

基于泛化的套利交易系统的设计与实现

王力文

(上海交通大学计算机科学与工程系 上海 200240) (上海机电工程研究所 上海 201109)

摘要 随着以股指期货为代表的金融衍生品的上市,针对国内金融市场将出现越来越多的对冲、期现套利、统计套利等较为复杂的交易策略等问题,提出了一种用程序替代人力进行复杂的运算和操作,实现大跨度的复杂交易,简化用户操作的套利交易解决方案,通过国外市场数据接口 SPTrader 和国内市场数据接口 CTP 获取行情数据,实现了对全球商品期货交易所的任意两合约的套利方法。首先介绍并分析了基于泛化套利交易系统的优势,然后对系统进行了分析和概要设计,重点阐述了策略设置、策略监控管理模块及系统逻辑架构,最后对系统的整体运行情况进行测试评估。

关键词 套利,泛化,程序化交易,策略监控,SPTrader

中图分类号 TP311 文献标识码 A

Design and Implementation of Arbitrage Trading System Based on Generalization

WANG Li-wen

(Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China)

(Shanghai Electro-Mechanical Engineering Institute, Shanghai 201109, China)

Abstract With the stock index futures as a representative of the financial derivatives coming into the market, more and more complex trading strategies such as hedging, arbitrage and statistical arbitrage are appearing in the domestic financial market. Therefore, this paper presented an arbitrage trading solution with procedures instead of human complex calculations and operations through the foreign market's data interface SPTrader and the domestic market data interface CTP. They can obtain market data to achieve the method of any two contracts in global Mercantile Exchange. This paper firstly introduced and analyzed the advantage of generalized arbitrage trading system, and then systematically analyzed and summarized the system design, which is focused on the policy setting, policy monitoring management module and system logic structure. And finally the overall operation of the system was evaluated.

Keywords Arbitrage, Generalization, Program trading, Strategy monitoring, SPTrader

1 前言

程序化、算法交易在发达国家的金融市场极为常见,套利交易由于收益稳定可靠、较低风险(甚至无风险)的特点,已经成为国际金融市场中不可或缺的一种交易手段。同时,随着市场的不断发展扩张以及上市品种的丰富多元化,越来越多的人也意识到其中蕴含的收入可观且数量巨大的套利机会。面对这些诱人的机会,很多人都希望通过跨品种、跨期限套利来减少单边敞口暴露造成的系统性风险损失,但因为缺少套利交易本身所要求的对市场的敏感与无法投入大量的时间和关注而止步不前。市面上有很多套利软件,但不同公司的不同价值主张导致用户面对林林总总的不同说辞而无法抉择,这也是我们的套利市场始终无法充分活跃起来的原因之一。

其实,套利交易的操作可以通过程序化的方式变得轻松易懂,通过计算机完成对多个数据的实时监控、呈现,同时基于其强大的计算能力进行评估和分析。由于现实中同时具备金融知识常识和编程代码技术的人才稀缺,市面上开发的大多数交易软件主要是面对金融从业人员,因此包含了不少行业术语,这要求使用者对金融交易有一定经验和了解,这也是

很多用户所不具备的硬性条件。再者,不同的交易软件在期货套利交易上都有不同的侧重(跨期、跨品种或跨市场),并没有开放一个完整的市场。

本文设计的套利交易系统正是提出了一种用程序替代人力进行复杂的运算和操作,实现大跨度的复杂交易,简化用户操作的解决方案,为因技术壁垒等原因而被阻挡在套利获益群体之外的人们提供了更自由、更开放的市场。

2 系统概述及原理

本系统的开发语言为 C++, C#; 开发工具为 Visual Studio 2013 和 Microsoft .NET Framework 4.5 的 Microsoft 图表控件。基于为用户呈现无任何限制的、完整的数据并实时更新数据,将交易分析过程简单化至所有用户都能理解并操作的理念,通过将不同交易接口通讯协议级别泛化,从而连接不同的交易市场,在对每个市场通过不同接口传输过来的期货合约信息进行整合,将其一并纳入我们的数据库中,其目的在于建立一个居于所有期货交易平台之上的整合型交易技术平台,最终实现对全球商品期货交易所的任意两合约套利的支持。

