

基于遗传编程的上证50指数 技术交易规则研究

瞿慧

南京大学 工程管理学院, 南京 210093

摘要:针对现有文献使用流行的技术交易规则检验市场有效性时存在的数据探测问题,提出基于遗传编程的技术交易规则,并将其用于中国股票市场有效性的检验。使用基本函数模块随机构建以树形结构表示的技术交易规则,采用遗传编程进行优化;采用可移动的训练窗-测试窗将研究数据划分为训练区间和测试区间,使用训练数据寻找最优技术交易规则,然后将其用于测试数据以检验优化结果的样本外性能。以2004年1月2日~2010年3月12日期间共计1503个交易日的上证50指数价格和成交量作为研究数据,研究结果表明,遗传编程优化得到的技术交易规则在考虑交易费用后,仍然能够较买入-持有策略获得统计显著的样本外超额收益,表明中国股票市场尚未达到弱式有效。

关键词:技术交易;市场有效性;遗传编程;优化

中图分类号:F830.91

文献标识码:A

文章编号:1672-0334(2010)05-0103-11

1 引言

有效市场假说将市场的有效性划分为弱式有效、半强式有效和强式有效3个层次,目前对中国股票市场有效性的检验主要集中在是否已达到弱式有效。在弱式有效的市场中,股票价格充分反映了其价格和成交量等历史交易数据中隐含的信息,任何基于历史价格、成交量和其他证券交易指标的投资策略都不可能较“买入-持有”策略获得超额收益,即技术分析无效。反之,如果存在能够产生超额收益的技术交易规则,即可否定市场的弱式有效性。

中国学者大多通过检验流行的技术交易规则的有效性来考察中国市场的弱式有效性,由于他们采用的技术交易规则是在多年市场起伏中由投资者逐步选择出来的,使这些检验实质上是用同一段历史数据先筛选出性能优越的技术交易规则,再评价这些技术交易规则的有效性,从而缺乏对技术交易规则有效性的样本外测试。可见,使用流行的技术交易规则检验市场的有效性存在数据探测的问题。尽管某些技术交易规则在多年的检验中可以获得显著的超额收益,但身处多年前的投资者却无从预知检

验结果,不可能从当时存在的大量技术交易规则中将它们选出并执行。因此,即使流行的技术交易规则存在超额收益,也不能完全否定市场的弱式有效性。

针对这一问题,本研究将检验基于遗传编程的技术交易规则的可获利性,将股票价格和成交量历史数据分为训练数据区间和测试数据区间,以实现技术交易规则有效性的真正样本外测试。在训练区间内根据设定的进化规则,逐代优化用树形结构表示的技术交易规则;然后使用测试数据检验所得最优技术交易规则的超额收益,并据此对中国股票市场的有效性做出评价。

2 相关研究评述

近年来,中国学者对技术交易规则在中国股票市场的有效性进行了不少研究,得到的结论并不一致。魏玉根实证发现,移动平均线规则运用于1992年11月~1999年10月的上证综指和深证成指,均可获得稳定的超额收益,因而得出中国市场尚未达到弱式有效的结论^[1];孙碧波采用1990年12月~2003

收稿日期:2010-04-20 修返日期:2010-09-06

基金项目:国家自然科学基金(70932003)

作者简介:瞿慧(1981-),女,江苏南通人,毕业于美国康奈尔大学,获博士学位,现为南京大学工程管理学院副教授,研究方向:计算金融等。E-mail:linda59qu@nju.edu.cn

年12月的上证指数研究发现,仅持有期可变的移动平均线规则能够产生超额收益^[2];曾劲松以1998年1月~2004年8月间的上证综指、深证综指以及交投活跃的个股为研究对象,发现移动平均线规则不能持续获得超额收益^[3];王志刚等以1996年1月~2005年12月间的上证综指为研究对象,证明移动平均线规则具有一定的预测能力,并指出技术交易规则的预测能力可能是由于股票收益率动态过程中的非线性特征^[4,5];邓杰等在不考虑交易成本的情况下,运用移动平均线和阻力线-支撑线等技术交易规则,实证研究1990年12月~2008年3月上证综指的有效性,发现仅阻力线-支撑线规则可以获得超额收益^[6]。然而,利用移动平均线、阻力线-支撑线等流行的技术交易规则检验市场的弱式有效性无法避免数据探测问题^[7,8]。因此,有必要采用更加严谨的方法检验技术交易规则在中国的有效性以及中国股票市场的弱式有效性。

随着计算机技术的发展,遗传编程这一计算密集型的优化方法在国外学者对金融市场有效性的研究中得到越来越多的应用。研究侧重于两个方面,一部分学者考察遗传编程优化得到的金融时间序列预测方法,进而分析建立在此预测基础上的技术交易规则的有效性^[9-11]。由于交易费用的存在,对金融时间序列的预测能力与相应买/卖决策的获利情况并不一致。另一方面,更多的学者通过检验遗传编程优化得到的技术交易规则在外汇^[12]、股票^[13,14]、股指^[8,15,16]、期货^[17,18]交易中的有效性,对相应市场的弱式有效性做出评价。这些研究不需要流行的技术交易规则的先验知识,只需根据具体市场的特点设计遗传编程的基本函数和控制参数,这些研究使用训练数据得到最优技术交易规则,然后使用测试数据检验规则的有效性。将训练数据与测试数据分开,并考虑训练区间与测试区间的不同划分方式,避免数据探测问题,真正实现对技术交易规则有效性的样本外测试。此外,由于经过优化得到的最优技术交易规则一般较为复杂,常常是多个流行的技术交易规则的组合,因此基于遗传编程的市场有效性检验将更为可信^[19]。

根据对现有文献的梳理,目前基于遗传编程的中国金融市场有效性检验的研究较少。因此,本研究结合中国股票市场的特点,设计并实现基于遗传编程的技术交易规则优化方法,并通过最优技术交易规则在多个测试区间的超额收益的全面检验,对中国股票市场的弱式有效性做出评价。

3 遗传编程的优化方法

遗传编程和遗传算法一样,都是模拟生物进化原理的优化方法^[20,21],它们提供了求解复杂优化问题的通用框架,通过对解空间中的初始种群进行逐代优化,对问题的最优解展开并行搜索,并通过设计合理的适应度函数来影响解的搜索方向。两者的差别在于,遗传算法用线性结构表示问题的解,遗传编

