、聚氯乙烯
中国金融期货交易所	沪深300指数、5年期国债期货

螺纹钢 期货合约

交易品种	螺纹钢
交易单位	10吨/手
报价单位	元（人民币）/吨
最小变动价位	1元/吨
涨跌停板幅度	不超过上一交易日结算价 $\pm 7\%$
合约交割月份	1、2、3、4、5、6、7、8、9、10、11、12月
交易时间	每周一至周五 上午：9:00—10:15 10:30—11:30 下午：13:30—15:00 连续交易时间：21:00至23:00
最后交易日	合约交割月份的15日（遇法定假日顺延）
最后交割日	最后交易日后第五个工作日
交割品级	标准品：符合国标GB1499.2-2007《钢筋混凝土用钢 第2部分：热轧带肋钢筋》HRB400或HRBF400牌号的 $\phi 16\text{mm}$ 、 $\phi 18\text{mm}$ 、 $\phi 20\text{mm}$ 、 $\phi 22\text{mm}$ 、 $\phi 25\text{mm}$ 螺纹钢。 替代品：符合国标GB1499.2-2007《钢筋混凝土用钢 第2部分：热轧带肋钢筋》HRB335或HRBF335牌号的 $\phi 16\text{mm}$ 、 $\phi 18\text{mm}$ 、 $\phi 20\text{mm}$ 、 $\phi 22\text{mm}$ 、 $\phi 25\text{mm}$ 螺纹钢。

交割地点	交易所指定交割仓库
最低交易保证金	合约价值的9%
交割方式	实物交割
交易代码	RB
上市交易所	上海期货交易所
最小交割单位	300吨
上市日期	2009年03月27日

期货市场的构成



期货投资分析方法综述

技术分析

- ★ 市场行为涵盖一切信息
- ★ 价格沿趋势移动
- ★ 历史会重演

基本分析

- ★ 因果关系明确，结论确定
- ★ 完备的即时资料
- ★ 极高的行业经验专业素质

程序化交易

- ★ 技术分析程序化
- ★ 数量统计分析、交易
- ★ 高频交易

价·量·时·空·仓

期货投资者的通病

- ☐ 满仓操作
- ☐ 逆势开仓
- ☐ 持仓综合症
- ☐ 判顶断底
- ☐ 死不认输
- ☐ 逆势抢反弹
- ☐ 频繁全天候操作
- ☐ 把握不好进出场时机
- ☐ 盲目跟风
- ☐ 主力盯单心态

工欲善其事 必先利其器

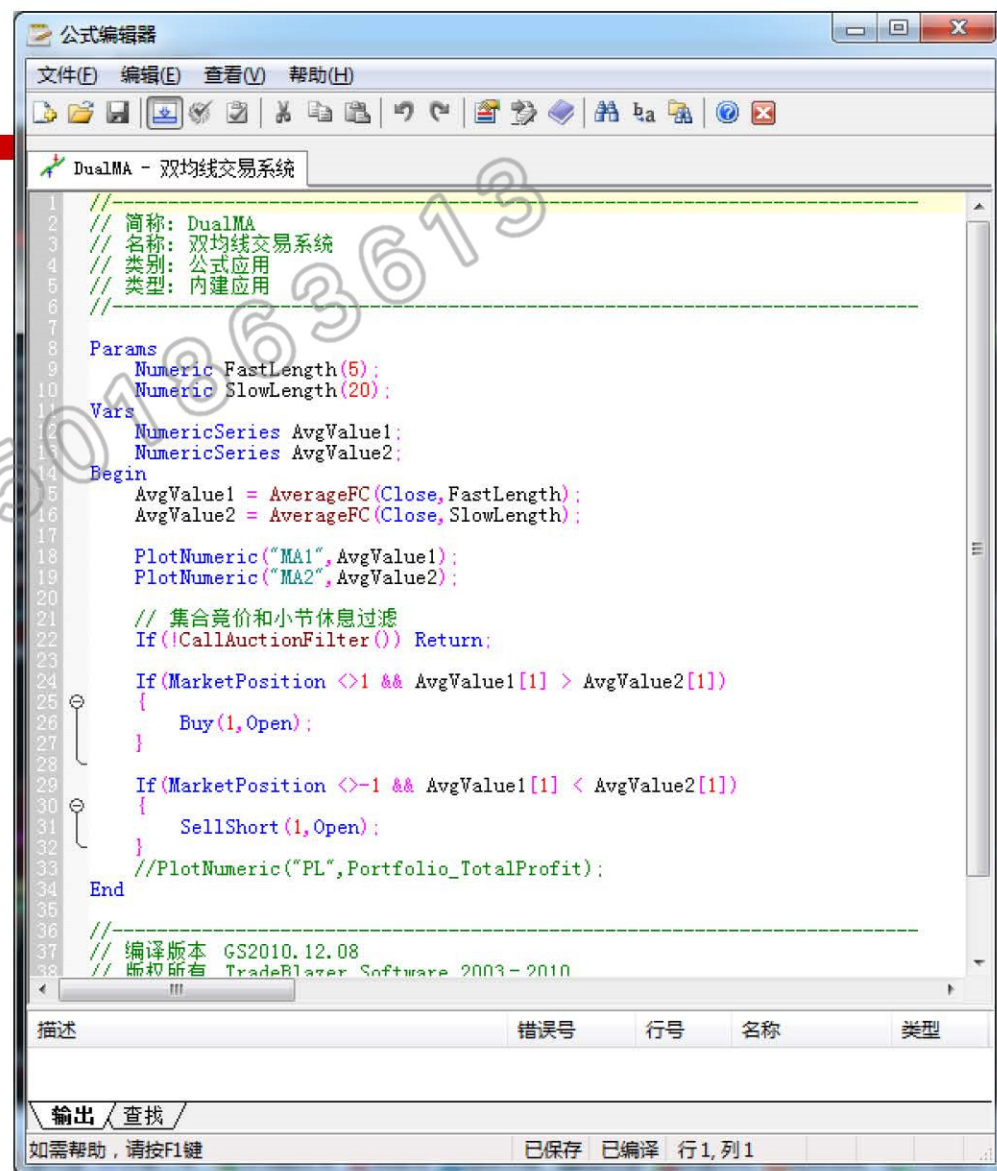
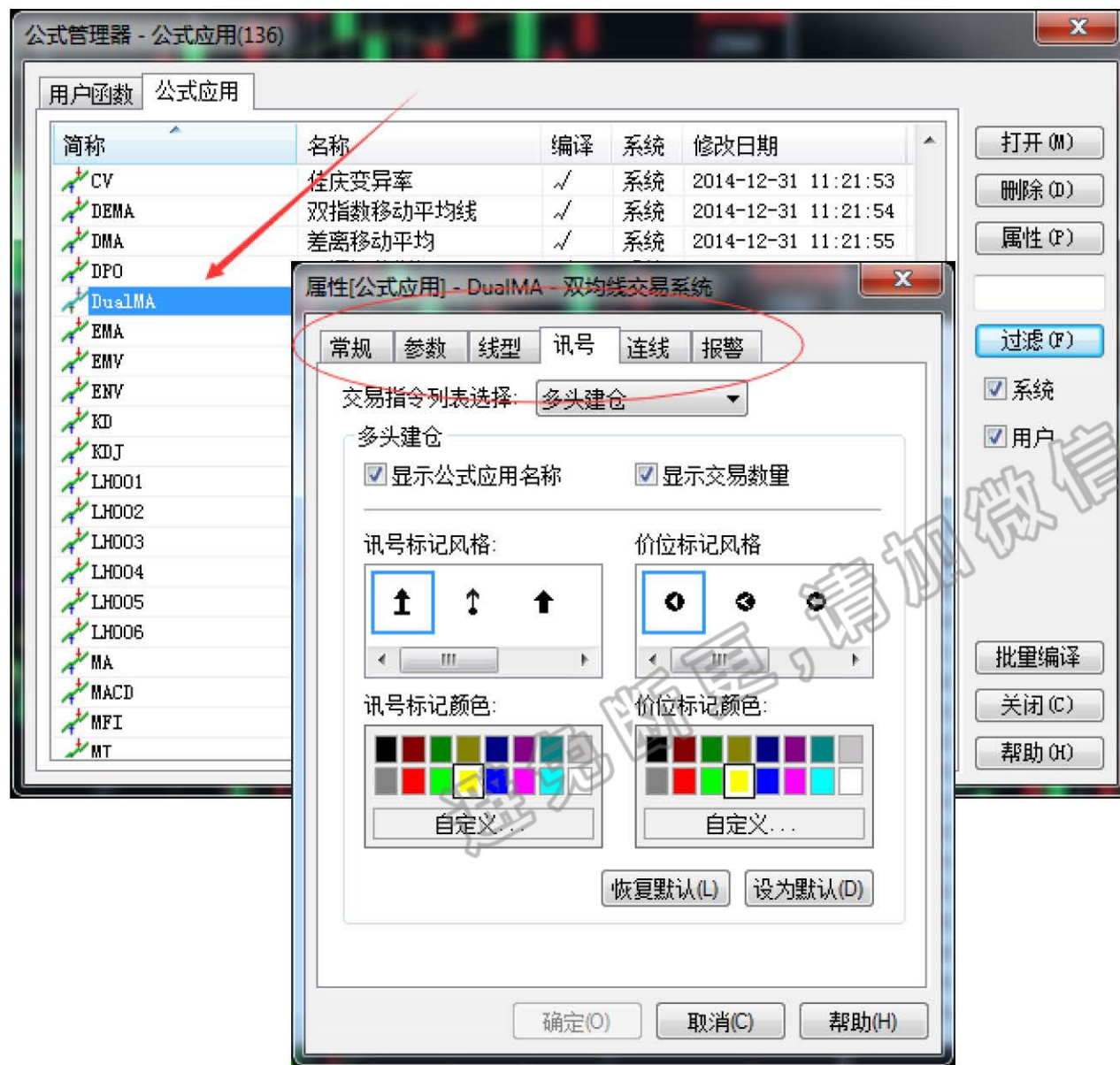
期货量化工具介绍