在客户端的具体操作中,用户只需要完成两个部分:1)套利公式的设计;2)战略的部署。套利公式是为方便用户部署战略而采用的一种简化设计。由于在任意的两份期货合约的套利交易中,不同的期货合约会因为同时跨了不止两个市场而导致价差受到汇率和商品本身市场价格的影响,从而在套利交易过程中面板上的价差过大或呈负数,这样的数据波动较为明显且不方便用户实施战略部署,所以我们引入了支持泛化的套利公式体系。在这个体系中,用户可以自己根据价差来编写和调整公式,使得面板上的价差达到理想值(一般理想值被认定为在 100 以内,而且波动幅度不会过于巨大)。所得的公式并没有标准值,也没有特定的求解方法,只需要用户通过设计和调整将数据变得易于分析,这是每一个用户都能确保达成的,不需要知识基础。

我们还做了很多方便用户操作的设计,包括简化国内交易接口,使系统更加智能,消灭了单合约锁仓的可能,开反方向有仓优先。希望以此帮助普通投资者减少因为错误下单导致的损失和多余的保证金成本。

3 泛化套利交易系统的优势

传统的套利系统针对两个特定的交易接口进行交易,无法涵盖所有市场,有一定的局限性,比如伦敦白银与国内白银相互跨市场交易,仅限于白银。而泛化是指由具体的、个别的扩大为一般的,它使得多态操作成为可能,即操作的实现是由它们所使用的对象的类而不是调用者确定的。简而言之,就是把相同的模式抽象出来,包含交易接口的一些通用功能模块(如行情模块、下单模块)都能在一定程度上接驳系统,因此交易泛化较广、较多。

使后台能够接入不同交易所提供的程序化交易接口,具有较高级别的应变性,不再受限于交易品种、交易市场和交易接口,由此通过程序化自动交易智能实现多个平台上任意两个不同标的的套利,如伦敦期货市场的白银不仅能与国内期货市场的白银实现跨市场交易,甚至能与国内非白银标的(铜等)实现跨期交易等。

在操作时只需要选定两个不同的期货品种,由后台的程序自动比较交易的买价、卖价差,并在交易时自动完成多个下单。在这样的运行模式下,程序替代人力进行复杂的运算和操作,将大跨度的复杂交易变得可能,简化了用户操作,也拓宽了其套利的视野。

与此同时,套利交易的类型也变得丰富,意味着套利市场的进一步广阔,套利机会的成倍增加,获取利润的空间也得到扩大,全球商品期货交易市场中的任意两合约的交易模式中蕴含着无限的套利可能。

4 套利交易系统的设计

4.1 术语、定义、缩写

接口:泛指实体把自己提供给外界的一种抽象化物(可以为另一实体),用以由内部操作分离出外部沟通方法,使其内部能被修改而不影响外界其他实体与其交互的方式。本软件中的接口主要由交易系统衍生而出,用于软件与交易系统之间的信息交换。

综合交易平台(Comprehensive Transaction Platform, CTP):由上海期货信息技术有限公司(上海期货交易所的全资子公司)开发的期货交易平台,CTP平台以“新一代交易所

系统”的核心技术为基础,具有稳定、高速、开放式接口,适合程序化交易软件运用和短线炒单客户使用,同时有着开放的接口、优异的性能、集中部署的创新模式以及经验丰富的技术背景。

SPTrader(境外期货网上交易系统):一个快速的电子交易平台,提供宝贵的市场数据。主要功能包括多元化的国外期货交易产品、快捷稳定的交易平台、实时的市场报价、专业的图表分析和灵活的用户自设接口。使用时需由大陆经由香港进行境外数据访问。