程用树形结构表示问题的解。树形结构的编码方式能够很好地描述技术交易规则,树形结构中的叶子节点对应于输入数据,非叶子节点(根节点和分支节点)对应于函数模块,其子节点提供输入参数。图1给出移动平均线规则的树形结构表示,该移动平均线规则在当前价格超过30日平均价格时产生买入信号,反之则产生卖出信号。使用当前价格信息和历史价格信息,自底向上递归计算该树直至根节点,即可根据其函数返回值做出买卖决策。

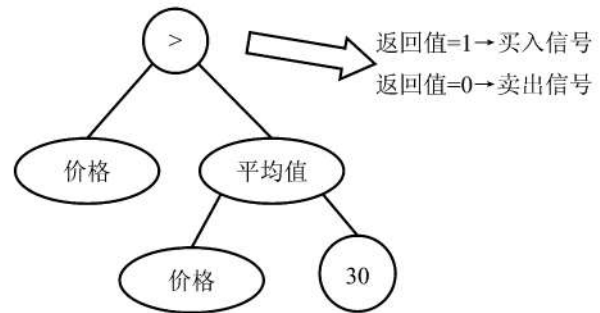


图1 30日移动平均线规则的树形结构表示
Figure 1 Tree Structure Corresponding to a 30-day Moving Average Rule

具体而言,遗传编程首先根据特定的问题,设定可供树形结构选用的函数模块和输入数据。在此基础上按设定的种群规模随机生成一组深度和形状任意的树,形成种群的初代。树的初始化中,节点的生成采用自顶向下的顺序,并根据一定的规则随机选择节点的函数或数据类型。同时,还需要根据优化目标设计适应度函数,用于计算每一棵树(问题的解)的适应度数值。

从父代种群向子代种群的进化通过复制、交叉和变异算子来实现。这里的复制是指将父代种群中的树直接拷贝进入子代种群;交叉是指交换父代种群中两棵树的子树(包括分支节点及其所有后代节点),并将新生成的树放入子代种群;变异是指随机改变父代种群中的某些树形结构,并将新生成的树放入子代种群^[8]。图2给出交叉算子的一个例子。首先,父代树1的节点M被随机选中,于是以节点M为根节点的子树m(图2中虚线标出部分)需要被替换。由于节点M的函数返回值为布尔型,接着在父代树2中也随机选择一个函数返回值类型为布尔型的节点N,用以节点N为根节点的子树n代替子树m,生成子代种群中的一棵新树。通常在设计复制、交叉、变异等进化规则时,会使父代种群中适应度数值较高的树更多地被选择去产生子代,使种群向更高适应度的方向进化。当子代种群中树的数量达到设定的种群规模时,一次进化完成。如此循环,直到进化次数达到预先设定的次数限制,或者适应度已达饱和时(如连续多代种群中的最优树具有相同的适应度数值),遗传编程将停止种群进化,此时种群中具有最高适应度数值的树就是近似的最优解。