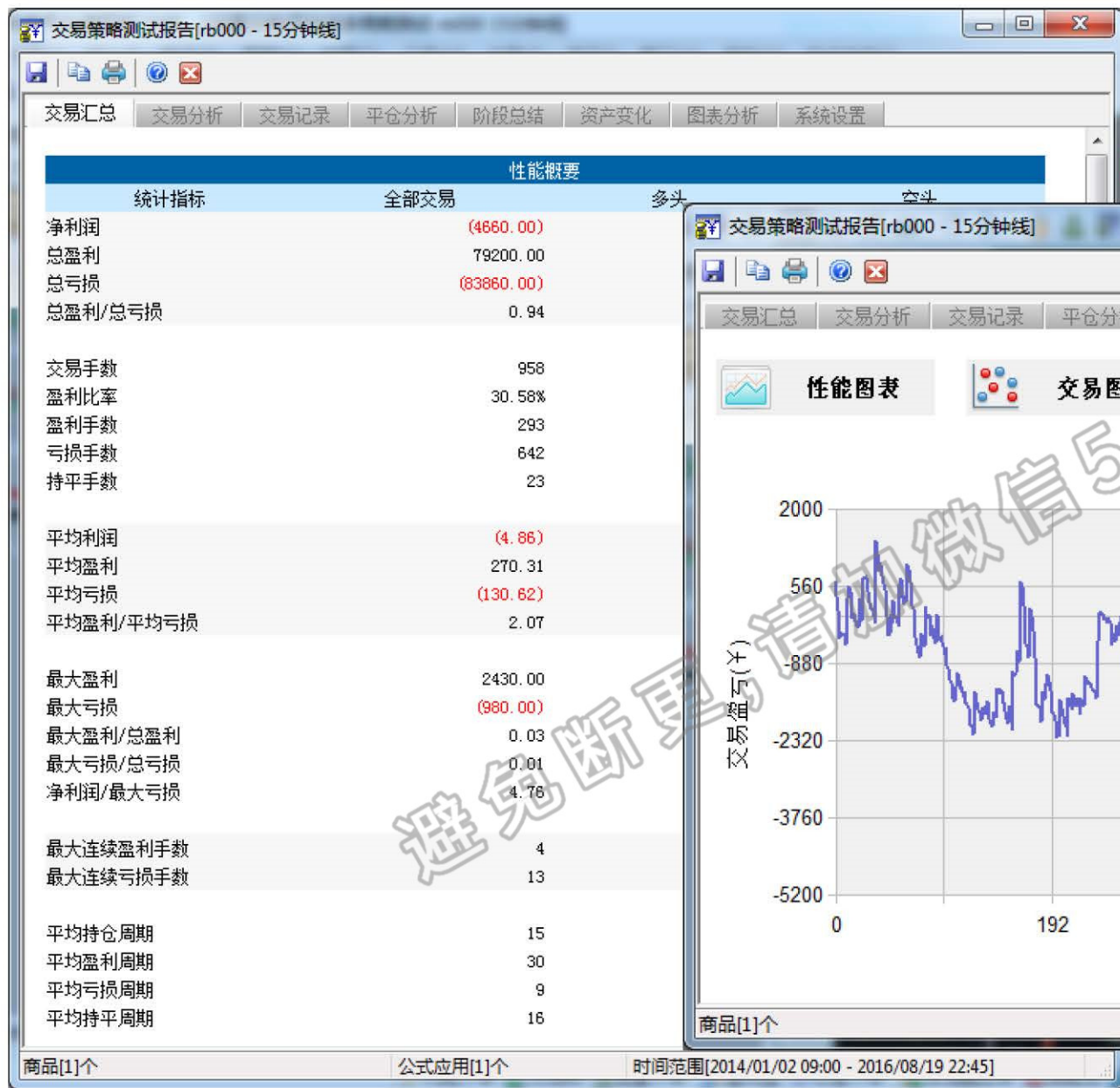
期货量化平台选择

- ☐ TB（交易开拓者）
- ☐ MC（Multicharts）
- ☐ 金字塔
- ☐ WH8（文华赢智）
- ☐ VN.PY（开源框架）
- ☐ 自行研发

避免断更，请加微信501863613







VN.PY

- ❑ 丰富的Python交易和数据API接口
- ❑ 事件驱动引擎 (vn.event)
- ❑ 开发示例 (vn.demo)
- ❑ 交易平台 (vn.trader)
- ❑ RPC模块 (vn.rpc)
- ❑ 官方网站 (www.vnpy.org) /知乎专栏
- ❑ 官方交流QQ群

```
eventEngine.py vtEngine.py
File Edit Source Refactor Project Debug Testing Git Tools Window Help
eventEngine.py vtEngine.py
MainEngine > __init__
170
171
172 def exit(self):
173     """退出程序前调用，保证正常退出"""
174     # 安全关闭所有接口
175     for gateway in self.gatewayDict.values():
176         gateway.close()
177
178     # 停止事件引擎
179     self.eventEngine.stop()
180
181     # 保存数据引擎里的合约数据到硬盘
182     self.dataEngine.saveContracts()
183
184
185 def writeLog(self, content):
186     """快速发出日志事件"""
187     log = VtLogData()
188     log.logContent = content
189     event = Event(type_=EVENT_LOG)
190     event.dict_['data'] = log
191     self.eventEngine.put(event)
192
193
194 def dbConnect(self):
195     """连接MongoDB数据库"""
196     if not self.dbClient:
197         try:
198             self.dbClient = MongoClient()
199             self.writeLog(u'MongoDB连接成功')
200         except ConnectionFailure:
201             self.writeLog(u'MongoDB连接失败')
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2
```