灾备:即灾难备援,它是指利用科学的技术手段和方法,提前建立系统化的数据应急方式,以应对灾难的发生。其内容包括数据备份和系统备份、业务连续规划、人员架构、通信保障、危机公关、灾难恢复规划、灾难恢复预案、业务恢复预案、紧急事件响应、第三方合作机构和供应链的危机管理等。

4.2 系统总体设计

本系统旨在为不同交易所提供的程序化交易提供一套实现任意市场上两个不同标的的套利解决方案,即通过整合国内外各大主要证券、商品现货、外汇、期货合约及其衍生品(如上海期货交易所、伦敦金属交易所等)市场数据的实时获取,主要实现了对任意两种交易标的的正向或反向套利。

程序化交易是证券交易方式的一次重大的创新。程序化交易主要是指利用计算机和现代化网络系统,按照预先设置好的交易模型和规则,在模型条件被触发时,由计算机瞬间完成组合交易指令,实现自动下单的一种新兴的电子化交易方式。设计人员用电脑程序对将交易策略的逻辑与参数进行运算后,再将交易策略系统化。每个投资者都可以根据自己的投资经验和策略,编写自己的交易模型,进行电脑自动交易。

如图 1 所示,系统分为 3 层,分别是数据源层、配置管理层、交易管理层。图 2 的系统边界图显示了系统主要利用国外市场数据接口 SPTrader 和国内市场数据接口 CTP 获取行情数据并下达交易指令的流程。套利部分的主要内容可实现对任意两种交易标的的价差分析,自定义交易策略编辑,本外币汇率修正,账户资金持仓委托查询,止盈止损灾备控制。

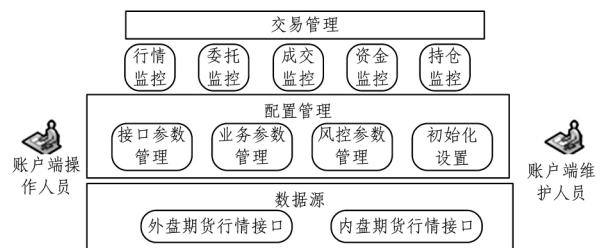


图 1 系统功能架构图

业务场景可以理解为用户首先由用户提供策略及相关交易参数,并由系统获取行情数据通道。系统对具体策略决策进行实时监控,并根据用户的参数实时向交易系统报单套利。

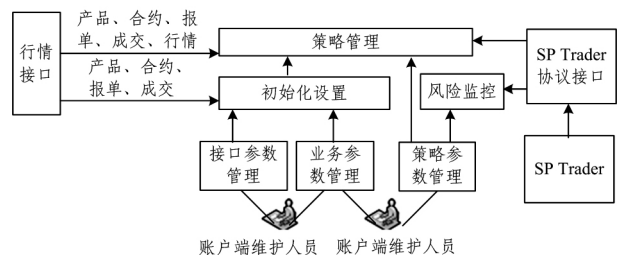


图 2 系统边界图

图 2 中风险监控的二级菜单即为图 1 中的行情监控、委托监控、成交监控、持仓监控和资金监控。

4.3 参数管理模块

参数管理主要实现当账户端使用策略管理功能时对不同策略指标进行相关的阈值设置。

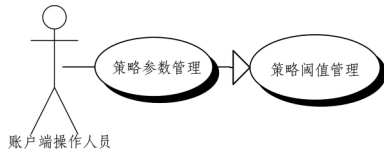


图 3 参数管理用例图

参数管理的一级菜单分为外部接口参数管理和基本业务参数管理。前者又分为交易数据接口设置(接入外盘 SPTrader API 网络协议和內盘 CTP 接口)和行情数据接口设置(外盘通过 SPTrader API 网络请求下单,內盘通过 CTP 传入期货交易所前置机)。后者的二级菜单分为策略设置和风险监控设置。

4.4 系统初始化模块

系统初始化主要实现每日初始化时需要进行的数据、参数准备工作。

如图 4 所示,通过设置策略参数将各用户的组合交易数据文件进行装载。在初始化阶段,由系统端维护人员根据用户、用户交易组合配置以及用户交易组合要求,对策略参数进行装载,以实现用户对交易参数的设置。持仓、资本数据初始化是对用户的昨持仓、资本数据进行初始化,作为当天的初始持仓、资本数据并导入系统。而合约信息初始化则是对市场行情中的合约基本信息数据进行初始化,并将结果导入系统。