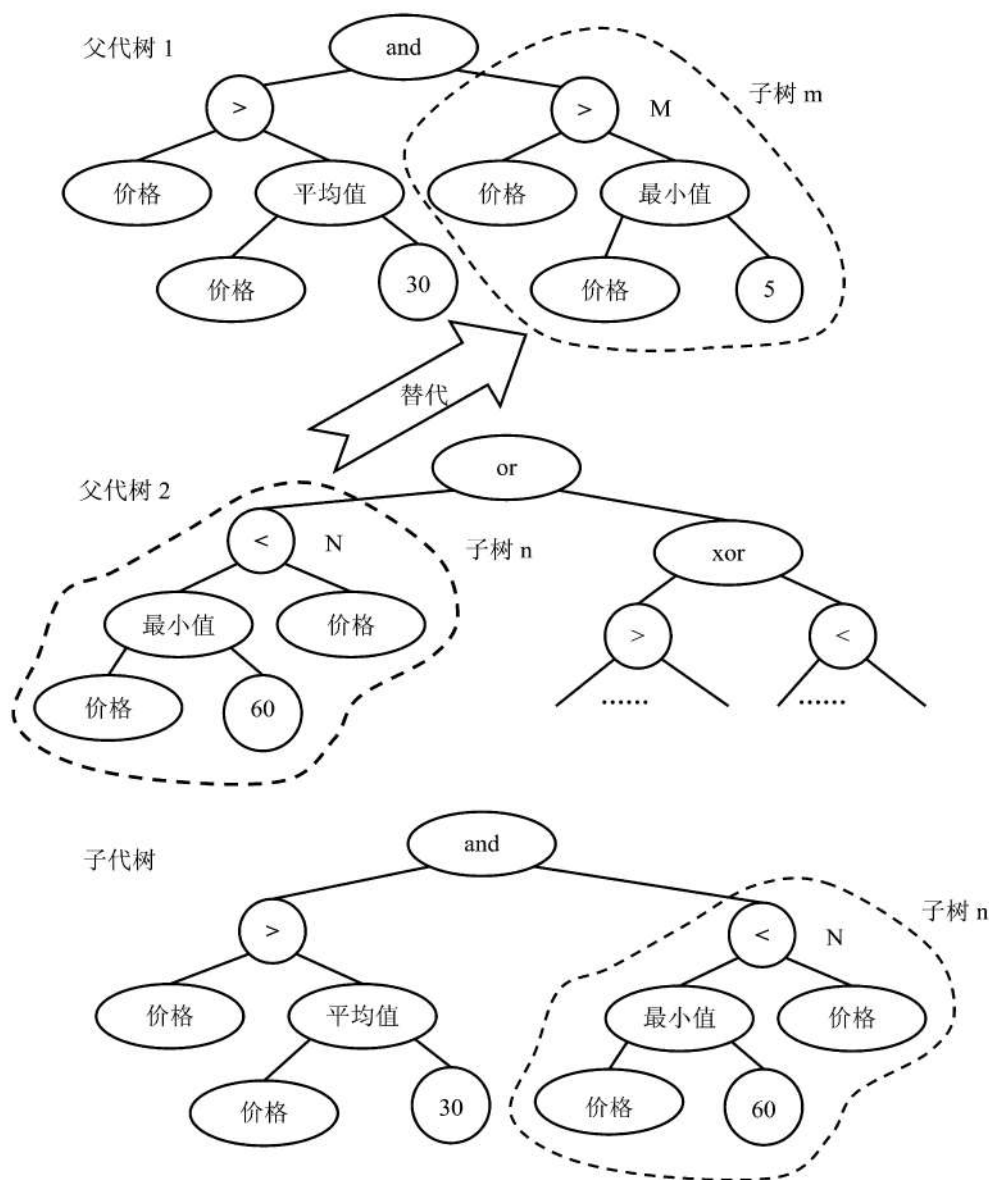


图 2 交叉算子示例

Figure 2 Illustration of Crossover Operator

4 基于遗传编程的中国股市技术交易规则研究

遗传编程作为一种较为通用的优化方法,根据不同的优化对象和优化目标,在函数模块以及输入数据、样本规模、适应度函数、进化规则等重要控制参数的设计上具有很大的灵活性。本研究将使用遗传编程寻找中国股票市场的最优技术交易规则,因此结合问题的具体特点进行如下设计。

4.1 数据类型、函数模块及基本规则

技术交易规则通过分析历史价格和历史成交量来做出买卖决策,为便于遗传编程捕捉价格变动大趋势外的细节规律,本研究将对数价格也作为输入数据。遗传编程的输入数据类型如表 1 所示。

遗传编程应用于股票市场技术交易规则研究时,通常采用基本算术运算符(加、减、乘、除)、逻辑

运算符(and、or、xor)、比较运算符(>、<),采用返回两个输入参数绝对差值的函数(norm)、返回指定窗长内数据序列的最大值(max)、最小值(min)和平均值(avg)的函数以及返回数据序列指定窗长的滞后值的函数(lag)等基本函数模块^[8,13]。

在此基础上,还引入另两个函数模块,即返回指定窗长内数据序列标准偏差的函数(std)和返回指定窗长内数据序列指数加权平均值的函数(eavg)。标准偏差函数可以反映价格与成交量序列的波动信息;指数加权平均值函数可以反映数据序列中的新、旧数据所具有的不同信息量,它对序列中各数据的加权随时间进行指数式衰减,越近期的数据加权越重,从而保证新数据对交易策略有更强的指导意义。数据序列 P 的指数加权平均值函数定义为

表1 遗传编程的输入数据类型
Table 1 Input Data Categories for Genetic Programming

数据类型	子类型	取值范围
序列数据	(1) 价格类: 开盘价 <i>OPR</i> , 收盘价 <i>CPR</i> , 最高价 <i>HPR</i> , 最低价 <i>LPR</i>	
	(2) 成交量类: 成交量 <i>VOL</i>	
	(3) 对数价格类: 对数开盘价 <i>LGOPR</i> , 对数收盘价 <i>LGCPR</i> , 对数最高价 <i>LGHPR</i> , 对数最低价 <i>LGLPR</i>	
布尔型数据		False (=0), True (=1)
窗宽数据		1~150 之间的整数
实数		(0,1]之间均匀分布的随机数

表2 遗传编程的函数模块及其说明
Table 2 Function Declarations for Genetic Programming

函数类型	函数名称	左参数	右参数	返回值	备注
序列运算	max, min, lag	(序列 + 时间)数据	窗长	(序列 + 时间)数据 序列实数	
	avg, std, eavg	(序列 + 时间)数据	窗长	序列实数	
算术运算	+, -, *, /	序列实数	(0,1]随机数	序列实数	
	norm	序列实数	序列实数	序列实数	左右参数必须为同一子类型序列数据
比较运算	<, >	序列实数	序列实数	布尔型	左右参数必须为同一子类型序列数据
逻辑运算	and, or, xor	布尔型	布尔型	布尔型	

$$\begin{cases} \text{eavg}(P_t, L) = P_t \cdot w + \text{eavg}(P_{t-1}, L) \cdot (1 - w), \\ w = \frac{2}{L+1}, t > L \\ \text{eavg}(P_t, L) = \text{avg}(P_t, t), t \leq L \end{cases} \quad (1)$$

其中, P_t 为时间标记在 t 日的数据序列 P , t 为普通平均值函数 (avg) 的窗长参数, L 为指数加权平均值函数的窗长参数, P_t 为数据序列 P 在 t 日的取值, w 为权重。

表2 给出本研究中使用的的基本函数模块。需要指出的是, 考虑到各子类型序列数据在数量级上有显著差别, 不同子类型 (如价格与成交量、对数价格与成交量) 之间的比较意义不大, 本研究规定比较运算符 (>, <) 和绝对差值计算函数的两个参数必须为同一子类型的序列数据, 这样可以有效避免一些无意义运算, 提高遗传编程的优化效率; 考虑到各子

类型序列数据与 (0, 1] 随机数的数量级悬殊, 彼此加减意义不大, 本研究规定序列实数加/减随机数的结果等于序列实数乘以 (1 加/减该随机数); 表2 中的 (序列 + 时间) 数据表示带有时间点标记的序列数据, 如 t 日的收盘价 CPR_t , 其时间点标记为 t ; 序列实数则表示序列数据的取值。

由于树形结构表示的技术交易规则最终需要给出买卖信号, 因此以布尔型返回值的函数作为树形结构的根节点, 返回值 1 为买入信号, 0 为卖出信号。由表2 可以看到, 可选作根节点的函数模块包括比较运算和逻辑运算。其中比较运算符的左右参数必须为同一子类型序列数据, 因此需要绑定序列类型, 也就是说该节点的所有序列数据后代节点只能采用同一序列类型 (即价格、成交量或对数价格 3 种子类型之一); 逻辑运算符的左右参数均为布尔型,