CTA策略

加载策略 全部初始化 全部启动 全部停止

double ema

初始化 启动 停止

name	className	author	vtSymbol	fastK
double ema	DoubleEm...	用Python...	IF1602	0.9

inited	trading	pos	fastMa0	fastMa1
False	False	0	0.0	0.0

21:43:35 CTA引擎启动成功
21:43:36 double ema的交易合约IF1602无法找到
21:43:36 策略加载成功

VnTrader

系统 功能 算法 帮助

代码 p1609 卖五
名称 棕榈油p1609 卖四
方向类型 多 卖三
开平 开仓 卖二
价格 0.0000 卖一 5518.0 1355
数量 0 最新 5516.0 -0.04%
价格类型 限价 买一 5516.0 263
交易所 买二
货币 买三
产品类型 买四
交易接口 买五

发单
全撤

合约代码	名称	最新价	成交量	持仓量	开盘价	最高价	最低价	买一价	买一量	卖一价	卖一量	时
p1609	棕榈油p1609	5516.0	200268	788240	5504.0	5520.0	5482.0	5516.0	263	5518.0	1355	21:44
p1608	棕榈油p1608	5476.0		6				5464.0	1	5570.0	1	99:99
p1607	棕榈油p1607	5510.0		2				5400.0	1	5498.0	1	99:99
p1606	棕榈油p1606	5436.0		14				5338.0	1	5434.0	1	99:99
p1605	棕榈油p1605	5426.0	9212	145462	5430.0	5432.0	5406.0	5426.0	311	5428.0	84	21:44
m1609	豆粕m1609	2341.0	177590	2653822	2340.0	2346.0	2337.0	2341.0	1349	2342.0	1187	21:44
m1608	豆粕m1608	2351.0		146				2348.0	6	2354.0	1	21:43
m1607	豆粕m1607	2368.0		324				2345.0	3	2362.0	6	21:44
m1605	豆粕m1605	2306.0	4742	421072	2305.0	2309.0	2302.0	2305.0	94	2306.0	12	21:44
IF1609	沪深300IF1609	3015.0	458	4051	3015.4	3023.2	2994.6	3013.2	1	3017.4	1	99:99
IF1606	沪深300IF1606	3139.0	1100	7759	3139.6	3144.8	3124.0	3139.0	1	3142.0	2	99:99
IF1605	沪深300IF1605	3193.8	1040	1602	3191.0	3199.0	3179.6	3191.2	1	3194.6	1	99:99
IF1604	沪深300IF1604	3239.4	14521	35953	3227.8	3244.4	3222.2	3239.0	2	3239.2	1	99:99

账户	昨结	净值	可用	手续费	保证金	平仓盈亏	持仓盈亏	接口
000200002874	9619242413.52	9613794263.52	9528094488.52		85699771.0		-5448150.0	XSPEED

成交	委托	持仓	
合约代码	名称	方向	持仓量
p1605	棕榈油p1605	空	1700
i1606	铁矿石i1606	空	30
i1606	铁矿石i1606	多	1
i1605	铁矿石i1605	空	37010
i1605	铁矿石i1605	多	9

以“海龟交易法则”为例

趋势型策略原理

常见的交易策略类型



交易系统的核心要素

- 市场：买卖什么？
- 头寸规模：买卖多少？
- 入市：什么时候买进？
- 止损：什么时候放弃一个亏损的头寸？
- 退出：什么时候退出一个盈利的头寸？
- 战术：怎么买卖？

海龟交易系统就是一个
包含了上述所有要素的
“完整的”交易系统

头寸规模计算

□ 波动性N: $N = (19 * PDN + TR) / 20$

■ PDN = 前一日的N值

■ TR = 当日的真实波动幅度

□ 真实波幅 = $\text{Max} (H-L, |H-PDC|, |PDC-L|)$

■ H = 当日最高价

■ L = 当日最低价

■ PDC = 前一日收盘价

□ 头寸规模单位 (手数)

= 账户的 1% / ($N * \text{每一最小交易单位}$)

(举例)

- 假设股票中国平安的N值 ($\approx$ ATR) 为5元
 - 账户的本金为100万元, 那么账户的1%就是10000元
 - 最小交易单位是1手股票, 也就是100股
- 那么对于中国平安这只股票, 可以分配的头寸规模单位是:
 - $10000 / (5 * 100) = 20$ (手股票)
- 即: 给中国平安分配的头寸规模单位是20手

□ 系统1：以20日突破为基础的短期系统

- 如上次突破是赢利性突破，那么当前的入市信号将被忽略
- 55日突破点保障信号

□ 系统2：以55日突破为基础的长期系统

□ 在突破点建立1个单位的头寸

- 按 $N/2$ 的价格间隔逐步扩大头寸
- 以上一份订单的实际成交价为基准

□ 直到总头寸达到规模上限

止损

退出

□ 任何一笔交易风险程度不超过账户的2%

□ 价格变动上限是 $2N$

□ 对于加仓情况，止损点上浮

□ 系统1:

■ 10日反向突破退出

□ 系统2:

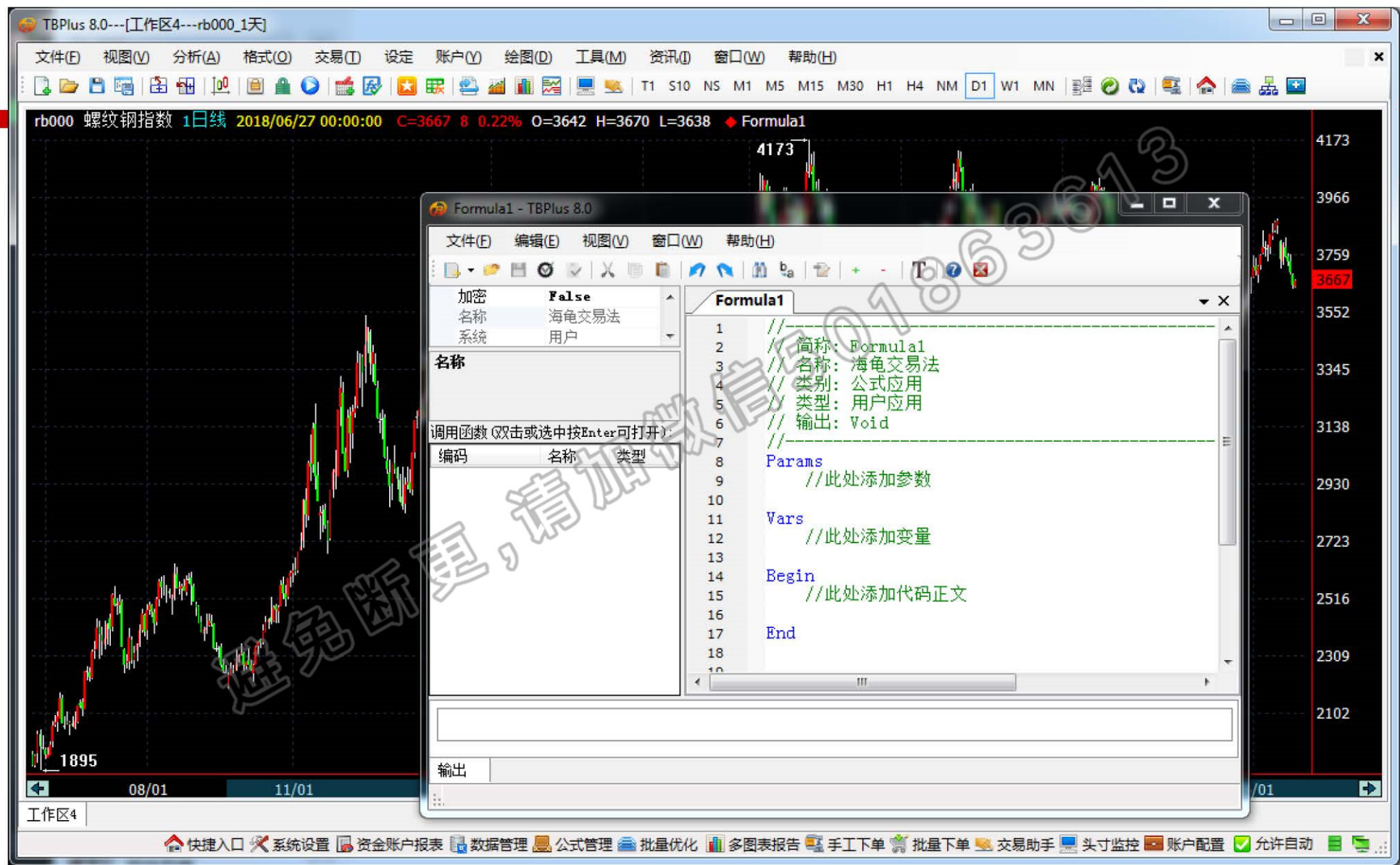
■ 20日反向突破退出

动手实践，从策略思想到编程实现，逐步求精

用TB开发商品期货趋势型策略

环境准备

- ❑ 运行TB，登录，新建工作区
- ❑ 加载交易品种：螺纹钢（指数），日K级别
- ❑ 设置商品，增加样本（K线）数到3000
- ❑ 设置公式应用，允许连续建仓，最多10次，最大持仓限制增加到2000合约
- ❑ 新建“用户应用”Formula1；文件菜单，公式管理，找到Formula1，选中并执行编译
- ❑ 插入公式应用，加载Formula1
- ❑ 如下页图，开始代码编辑.....