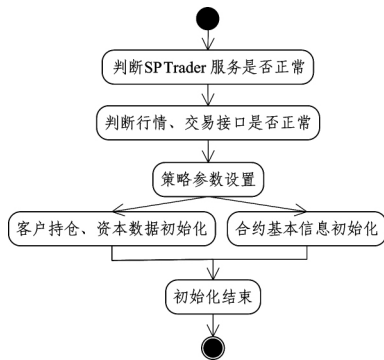


图 4 系统初始化活动图

4.5 监控项管理模块

如表 1 所列,监控项管理主要实现一些必要的参数信息设置,如装载策略管理、交易监控项参数文件、启动或停止策略分析、交易监控。

表 1 监控项管理示意表

一级菜单	二级菜单	说明
策略分析	策略组合管理	交易组合构建
	策略组合分析	交易组合价格变化频率
策略监控	组合交易开关	开平仓条件设定

如图 5 所示,在初始化阶段,由账户端的维护人员根据用户提交的交易组合要求对交易项参数文件进行装载。在分析用户策略后,对特定用户、特定策略项进行启动和停止操作,

前期先假定盘中不会进行启动或停止监控操作。

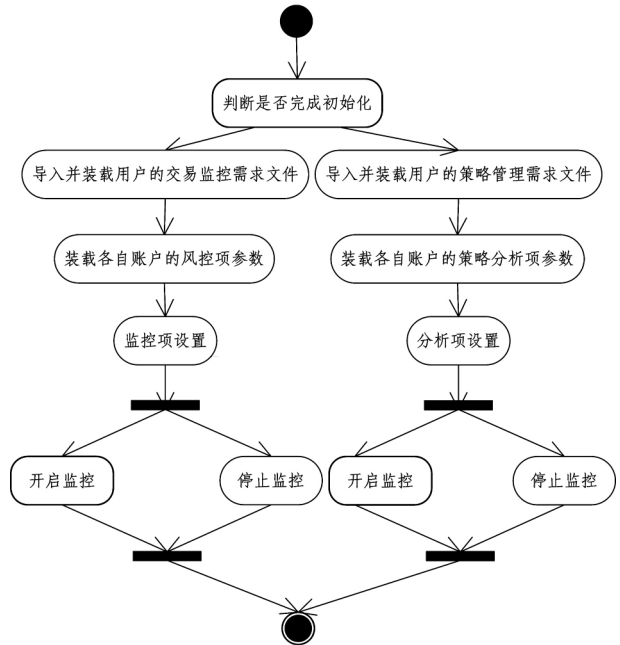


图 5 监控项管理活动图

4.6 逻辑架构设计

输入适配层:主要用途为获取程序所需的基本数据,由若干的输入适配器组成,这些输入适配器主要完成 4 个部分的输入。

- 1)从期货接口、现货接口等行情系统获取源数据。
- 2)用户输入套利标的资产的自定义价差计算公式。
- 3)用户输入套利策略的方向、开仓条件等策略。
- 4)用户输入平仓条件、撤单控制等信息。