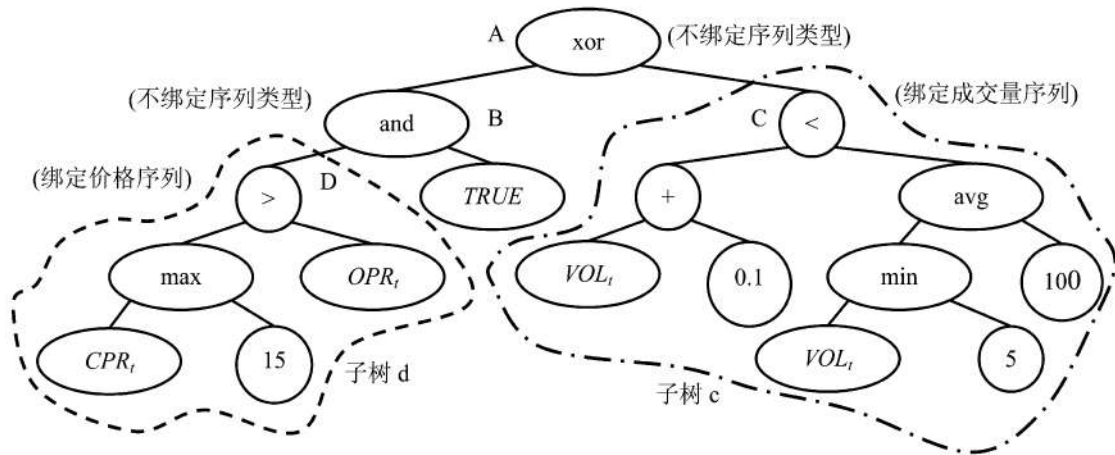


图 3 树形结构技术交易规则示例

Figure 3 Illustration of a Tree Structure Technical Trading Rule

因此不需绑定序列类型。图 3 给出树形结构技术交易规则的一个示例,根节点 A 随机选择为异或 (xor) 函数,无须绑定序列类型。其左子节点 B 随机选择为返回布尔值的与 (and) 函数,仍然不需绑定序列类型。B 的左子节点 D 随机选择为返回布尔值的 > 函数,并随机选择绑定价格类序列,因此子树 d 中只会出现价格类序列数据,如图 3 中的收盘价 CPR_t 和开盘价 OPR_t 。根节点 A 的右子节点 C 随机选择为返回布尔值的 < 函数,并随机选择绑定成交量类序列,因此子树 c 中只会出现成交量类序列数据 VOL_t 。

4.2 适应度数值的计算

已有文献中通常以检验最优技术交易规则是否较买入 - 持有策略有超额收益来评价股票市场的弱式有效性^[8],因此本研究也采用相同的适应度函数定义。对于长度为 T 日(日历日)的数据区间,假定初始时所有资产都处于现金状态,并规定 T 日结束时资产回到现金状态。技术交易规则在该数据区间的总收益 r 为

$$r = \sum_{t=1}^T r(t) \cdot I_b(t) + \sum_{t=1}^T rf(t) \cdot I_s(t) + n \cdot \log \frac{1-c}{1+c}$$

$$r(t) = \ln P(t) - \ln P(t-1)$$

$$rf(t) = \ln \frac{1 + Rfree(t)}{360} \tag{2}$$

其中, $r(t)$ 为 t 日连续复利的证券收益率; $I_b(t)$ 为资产处于股票状态的指标函数,若资产在 t 日处于股票状态则 $I_b(t)$ 取值为 1,若资产在 t 日处于现金状态则 $I_b(t)$ 取值为 0; $rf(t)$ 为 t 日连续复利的证券无风险利率; $I_s(t)$ 为资产处于现金状态的指标函数,若资产在 t 日处于现金状态则 $I_s(t)$ 取值为 1,若资产在 t 日处于股票状态则 $I_s(t)$ 取值为 0; n 为 T 日内买 - 卖操作的对数; c 为交易费用,在中国股票市场主要包括印花税、佣金、过户费等,本研究中将交易费用设为成交价格的 0.4%; $P(t)$ 为证券价格,本研究中使用日收盘价格; $Rfree(t)$ 为年化无风险利率,本研究中使用人民币活期存款年利率。

而买入 - 持有策略在该 T 日区间的总收益 $r_{buy-hold}$ 为

$$r_{buy-hold} = \sum_{t=1}^T r(t) + \log \frac{1-c}{1+c} \tag{3}$$

技术交易规则在该 T 日区间的适应度数值 f 即为该规则较买入 - 持有策略的年化超额收益,即

$$f = \frac{r - r_{buy-hold}}{T} \cdot 360 \tag{4}$$

4.3 进化规则

遗传编程的种群规模是一个重要的参数,规模过小可能不足以找到全局最优解,规模过大则将造成不必要的计算资源耗费。综合遗传编程现有文献的研究成果,并结合本研究设定的函数模块个数和输入数据种类考虑,将遗传编程的种群规模 K 设定为 2 000。

一代种群生成后,遗传编程将使用训练数据计算其中每一棵树的适应度数值,并按适应度数值由高到低的顺序将种群中的树排序。在此基础上,遗传编程将通过复制、交叉和变异,完成父代种群向子代种群的一次进化。本研究的进化规则设计如下。

(1) 复制

将父代种群中适应度数值排在前 5% 且适应度数值不小于零的树直接复制进入子代。

(2) 变异

尽管 Koza 指出变异对于遗传编程的重要性较低^[20],但研究者通过试验发现,如果不引入变异机制,逐代进化的种群很容易陷入局部最优。因此本研究引入简单的变异规则,规定子代种群中 5% 的成员为随机产生的新树,以保证每次进化种群中个体的多样性。

(3) 交叉

为了使父代种群中适应度数值较高的树更多地产生子代,树的适应度数值将控制其被选中参与交叉的概率,即适应度数值越高被选中的几率越大。针对部分父代树的适应度数值为负数的情况,本研

究对由大到小排序的适应度数值做如下处理,即

$$f_{\delta} = \begin{cases} 1.2 \cdot |f_K|, f_K < 0 \\ -0.8 f_K, f_K \geq 0 \end{cases}$$

$$nf_i = f_i + f_{\delta}$$

$$i = 1, 2, \dots, K \quad (5)$$

其中, f_{δ} 为适应度调整量; f_K 为父代种群中排在最末的树的适应度数值(最小值); nf_i 为第 i 棵树的适应度调整值, 决定该树被选中的概率; f_i 为父代种群中第 i 棵树的适应度数值。