休息一下
5分钟后回来

避免断更，请加微信501863613

初步实现

□ 建仓信号

- 突破（20日）前高：开多信号
- 跌破（20日）前低：开空信号

□ 仓位计算（无持仓情况下）

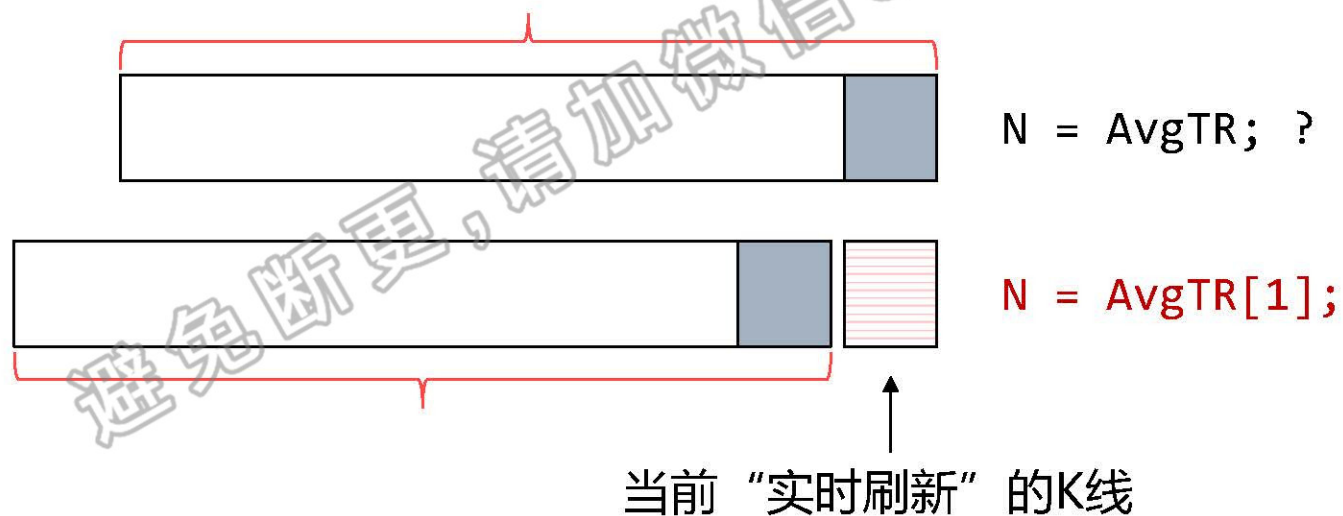
- 突破时建立一个初始的头寸单位

□ 退出（有持仓情况下）

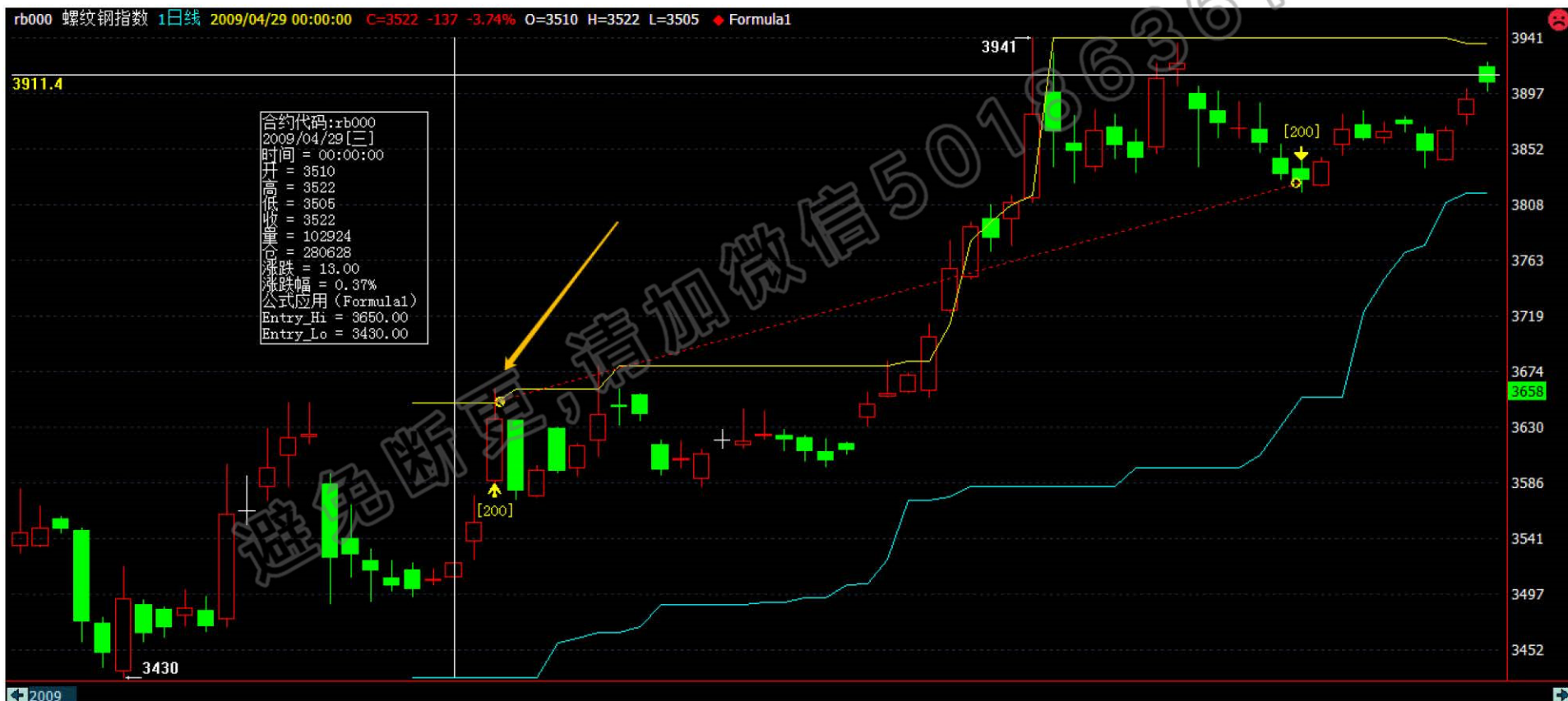
- 跌破（10日）前低时平多
- 突破（10日）前高时平空

序列变量平移取值

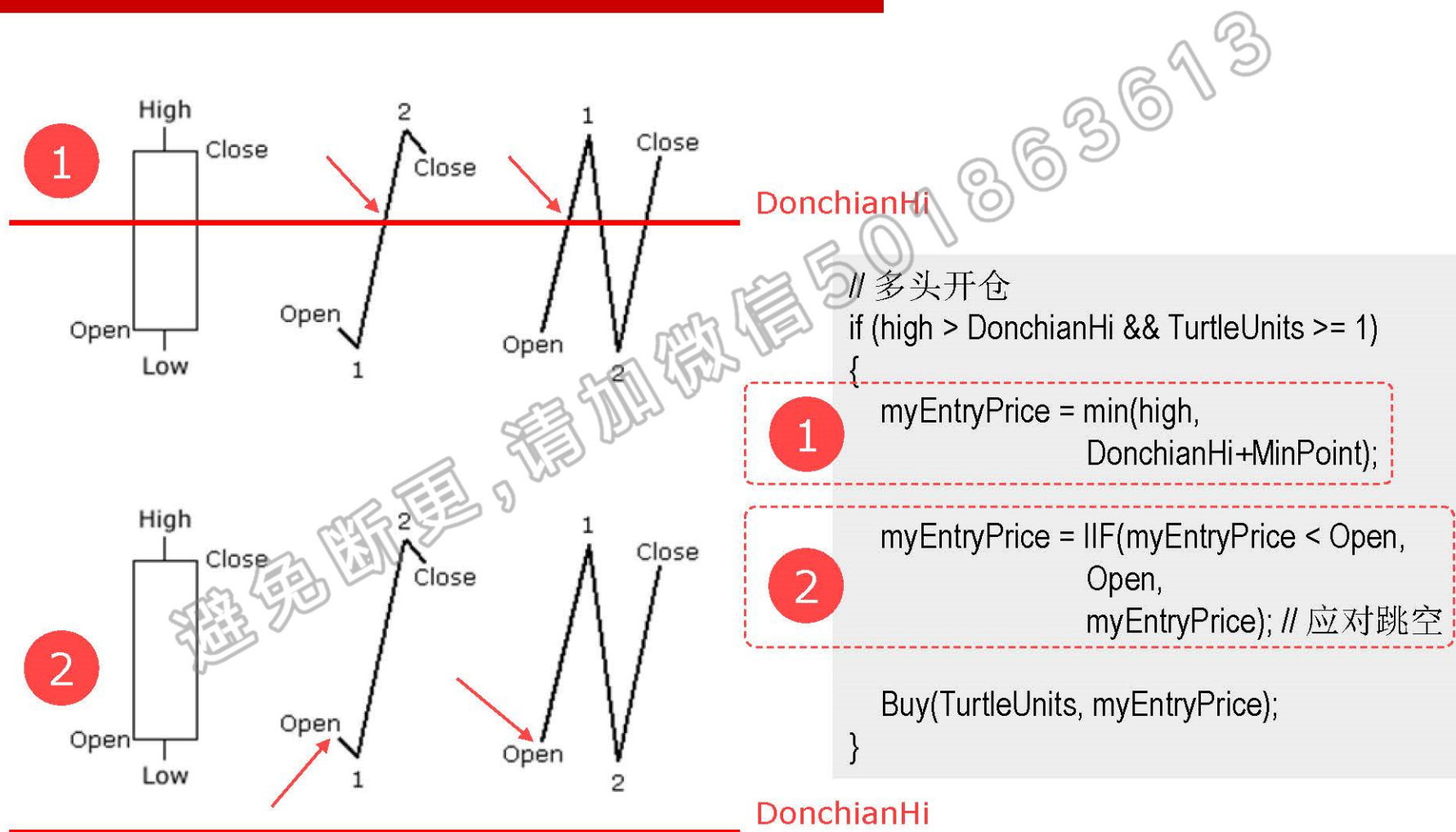
```
AvgTR = XAverage(TrueRange, ATRLength);  
N = AvgTR[1];
```



突破入场示意图



入场点代码解析



Params

```
Numeric RiskRatio(1); // 1% risk ratio  
Numeric ATRLength(20); // ATR window size  
Numeric boLength(20); // breakout  
Numeric teLength(10); // trailing exit
```

Vars

```
Numeric N;  
NumericSeries AvgTR;  
Numeric TotalEquity;  
Numeric TurtleUnits;  
NumericSeries DonchianHi;  
NumericSeries DonchianLo;  
Numeric MinPoint;  
Numeric ExitLowestPrice;  
Numeric ExitHighestPrice;  
Numeric myEntryPrice;  
Numeric myExitPrice;
```

Begin

```
if (!CallAuctionFilter()) return;  
  
MinPoint = MinMove * PriceScale;  
AvgTR = XAverage(TrueRange, ATRLength);  
N = AvgTR[1];  
TotalEquity = Portfolio_CurrentCapital() + Portfolio_UsedMargin();  
TurtleUnits = (TotalEquity*RiskRatio/100) / (N*ContractUnit()*BigPointValue());  
TurtleUnits = IntPart(TurtleUnits);
```