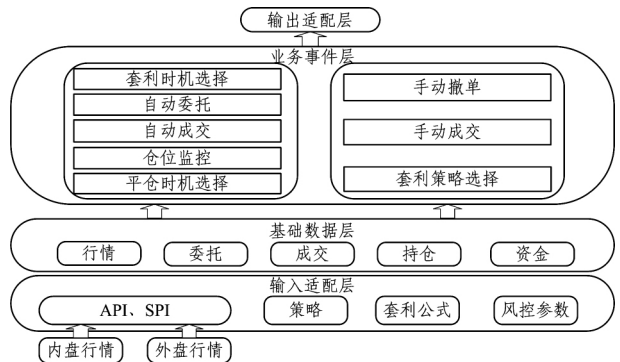


图 6 系统逻辑架构图

输入适配层是外部数据接入本系统的通道,主要从外部获取数据,并将数据进行初步整合,分发到基础数据层。

基础数据层:把输入后的数据进行简单的处理之后,形成最基本的数流,供后续的数据处理订阅使用。基础数据服务提供的数据包括行情、委托、成交、持仓、基金等,也可以根据系统的需要逐步扩展。不同的数据流内容可以根据不同的业务来定义不同的类型,同样的数据也可以根据需求提供不同形式的多套数据副本包。基础数据也可以通过 API 和 SPI (Serial Peripheral Interface, 串行外设接口) 提供给外部交易系统使用。

基础数据层主要从业务数据的角度进行扩展,根据业务需要,接入更多的数据,提供范围更广的数据服务。

业务事件层:接收来自基础数据层的数据,按照具体的业务规则,从基础的数据流中捕获特定的业务事件。本系统针对特定的套利策略进行监控,从而发现市场和交易过程中的套利时机,并进行开仓平仓操作。业务事件同样以数据流的形式供输出适配层使用,也可以公开给外部系统订阅使用。

业务事件层主要从业务角度进行扩展,横向增加不同的业务功能,纵向实现对不同适配器的扩展性处理。

输出适配层:通过收集相应基础数据层和业务时间层发出的指令,通过特定的接口,将事件流和数据流发送给相关的系统,比如内外盘交易系统和数据记录系统等。

5 系统的应用及测试

5.1 系统的应用

5.1.1 交易流程模式化

软件客户端源代码采用 Visual Studio 2013 开发,为了不影响使用,需用 VS2013 打开 SLN 文件,使主程序开始运行;同时还需安装 SPTrader,图 7 为进入外盘 SPTrader 的主界面。点击“工具”菜单下的“交易应用程序界面”后即可部署交易接口,在 SPTrader API 通讯监控窗口中设置相关参数,并启动 API。

打开套利交易系统登录界面,进行相应通讯设置后,分别输入内盘、外盘帐户密码并登录,即可显示如图 8 所示的系统主界面。在后 A 和先 B 分别选择要进行套利的品种,点击添加,根据汇率、交割费用、交易费用等在价差公式处输入两个套利品种的换算公式,查看买价差和卖价差。

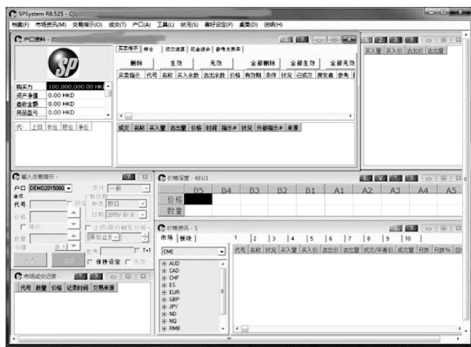


图 7 外盘 SPTrader 主界面

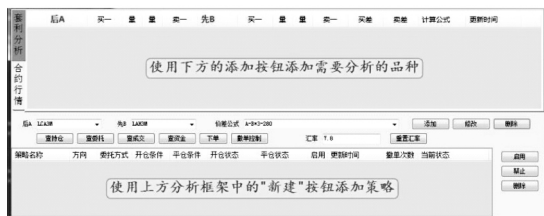


图 8 套利交易系统主界面

交易流程模式化体现了“监控→开仓→持仓→平仓→交易完成”的功能。

5.1.2 交易策略参数化

如图 9 所示,点击“新建”按钮可添加套利策略,在“策略”窗口中可选择或输入委托方式、手数、方向、开仓平仓条件等并进行保存。通过“套利策略监控”窗口启用生成的策略,等待满足条件的策略成交。交易策略参数化体现在完成参数配置后即可实现交易策略的配置。