本研究规定仅对非叶子节点的子树进行交叉, 同时交叉必须遵循两条原则。首先, 交叉节点的函数返回值类型必须相同。根据该原则, 函数模块可以分为表 3 中的 3 组, 各组内的函数可以互相替代。由于组 1 中函数的返回值也可作为序列实数, 它们可以替代返回序列实数的组 2 函数; 反向则不允许。其次, 交叉节点的函数返回值若为序列数据, 则必须对应于同一序列数据子类型, 即表 3 中组 1 和组 2 的函数仅能被操作同样序列数据子类型的函数替代。如图 3 中子树 c 的最小值 (min) 函数操作的是成交量类的序列数据, 则交叉时不能被操作价格类

序列数据的节点代替, 这一原则可以保证交叉产生的新树满足前面提出的序列类型绑定的要求。

表 3 交叉原则表

Table 3 Rule for Crossover Operator

组号	函数名
1	max, min, lag
2	avg, std, eavg, +, -, *, /, norm
3	<, >, xor, or, and

4.4 遗传编程的试验流程

遗传编程的一次优化过程包括初代种群的生成以及从父代种群向子代种群的多次进化, 图 4 给出本研究中遗传编程一次试验的流程。考虑到深度较大的交易规则树含有较大比例的冗余及无效分支, 因此本研究将树形结构的初始化深度限制在 9 层以内, 将交叉生成的子树深度限制在 20 层以内, 进化终止条件设置为种群的进化次数达到 20 次, 或者最优适应度数值在连续 10 代种群中保持不变。

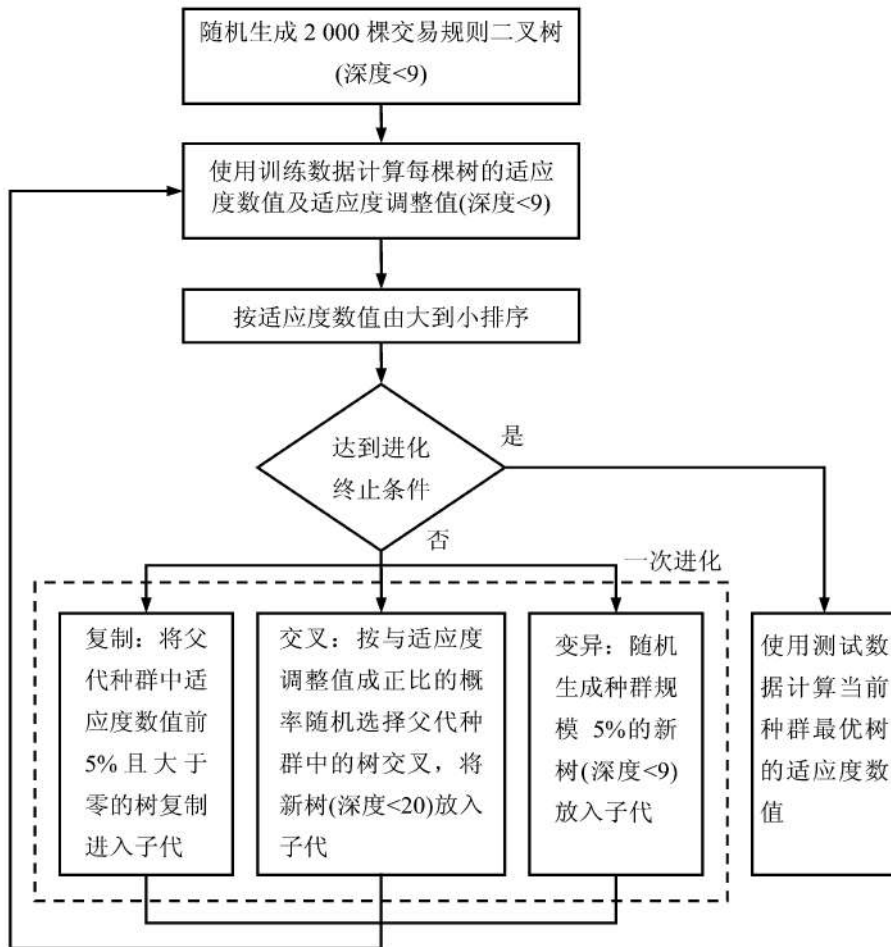


图 4 遗传编程试验的流程图

Figure 4 Flow Chart for One Trial of Genetic Programming

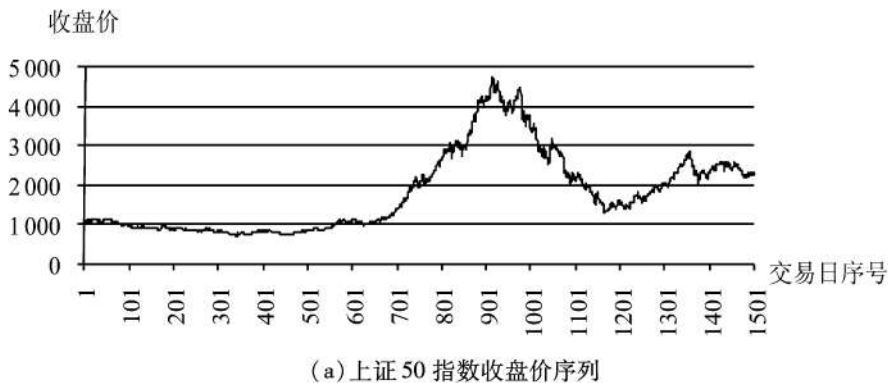


图5 上证50指数走势图

Figure 5 Historical Data for SSE 50 Index

5 实证

5.1 数据和试验

本研究使用2004年1月2日~2010年3月12日期间共计1503个交易日的上证50指数价格和成交量作为研究数据,图5给出上证50指数的收盘价序列和成交量序列在这段时期的走势。