【初步实现的（简化版）代码】


```
DonchianHi = HighestFC(High[1], boLength);
DonchianLo = LowestFC(Low[1], boLength);
// PlotNumeric("Entry_Hi", DonchianHi);
// PlotNumeric("Entry_Lo", DonchianLo);

ExitLowestPrice = LowestFC(Low[1], teLength);
ExitHighestPrice = HighestFC(High[1], teLength);

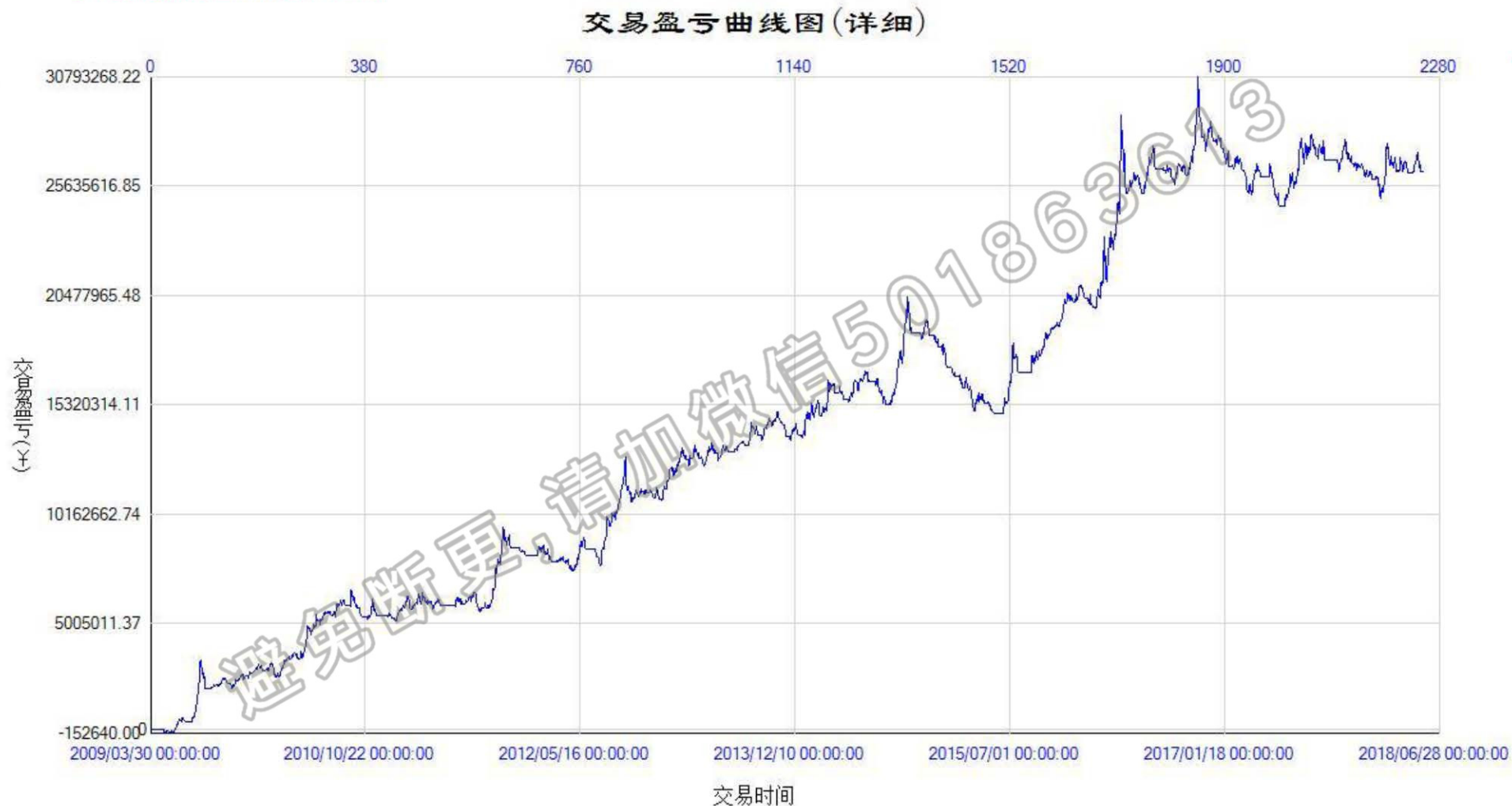
if (MarketPosition == 0)
{
    // 多头开仓
    if (high > DonchianHi && TurtleUnits >= 1)
    {
        myEntryPrice = min(high, DonchianHi+MinPoint);
        myEntryPrice = IIF(myEntryPrice < Open, Open, myEntryPrice); // 应对跳空情况
        Buy(TurtleUnits, myEntryPrice);
    }
    // 空头开仓
    if (Low < DonchianLo && TurtleUnits >= 1)
    {
        myEntryPrice = max(low, DonchianLo-MinPoint);
        myEntryPrice = IIF(myEntryPrice > Open, Open, myEntryPrice); // 应对跳空情况
        SellShort(TurtleUnits, myEntryPrice);
    }
}
```

```

if (MarketPosition == 1)
{
    if (Low < ExitLowestPrice)
    {
        // 多头退出
        myExitPrice = max(Low, ExitLowestPrice-MinPoint);
        myExitPrice = IIF(myExitPrice > Open, Open, myExitPrice); // 应对跳空情况
        Sell(0, myExitPrice);
    }
    else {
        // 多头加仓逻辑... 多头止损逻辑...
    }
}
else if (MarketPosition == -1)
{
    if (High > ExitHighestPrice)
    {
        // 空头退出
        myExitPrice = min(High, ExitHighestPrice+MinPoint);
        myExitPrice = IIF(myExitPrice < Open, Open, myExitPrice); // 应对跳空情况
        BuyToCover(0, myExitPrice);
    }
    else {
        // 空头加仓逻辑... 空头止损逻辑...
    }
}
}
End

```

【简化版回测结果】



【螺纹钢】总收益：263.10%，最大回撤：14.98%，胜率：51.76%，盈亏比：2.49

完整逻辑

□ 增仓动作（有持仓情况下）

- 多头：相对于上次入场价格，每上涨 $N/2$ 就新增一个单位的多头头寸，每根K线内可多次加仓
- 空头：相对于上次入场价格，每下跌 $N/2$ 就新增一个单位的空头头寸，每根K线内可多次加仓

□ 退出（有持仓情况下）

- 新增规则：相对于上次入场价格，如超过最大允许的损失幅度（ $2*N$ ），则清仓
- 加仓所在的那根K线内，不进行止损

其它细节

- 长周期突破点保障信号
 - 入市过滤条件
- 前次突破失败与否的判别
 - 盈利性退出
 - 亏损性退出

```
if (MarketPosition == 0 && \  
    (!LastProfitableTradeFilter or PreBreakoutFailure))  
{  
    // ...  
}
```

Params

```
Numeric RiskRatio(1);           // % Risk Per N ( 0 - 100)
Numeric ATRLength(20);          // 平均波动周期 ATR Length
Numeric boLength(20);           // 短周期 BreakOut Length
Numeric fsLength(55);           // 长周期 FailSafe Length
Numeric teLength(10);           // 离市周期 Trailing Exit Length
Bool LastProfitableTradeFilter(True); // 使用入市过滤条件
```

Vars

```
Numeric MinPoint;               // 最小变动单位
NumericSeries AvgTR;            // ATR
Numeric N;                      // N 值
Numeric TotalEquity;            // 按最新收盘价计算出的总资产
Numeric TurtleUnits;           // 交易单位
NumericSeries DonchianHi;       // 唐奇安通道上轨, 延后1个Bar
NumericSeries DonchianLo;       // 唐奇安通道下轨, 延后1个Bar
NumericSeries fsDonchianHi;     // 唐奇安通道上轨, 延后1个Bar, 长周期
NumericSeries fsDonchianLo;     // 唐奇安通道下轨, 延后1个Bar, 长周期
Numeric ExitHighestPrice;       // 离市时判断需要的N周期最高价
Numeric ExitLowestPrice;        // 离市时判断需要的N周期最低价
Numeric myEntryPrice;           // 开仓价格
Numeric myExitPrice;            // 平仓价格
Bool SendOrderThisBar(False);   // 当前Bar有过交易
NumericSeries preEntryPrice(0); // 前一次开仓的价格
BoolSeries PreBreakoutFailure(false); // 前一次突破是否失败
```