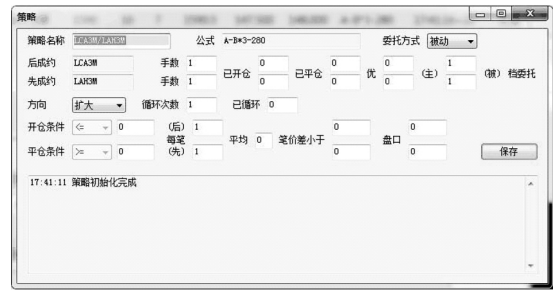


图 9 套利策略参数设定

5.1.3 交易监控可视化

由用户根据需求输入策略参数,包括套利公式、成约手数、已开仓、已平仓手数、委托方式、优先委托档数、套利方向等,经过参数配置输入适配器传递给套利交易服务器以进行后台计算与分析。策略启用后,通过判别先后成约价差以及是否符合开、平仓条件,系统会根据不同情况为用户显示策略状态,生成回报日志(见图 10),提示成约是否成交以及对应成交价格。

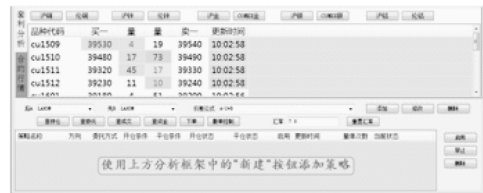


图 10 策略状态报告

除了上述提到的 3 点以外,交易过程的自动化也不言而喻。由 4.5 节的监控项管理模块设计和图 6 的逻辑架构分析可知,下单、撤单、追单、触发开仓、平仓、止盈、止损等业务都由系统自动完成。另外,通过风险监控模块中包含的行情、委托、成交、资金等监控项可以实时估算出收益、风险率等指标。

综上所述,系统中交易流程模式化、交易策略参数化、交易监控可视化、交易过程自动化、交易结果指标化这些机制和特点可以让对计算机技术或金融知识掌握尚浅的普通投资者灵活运用系统。

5.2 系统测试

本次测试环境为 Microsoft Windows Server 2008 R2 64Bit,16GB 内存,Intel Xeon E5620*2 处理器。功能测试包含了行情测试、套利面板测试、策略面板测试、策略监控测试。此外,还通过同时运行 15 个策略来测试系统的稳定性。整个测试过程中,Trade.vshost 主进程保持稳定,系统占用率在 10%左右。

关于行情测试,图 11 示出了沪铜的行情数据表,同样还有国内和国外交易账户手动下单的测试,在这里不一一列举。



图 11 行情数据表:沪铜

除了上述功能测试与性能测试外,内核测试也尤其重要。这里列举个别测试用例来说明如何对内核进行性能评估。

1)对核心初始化/退出 100 次的性能作出评估:
TEST(Core,InitExit100Times)

```

{
    int i;
    cout <<“该项测试用于对核心初始化/退出次的性能作出评估”<< endl;
    for (i=0;i<100;i++)
    {
        ASSERT_TRUE(SrvCoreInit());
        ASSERT_TRUE(SrvCoreExit());
    }
}

```

2)对 IOlock 单线程获取/释放 50000000 次的性能作出评估:

```

TEST (Core,OneThreadIOlockPerformance)
{
    cout <<“该项测试用于对 IOlock 单线程获取/释放 50000000 次的性能作出评估”<< endl;
    IOlock lock;
    int i;
    SrvCoreInitIOlock(&lock);
    for (i=0;i<50000000;i++)
    {
        SrvCoreEnterIOlock(&lock);
        SrvCoreLeaveIOlock(&lock);
    }
}

```

3)用于对 IOlock 双线程各获取/释放 25000000 次的性能作出评估:

```

DWORD WINAPI TwoThreadIOlockPerformance (LPVOID lpParameter)
{
    int i;
    IOlock *plock=(IOlock *)lpParameter;
    for (i=0;i<25000000;i++)
    {
        SrvCoreEnterIOlock(plock);
        SrvCoreLeaveIOlock(plock);
    }
    return 0;
}
TEST (Core,TwoThreadIOlockPerformance)
{
    cout <<“该项测试用于对 IOlock 双线程各获取/释放 000000 次的性能作出评估”<< endl;
    IOlock *plock=new IOlock;
    int i;