遗传编程需要将研究数据分为训练区间和测试区间,前者用于优化技术交易规则,后者用于测试最优技术交易规则的样本外超额收益。为了防止固定训练区间和测试区间可能产生的数据探测问题,本研究将训练区间设计为长度为500的移动窗,将测试区间设计为长度为400的移动窗。考虑到序列操作函数的窗长可能取1~150之间的整数值,而且树形结构交易规则中可能出现多个序列操作函数,训练窗-测试窗的起始组合设为300~800~1200。规定训练窗和测试窗移动的步进值为50,则研究数据共可分为7组训练窗-测试窗组合。对于每一组训练窗-测试窗组合,本研究都进行100次独立的遗传编程试验,并通过对这700次独立试验结果的统计分析检验中国股票市场的弱式有效性。

为了检验优化方法的稳健性,本研究采用长度为600和400的移动窗分别作为训练区间和测试区间,以300~900~1300为第一组,将研究数据分成步进值为50的5组训练窗-测试窗组合。对每一组训练窗-测试窗组合同样进行100次独立的遗传编程

试验,并对这500次独立试验的结果进行统计分析。

5.2 结果和分析

图6显示了训练区间和测试区间分别为500和400移动窗的700次独立遗传编程试验,在各自的训练区间所得的最优技术交易规则应用于相应测试区间时的年化超额收益以及买-卖交易对数。大部分的最优技术交易规则在测试区间的年化超额收益都大于零,而它们的平均值和中位数分别为11.699%和8.038%。最优技术交易规则在测试区间(400个交易日)的交易频率总体来说比较低,它们的买-卖交易次数的平均值和中位数分别为8对和5对。

表4给出对这700次独立遗传编程试验得到的最优技术交易规则的样本外超额收益的统计分析。在这700次独立试验中,样本外超额收益为正值的共计487次,占69.6%;样本外超额收益非负的共计547次,占78.1%。每一组训练窗-测试窗组合的100次独立试验均有半数以上获得正的样本外超额收益,有75%左右获得非负的样本外超额收益。每一组的100次独立试验在相应测试区间内超额收益的平均值和中位数都是正值。此外,从表4中计算的t统计量可以看出,每一组训练窗-测试窗组合下,100次独立试验的样本外超额收益的平均值都显著大于0。可见,技术交易规则在考虑了交易费用后,仍然能够较买入-持有策略产生统计显著的样本外超额收益。因此,中国股票市场尚未达到弱式有效。

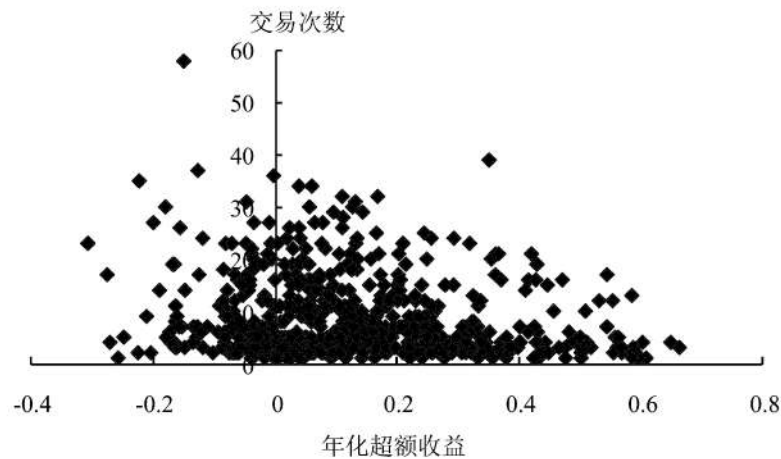


图6 最优技术交易规则(700次遗传编程试验)在测试区间的年化超额收益与交易次数
Figure 6 Annual Excess Return and Number of Buy-sell Pairs during Testing Period for Best Technical Trading Rules Generated by 700 Genetic Programming Trials

表4 基于遗传编程的最优技术交易规则(训练窗长500)在测试区间的年化超额收益统计(f 为年化超额收益)

Table 4 Annual Excess Return Statistics of Genetic Programming Generated Best Technical Trading Rules (Training Window Size 500) in Testing Period (f Is Annual Excess Return)

训练区间	测试区间	试验数	f 为正	f 非负	f 平均值 (%)	f 中位数 (%)	f 标准差 (%)	t 统计量 ($f > 0$)	平均买卖对数
300~800	800~1 200	100	72	77	14.226	7.911	19.368	7.345	9
350~850	850~1 250	100	52	67	7.137	0.586	17.631	4.048	6
400~900	900~1 300	100	64	81	13.592	9.049	17.714	7.673	5
450~950	950~1 350	100	63	81	14.315	6.121	21.156	6.766	6
500~1 000	1 000~1 400	100	84	87	16.710	12.427	16.159	10.341	11
550~1 050	1 050~1 450	100	77	78	9.464	10.091	13.792	6.862	10
600~1 100	1 100~1 500	100	75	76	6.449	6.873	12.689	5.082	9
总计		700	487	547	11.699				8

表5给出训练区间和测试区间分别为600和400移动窗的500次独立遗传编程试验结果的统计分析。在这500次独立试验中,样本外超额收益为正值的共计369次,占73.8%;样本外超额收益非负的共计394次,占78.8%,与表4中700次独立试验的统计结果类似。从表5中的 t 统计量可以看出,每一组训练窗-测试窗组合下,100次独立试验的样本外超额收益的平均值都显著大于0。因此,这500次独立遗传编程试验的结果仍然支持中国股票市场尚未达到弱式有效的结论。

此外,由于对于上述任一组训练窗-测试窗组合,100次独立遗传编程试验的样本外超额收益的平

均值都为正值,可以设计基于遗传编程的程序化交易系统。该系统在设定的训练区间内进行100次独立遗传编程试验,然后在相应的测试区间将得到的100个最优技术交易规则分别运用于资金被分成的100等分之一。100个最优技术交易规则的共同运用,将确保该程序化交易系统在相应测试区间内获得正的样本外超额收益,持续地战胜市场。

为了对该遗传编程优化方法产生的最优技术交易规则有更具体的认识,对上述12组训练窗-测试窗组合共计1 200次独立遗传编程试验产生的最优技术交易规则树形结构进行研究。首先,遗传编程产生的最优技术交易规则大多具有相对复杂的嵌套

表 5 基于遗传编程的最优技术交易规则(训练窗长 600)
在测试区间的年化超额收益统计(f 为年化超额收益)