Begin

```
// 集合竞价过滤
if (!CallAuctionFilter()) return;
```

【TB自带示例（完整版）代码】

```
If(BarStatus == 0)
{
    preEntryPrice = InvalidNumeric;
    PreBreakoutFailure = false;
}

MinPoint = MinMove*PriceScale;
AvgTR = XAverage(TrueRange,ATRLength);
N = AvgTR[1];
TotalEquity = Portfolio_CurrentCapital() + Portfolio_UsedMargin();
TurtleUnits = (TotalEquity*RiskRatio/100)/(N * ContractUnit()*BigPointValue());
TurtleUnits = IntPart(TurtleUnits); // 对小数取整

DonchianHi = HighestFC(High[1],boLength);
DonchianLo = LowestFC(Low[1],boLength);

fsDonchianHi = HighestFC(High[1],fsLength);
fsDonchianLo = LowestFC(Low[1],fsLength);

ExitLowestPrice = LowestFC(Low[1],teLength);
ExitHighestPrice = HighestFC(High[1],teLength);

Commentary("N="+Text(N));
Commentary("preEntryPrice="+Text(preEntryPrice));
Commentary("PreBreakoutFailure="+IIFString(PreBreakoutFailure,"True","False"));
```

```

// 当不使用过滤条件, 或者使用过滤条件并且条件为PreBreakoutFailure为True进行后续操作
If(MarketPosition == 0 && ((!LastProfitableTradeFilter) Or (PreBreakoutFailure)))
{
    // 突破开仓
    If(High > DonchianHi && TurtleUnits >= 1)
    {
        // 开仓价格取突破上轨+一个价位和最高价之间的较小值, 更接近真实情况, 并能尽量保证成交
        myEntryPrice = min(high,DonchianHi + MinPoint);
        myEntryPrice = IIF(myEntryPrice < Open, Open,myEntryPrice); // 跳空用开盘价代替
        preEntryPrice = myEntryPrice;
        Buy(TurtleUnits,myEntryPrice);
        SendOrderThisBar = True;
        PreBreakoutFailure = False;
    }

    If(Low < DonchianLo && TurtleUnits >= 1)
    {
        // 开仓价格取突破下轨-一个价位和最低价之间的较大值, 更接近真实情况, 并能尽量保证成交
        myEntryPrice = max(low,DonchianLo - MinPoint);
        myEntryPrice = IIF(myEntryPrice > Open, Open,myEntryPrice); // 跳空用开盘价
        preEntryPrice = myEntryPrice;
        SendOrderThisBar = True;
        SellShort(TurtleUnits,myEntryPrice);
        SendOrderThisBar = True;
        PreBreakoutFailure = False;
    }
}

```



```

// 长周期突破开仓 Failsafe Breakout point
If(MarketPosition == 0)
{
    Commentary("fsDonchianHi="+Text(fsDonchianHi));
    If(High > fsDonchianHi && TurtleUnits >= 1)
    {
        // 开仓价格取突破上轨+一个价位和最高价之间的较小值, 更接近真实情况, 并能尽量保证成交
        myEntryPrice = min(high,fsDonchianHi + MinPoint);
        myEntryPrice = IIF(myEntryPrice < Open, Open,myEntryPrice); // 跳空用开盘价代替
        preEntryPrice = myEntryPrice;
        Buy(TurtleUnits,myEntryPrice);
        SendOrderThisBar = True;
        PreBreakoutFailure = False;
    }

    Commentary("fsDonchianLo="+Text(fsDonchianLo));
    If(Low < fsDonchianLo && TurtleUnits >= 1)
    {
        // 开仓价格取突破下轨-一个价位和最低价之间的较大值, 更接近真实情况, 并能尽量保证成交
        myEntryPrice = max(low,fsDonchianLo - MinPoint);
        myEntryPrice = IIF(myEntryPrice > Open, Open,myEntryPrice); // 跳空用开盘价代替
        preEntryPrice = myEntryPrice;
        SellShort(TurtleUnits,myEntryPrice);
        SendOrderThisBar = True;
        PreBreakoutFailure = False;
    }
}

```

```

If(MarketPosition == 1) // 有多仓的情况
{
    Commentary("ExitLowestPrice="+Text(ExitLowestPrice));
    If(Low < ExitLowestPrice)
    {
        myExitPrice = max(Low,ExitLowestPrice - MinPoint);
        myExitPrice = IIF(myExitPrice > Open, Open,myExitPrice); // 跳空用开盘价代替
        Sell(0,myExitPrice);    // 数量用0的情况下将全部平仓
    }Else
    {
        If(preEntryPrice!=InvalidNumeric && TurtleUnits >= 1)
        {
            If(Open >= preEntryPrice + 0.5*N) // 如开盘超过设定的1/2N,则直接用开盘价增仓
            {
                myEntryPrice = Open;
                preEntryPrice = myEntryPrice;
                Buy(TurtleUnits,myEntryPrice);
                SendOrderThisBar = True;
            }
            while(High >= preEntryPrice + 0.5*N) // 以最高价为标准, 判断能进行几次增仓
            {
                myEntryPrice = preEntryPrice + 0.5 * N;
                preEntryPrice = myEntryPrice;
                Buy(TurtleUnits,myEntryPrice);
                SendOrderThisBar = True;
            }
        }
    }
}

```

```

// 止损指令
If(Low <= preEntryPrice - 2 * N && SendOrderThisBar == false) // 加仓Bar不止损
{
    myExitPrice = preEntryPrice - 2 * N;
    myExitPrice = IIF(myExitPrice > Open, Open, myExitPrice); // 跳空用开盘价代替
    Sell(0, myExitPrice); // 数量用0的情况下将全部平仓
    PreBreakoutFailure = True;
}
}
}Else If(MarketPosition == -1) // 有空仓的情况
{
    // 求出持空仓时离市的条件比较值
    Commentary("ExitHighestPrice="+Text(ExitHighestPrice));
    If(High > ExitHighestPrice)
    {
        myExitPrice = Min(High, ExitHighestPrice + MinPoint);
        myExitPrice = IIF(myExitPrice < Open, Open, myExitPrice); // 跳空用开盘价代替
        BuyToCover(0, myExitPrice); // 数量用0的情况下将全部平仓
    }Else
    {
        If(preEntryPrice != InvalidNumeric && TurtleUnits >= 1)
        {
            If(Open <= preEntryPrice - 0.5*N) // 如开盘超过设定的1/2N,则直接用开盘价增仓
            {

```

```

        myEntryPrice = Open;
        preEntryPrice = myEntryPrice;
        SellShort(TurtleUnits,myEntryPrice);
        SendOrderThisBar = True;
    }

    while(Low <= preEntryPrice - 0.5*N) // 以最低价为标准, 判断能进行几次增仓
    {
        myEntryPrice = preEntryPrice - 0.5 * N;
        preEntryPrice = myEntryPrice;
        SellShort(TurtleUnits,myEntryPrice);
        SendOrderThisBar = True;
    }
}

// 止损指令
If(High >= preEntryPrice + 2 * N &&SendOrderThisBar==false) // 加仓Bar不止损
{
    myExitPrice = preEntryPrice + 2 * N;
    myExitPrice = IIF(myExitPrice < Open, Open,myExitPrice); // 跳空用开盘价代替
    BuyToCover(0,myExitPrice); // 数量用0的情况下将全部平仓
    PreBreakoutFailure = True;
}

}

}

End