```

```

HANDLE hThread;
SrvCoreInitIOlock(plock);
hThread = CreateThread (NULL, 0, TwoThreadIOlockPerformance,plock,0,NULL);
for (i=0;i<25000000;i++)
{
    SrvCoreEnterIOlock(plock);
    SrvCoreLeaveIOlock(plock);
}
WaitForSingleObject(hThread,INFINITE);
CloseHandle(hThread);
delete plock;
}

```

结束语 本套利交易系统通过“傻瓜式”交易,不但使得没有同时掌握计算机技术和丰富金融知识的普通投资者拥有了更自由开放的平台,也让程序化交易策略变得更简单易懂、方便快捷。通过引入泛化的概念,实现了全球两两标的资产之间的套利模式,为投资者们提供了更大的自由度和更好的用户体验。然而,套利的盈利能力往往在一定程度上取决于系统的速度。据调查研究发现,拥有世界上最先进交易系统的 Citadel 公司在不到 3 分钟的时间内竟可以下达高达 2100 万股的交易指令,类似国外的优秀高频交易团队的一笔交易的时间大多在微妙级别,而本文系统使用外盘接口 SPTrader 时仅花在大陆到香港的网络延迟就在毫秒级别。换言之,使用 SPTrader 的一笔交易所花的时间对于优秀的交易团队而言可以完成上千笔交易,导致即使出现好的交易机会也因为比别人慢而抓不住。故在下一步工作中,将围绕因外盘接口 SPTrader 在交易速度上有所局限而降低了套利交易的盈利能力这一方面进行研究与改善。

参考文献

- [1] 高婷婷. 基于程序化的 ETF 套利交易系统的设计与实现[D]. 黑龙江:黑龙江大学,2013.
- [2] 高翔. 中国股指期货市场跨期套利策略的分析与实践[D]. 成都:西南财经大学,2012.
- [3] 王宏伟. 股指期货套期保值和套利策略分析[D]. 北京:中国科学院研究生院,2013.
- [4] 丁超群. 期货程序化套利平台研究与实现[D]. 上海:上海交通大学,2014.
- [5] BIALKOWSKI J. Stock Index Future Arbitrage in Emerging Markets: Polish Evidence[J]. International Review of Financial Analysis, 2008, 17(2): 363-381.
- [6] 吕鸿杰. 基于 SOA 技术框架数据分析系统的研究与实践[D]. 成都:西南交通大学,2008.
- [7] (上接第 503 页)
- [8] BUCKLEW J. A, Sadowsky, et al. A Contribution To The Theory Of Chernoff Bounds[J]. IEEE Transactions on Information Theory, 1991, 39(1): 75-75.
- [9] 顾雅娟, 仰枫帆. 基于修正的切尔诺夫界的尾部概率的估计[C]// 全国青年通信学术会议. 2010.
- [10] MALLOY M, NOWAK R. Sequential analysis in high-dimensional multiple testing and sparse recovery[J]. Mathematics, 2011, 42(4): 2661-2665.
- [11] http://plasma-lab.gforge.inria.fr/plasma_lab_doc/1.4.0/html/algorithms/mdp.html#smart-sampling-algorithms.
- [12] KIM Y, KIM M, KIM T H. Statistical Model Checking for Safety Critical Hybrid Systems: An Empirical Evaluation[M]. Hardware and Software: Verification and Testing. 2013: 162-177.
- [13] KWIATKOWSKA M, NORMAN G, SPROTON J. Probabilistic Model Checking of the IEEE 802.11 Wireless Local Area Network Protocol[M]. Process Algebra and Probabilistic Methods: Performance Modeling and Verification. Springer Berlin Heidelberg, 2002: 411-423.
- [14] PNUELI A, ZUCK L D. Probabilistic Verification[J]. Information & Computation, 1993, 103(1): 1-29.
- [15] <https://project.inria.fr/plasma-lab>.