Table 5 Annual Excess Return Statistics of Genetic Programming Generated Best Technical Trading Rules
(Training Window Size 600) in Testing Period (f Is Annual Excess Return)

训练区间	测试区间	试验数	f 为正	f 非负	f 平均值 (%)	f 中位数 (%)	f 标准差 (%)	t 统计量 ($f > 0$)	平均买卖对数
300 ~ 900	900 ~ 1 300	100	62	71	9.810	5.759	17.577	5.581	8
350 ~ 950	950 ~ 1 350	100	68	77	15.612	12.506	21.084	7.405	5
400 ~ 1 000	1 000 ~ 1 400	100	91	92	18.725	16.134	14.797	12.654	10
450 ~ 1 050	1 050 ~ 1 450	100	76	82	10.042	7.509	13.198	7.609	7
500 ~ 1 100	1 100 ~ 1 500	100	72	72	6.972	6.753	11.845	5.886	9
总计		500	369	394	12.232				7.8

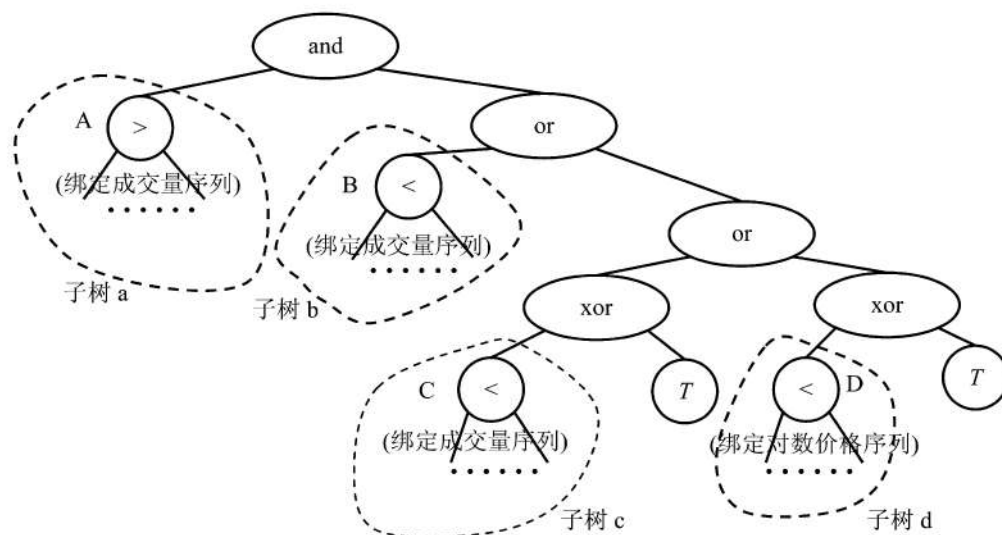


图 7 遗传编程最优技术交易规则示例

Figure 7 Example of Genetic Programming Generated Best Technical Trading Rule

结构,其中一些树形结构可以通过消除其中的冗余规则加以简化,如消除多层 lag 函数的嵌套等,这样可以提高程序化交易系统的计算效率。另外需要指出的是,成交量信息在这些最优技术交易规则中得到充分的运用,所观察的 1 200 个最优技术交易规则树形结构中有 1 132 个树形结构存在至少 1 棵绑定成交量序列的子树,占 94.3%。其中一些树形结构则主要依靠成交量信息做出买卖决策,例如图 7 中训练窗 - 测试窗组合为 500 ~ 1 000 ~ 1 400 时具有最大样本外年化超额收益的最优技术交易规则,其树形结构共有 9 层,53 个节点,包括 4 棵绑定序列类型的子树,其中 3 棵子树(子树 a、子树 b、子树 c)均绑定成交量序列。这一现象与成交量信息在中国股市技

术分析者中得到广泛运用相符合(如中国股谚中对“量”多有提及),也表明纳入成交量信息对于中国股票市场有效性检验的重要性。

6 结论

股票市场的有效性是众多经济金融理论的基本假设,技术交易规则的有效性因其直接与有效市场假说相矛盾而成为学术界关注的焦点。本研究设计实现了模仿生物遗传学和自然选择机理的遗传编程优化方法,用上证 50 指数进行实证研究,由基本函数模块构建出在现实的交易费用下较买入 - 持有策略有显著超额收益的技术交易规则,得出中国股票市场尚未达到弱式有效的结论。该遗传编程优化方

法不依赖于流行技术交易规则的先验知识,有效避免了现有研究使用流行的技术交易规则检验市场有效性时存在的数据探测问题。采用可移动的训练窗-测试窗将研究数据划分为前后相继的训练区间和测试区间,通过不同的窗长组合以及训练窗-测试窗的步进移动,进一步消除数据探测的可能性。

本研究是遗传编程应用于中国股票市场技术交易的初步探索,基于遗传编程的中国股市技术交易规则研究还有很多值得深入的方向。如改进遗传编程的进化规则以控制树形结构技术交易规则的冗余度;在适应度函数中结合风险惩罚机制,定制适用于不同风险态度交易者的最优技术交易规则;调整树形结构的根节点函数模块,使技术交易规则适用于中国新近开展的融资融券交易,等等。而本研究在设计函数模块时引入的序列类型绑定功能,使本研究的遗传编程优化方法可以灵活地扩展输入数据类型,如引入反映国际指数每日变化的序列、引入对数成交量序列等,以寻求最优技术交易规则性能的进一步提升。在此基础上,还可以用具备遗传编程这一机器学习技术的异质个体来组建人工股票市场,并研究市场的演化机理。我们将在后续研究中对此进行探索。

参考文献:

- [1] 魏玉根. 技术交易系统与我国股票市场有效性的实证分析[J]. 经济科学, 2000(2):56-63.
Wei Y G. Technical Trading Systems and Empirical Tests of Chinese Stock Market's Efficiency[J]. Economic Science, 2000(2):56-63. (in Chinese)
- [2] 孙碧波. 移动平均线有用吗?——基于上证指数的实证研究[J]. 数量经济技术经济研究, 2005, 22(2):149-156.
Sun B B. Are Moving Average Rules Profitable? Evidence from Shanghai Composite Index[J]. The Journal of Quantitative & Technical Economics, 2005, 22(2):149-156. (in Chinese)
- [3] 曾劲松. 技术分析与中国股票市场有效性[J]. 财经问题研究, 2005(8):27-30.
Zeng J S. Technical Analysis and Chinese Stock Market's Efficiency[J]. Research on Financial and Economic Issues, 2005(8):27-30. (in Chinese)
- [4] 王志刚, 曾勇, 李平. 技术交易规则预测能力与收益率动态过程——基于 Bootstrap 方法的实证研究[J]. 数量经济技术经济研究, 2007, 24(9):122-133.
Wang Z G, Zeng Y, Li P. Predictability of Technical Trading Rules and Return Dynamics: An Empirical Research Based on Bootstrap Method[J]. The Journal of Quantitative & Technical Economics, 2007, 24(9):122-133. (in Chinese)
- [5] 王志刚, 曾勇, 李平. 中国股票市场技术分析非线性预测能力的实证检验[J]. 管理工程学报, 2009, 23(1):149-153.
Wang Z G, Zeng Y, Li P. An Empirical Test on the Non-linear Predictability of Technical Analysis on China Stock Market[J]. Journal of Industrial Engineering and Engineering Management, 2009, 23(1):149-153. (in Chinese)
- [6] 邓杰, 唐国兴. 中国股票市场技术交易规则有效性的实证研究[J]. 华东经济管理, 2009, 23(5):135-140.
Deng J, Tang G X. Empirical Research on Technology Trading Rules Effectiveness in Chinese Stock Market[J]. East China Economic Management, 2009, 23(5):135-140. (in Chinese)
- [7] 陈浪南, 王艺明. 技术交易规则与超常收益研究[J]. 经济研究, 2001(12):73-81.
Chen L N, Wang Y M. Research on Technical Trading Rules and Excess Return[J]. Economic Research Journal, 2001(12):73-81. (in Chinese)
- [8] Allen F, Karjalainen R. Using Genetic Algorithms to Find Technical Trading Rules[J]. Journal of Financial Economics, 1999, 51(2):245-271.
- [9] Kaboudan M. Genetic Programming Prediction of Stock Prices[J]. Computational Economics, 2000, 16(3):207-236.
- [10] Rajabioun R, Rahimi-Kian A. A Genetic Programming Based Stock Price Predictor together with Mean-variance based Sell/buy Actions[C] // Proceedings of the World Congress on Engineering. London, 2008.
- [11] Álvarez Díaz M. Speculative Strategies in the Foreign Exchange Market Based on Genetic Programming Predictions[J]. Applied Financial Economics, 2010, 20(6):465-476.
- [12] Neely C, Weller P, Dittmar R. Is Technical Analysis in the Foreign Exchange Market Profitable? A Genetic Programming Approach[J]. Journal of Financial and Quantitative Analysis, 1997, 32(4):405-426.
- [13] Potvin J Y, Soriano P, Vallée M. Generating Trading Rules on the Stock Markets with Genetic Programming[J]. Computers & Operations Research, 2004, 31(7):1033-1047.
- [14] Wilson G, Banzhaf W. Algorithmic Trading with Developmental and Linear Genetic Programming[C] // Riolo R, O'Reilly U M, McConaghy T. Genetic Programming Theory and Practice VII. New York: Springer, 2010:119-134.
- [15] Yu T, Chen S H, Kuo T W. Discovering Financial Technical Trading Rules Using Genetic Programming with Lambda Abstraction[C] // O'Reilly U M, Yu T, Riolo R, Worzel B. Genetic Programming Theory and Practice II. New York: Springer, 2005:11-30.

- [16] How J, Ling M, Verhoeven P. Does Size Matter? A Genetic Programming Approach to Technical Trading [J]. *Quantitative Finance*, 2010, 10(2): 131–140.
- [17] Wang J. Trading and Hedging in S&P 500 Spot and Futures Markets Using Genetic Programming [J]. *The Journal of Futures Markets*, 2000, 20(10): 911–942.
- [18] Roberts M. Technical Analysis and Genetic Programming: Constructing and Testing a Commodity Portfolio [J]. *The Journal of Futures Markets*, 2005, 25(7): 643–660.
- [19] Hsu P H, Kuan C M. Reexamining the Profitability of Technical Analysis with Data Snooping Checks [J]. *Journal of Financial Econometrics*, 2005, 3(4): 606–628.
- [20] Koza J R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection* [M]. Cambridge: The MIT Press, 1992.
- [21] Langdon W B, Poli R, McPhee N F, Koza J R. *Genetic Programming: An Introduction and Tutorial, with a Survey of Techniques and Applications* [C] // Fulcher J, Jain L C. *Computational Intelligence: A Compendium*. Berlin: Springer, 2008: 927–1028.

Using Genetic Programming to Find Technical Trading Rules for SSE 50 Index

QU Hui

School of Management and Engineering, Nanjing University, Nanjing 210093, China

Abstract: To avoid data snooping bias induced by the common approach of testing market efficiency through popular technical trading rules, genetic programming is proposed to find optimal technical trading rules. The first generation of technical trading rules is randomly generated tree structures of basic functions, optimized by genetic programming. Data are divided into training period and testing period using moving windows. The technical trading rules get optimized generation by generation using the training data, while the subsequent testing data are used to test the optimization result's out-of-sample performance. Using daily price and volume data for the SSE 50 index from January 2, 2004 to March 12, 2010, we find out that genetic programming generated best technical trading rules have statistically significant out-of-sample excess return compared with buy-and-hold considering realistic transaction cost. Therefore, we conclude that Chinese stock market has not achieved weak-form efficiency.

Keywords: technical trading; market efficiency; genetic programming; optimization

Received Date: April 20th, 2010 **Accepted Date:** September 6th, 2010

Funded Project: Supported by the National Natural Science Foundation of China(70932003)

Biography: Dr. QU Hui, a Jiangsu Nantong native (1981 –), graduated from Cornell University and is an associate professor in the School of Management and Engineering at Nanjing University. Her research interests include computational finance, etc. E-mail: linda59qu@nju.edu.cn

□