```

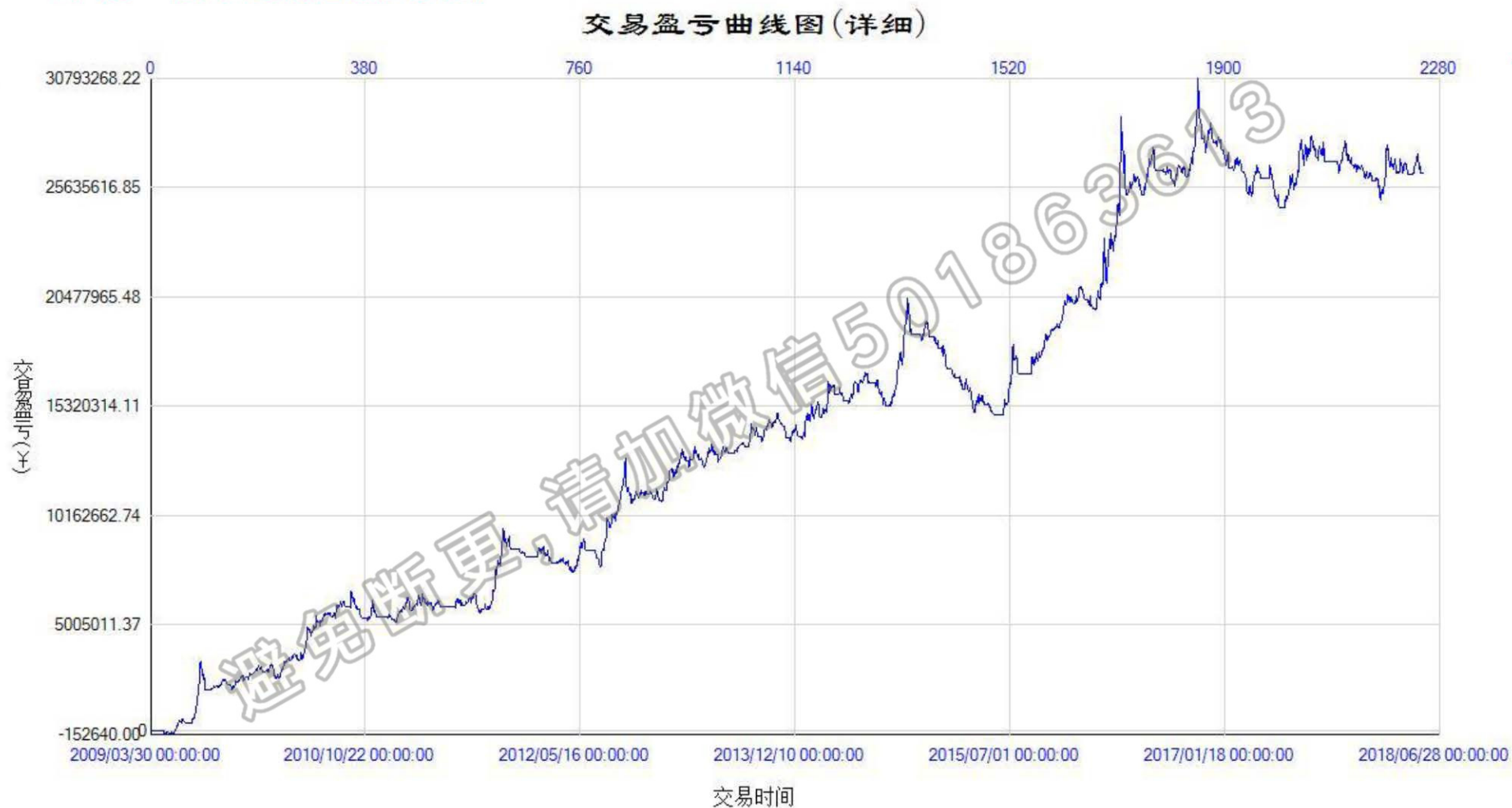



【完整版回测结果】



【螺纹钢】总收益：700.22%，最大回撤：21.79%，胜率：50.09%，盈亏比：2.01

对比 - 【简化版回测结果】



【螺纹钢】总收益：263.10%，最大回撤：14.98%，胜率：51.76%，盈亏比：2.49

思考

- 完整实现的效果一定比简化版好吗？
- 思考问题的本质
 - 突破：趋势型策略，顺势加仓，在什么情况下回调风险大？
 - 对品种、周期的选择有什么要求？
- 尝试改写完整版的代码？
 - 同时进行跨市场、多品种的交易
- 测试不同级别：60分钟、30分钟、周线
- 测试多个品种：RB -> J, RU, PTA

期货交易策略的注意事项

- 移仓换月（主力合约切换）
- 杠杆效应（保证金交易）
- 波动幅度（隔夜跳空风险）
- 套利机会（跨期、跨品种）

避免断更，请加微信501863613

课后练习

- 用TB测试海龟交易法在不同周期、不同品种上的表现，撰写测试报告
 - 测试不同级别：60分钟、30分钟、周线
 - 测试多个品种：RB -> J, RU, PTA
 - 对比完整版和简化版的表现，分析原因

下节课预告

☐ 题目：用Python开发均值回复型股票策略

- 编程语言：Python

- 运行平台：聚宽 - <http://joinquant.com>

- 准备工作：

- ☐ 提前在聚宽上注册账号

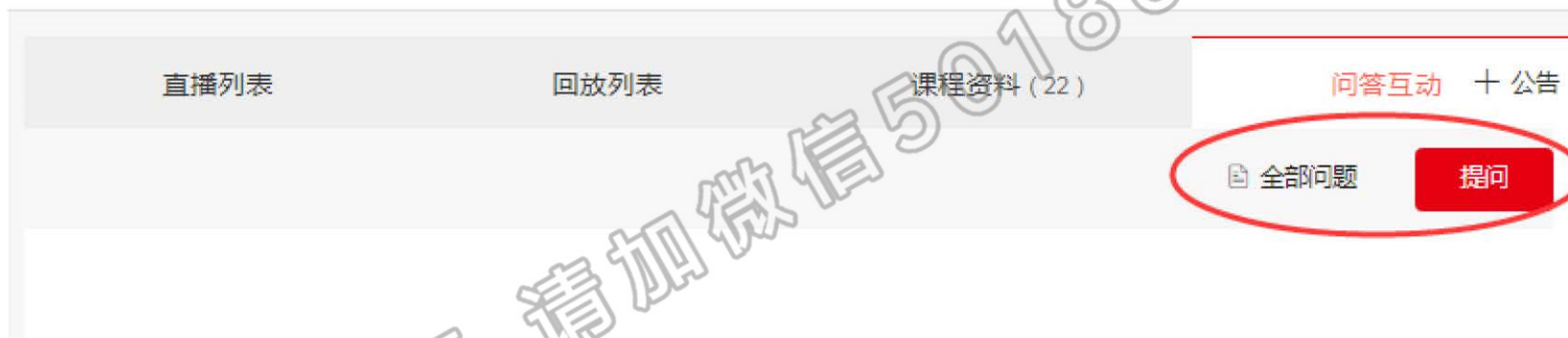
- ☐ 阅读并熟悉聚宽的API文档

- ☐ 自学Python和pandas

问答互动

在所报课的课程页面，

- 1、点击“全部问题”显示本课程所有学员提问的问题。
- 2、点击“提问”即可向该课程的老师 and 助教提问问题。



联系我们

小象学院：互联网新技术在线教育领航者

— 微信公众号：**小象学院**



避免断更，请加微信501863613

THANKS