

國立政治大學資訊管理研究所

碩士學位論文

指導教授：劉文卿 博士

基於 Hadoop 雲端運算架構建立  
策略交易與回測模擬平台



研究生：黃柏翰 撰

中華民國一百零一年三月

# 摘要

為了讓一般的投資大眾能享有智慧型、系統化的策略交易環境，本研究計畫發展一個可供大量使用者共用、並且容易上手的策略交易平台。為了達到這個目的，此平台必須擁有快速且大量的運算能力，雲端運算所提供之大量且可擴充的運算能力，使之成為最適宜的平台。為滿足不同使用者不同的投資偏好，此平台提供多項常用之技術指標與 K 線型態辨識功能讓使用者利用基因演算法產生符合其偏好的交易策略。在策略產生之後，使用者可以在平台上檢視交易策略在不同商品、不同時間區間上的表現，並從最後的策略回測報告中加以評估，挑選出獲利能力、波動程度與交易頻率都符合需求的交易策略。

**關鍵詞：**雲端運算、基因演算法、演算法交易、技術分析、K 線型態

# 目錄

摘要.....	1
目錄.....	2
圖目錄.....	3
表目錄.....	4
第一章 緒論.....	5
第一節 研究背景與目的.....	5
第二章 文獻探討.....	7
第一節 雲端運算.....	7
第二節 Apache Hadoop.....	9
第三節 HBase.....	12
第四節 技術分析.....	13
第五節 基因演算法.....	15
第三章 研究流程與系統架構.....	17
第一節 研究流程.....	18
第二節 系統架構.....	20
一、市場狀態計算模組.....	21
二、基因演算法模組.....	23
三、前端網頁介面.....	29
第四章 實驗數據分析.....	31
第一節 模擬速度測試與分析.....	31
第二節 策略回測實驗數據.....	34
第三節 基因演算法參數設定.....	40
第五章 結論.....	42
第一節 研究結論.....	42
第二節 未來展望.....	43
參考文獻.....	44
附錄一.....	46
附錄二.....	59

# 圖目錄

圖 1、Jobtracker 與 tasktracker 示意圖.....	10
圖 2、HDFS 架構圖.....	11
圖 3、HBase 架構圖.....	12
圖 4、基因演算法運作流程圖.....	15
圖 5、本研究之研究流程圖.....	18
圖 6、系統架構圖.....	20
圖 7、市場狀態計算模組架構圖.....	21
圖 8、SRN Model 示意圖.....	22
圖 9、染色體編碼示意圖.....	24
圖 10、整組基因交配示意圖.....	25
圖 11、部分基因交配示意圖.....	25
圖 12、基因位置突變示意圖.....	26
圖 13、基因參數突變示意圖.....	26
圖 14、基因演算法模組系統架構圖-Horizontal Design.....	27
圖 15、基因演算法模組系統架構圖-Vertical Design.....	28
圖 16、染色體長度=2 時，不同染色體族群數量在不同演化代數之獲利率折線圖.....	35
圖 17、染色體長度=4 時，不同染色體族群數量在不同演化代數之獲利率折線圖.....	36
圖 18、染色體長度=8 時，不同染色體族群數量在不同演化代數之獲利率折線圖.....	37
圖 19、染色體長度=16 時，不同染色體族群數量在不同演化代數之獲利率折線圖.....	38
圖 20、染色體長度=32 時，不同染色體族群數量在不同演化代數之獲利率折線圖.....	39

# 表目錄

表 1、模擬演化時間表(叢集模式-1 個 Jobtracker 和 3 個 Tasktracker).....	31
表 2、模擬演化時間表(叢集模式-1 個 Jobtracker 和 6 個 Tasktracker).....	32
表 3、模擬演化時間表(單機虛擬分散模式).....	33
表 4、染色體長度=2 時，不同染色體族群數量在不同演化代數之獲利率 .....	35
表 5、染色體長度=4 時，不同染色體族群數量在不同演化代數之獲利率 .....	36
表 6、染色體長度=8 時，不同染色體族群數量在不同演化代數之獲利率 .....	37
表 7、染色體長度=16 時，不同染色體族群數量在不同演化代數之獲利率 .....	38
表 8、染色體長度=32 時，不同染色體族群數量在不同演化代數之獲利率 .....	39



# 第一章 緒論

## 第一節 研究背景與目的

自民國 92 年起，台灣多次出現「實質薪資負成長」的情形，意即薪資成長的幅度跟不上通貨膨脹、物價上漲的程度。除了民生用品、原物料價格等顯著上漲外，其中又以房價的上漲最為可觀，根據國內房仲業者信義房屋與美國西維吉尼亞大學合作編制的信義房價指數計算，近三年台灣地區房價指數成長約為 20%，其中台北市房價指數成長竟然高達 50.2%。在這樣的時空背景之下，如何妥善的投資理財逐漸成為人們所關注的重要課題，就算不見得能透過投資致富，也要試著藉由投資來避免通貨膨脹一點一滴地侵蝕掉自己的資產。

根據 Aite Group 2006 年的調查 (The Economist, 2006) 顯示，歐洲及美國股票金融市場中有 30% 的成交量，是經由自動化交易或演算法交易完成的，估計在 2010 年會持續成長到 50%。另外，在經濟日報 2009 年 9 月的報導中 (賴育連，2009)，俄羅斯的 TS Support 執行長丹尼斯葛洛帕 (Dennis Globa) 表示，藉由演算法交易證券所占的比例在過去 3 年，由 30% 成長到 70%，可見自動化交易在金融市場上的重要性與日俱增。

這些使用演算法自動化交易的市場參與者多半為大型投資機構，而一般中小型投資人多半只能依靠本身的觀察與判斷來進行投資。根據朱成志(2004)的研究指出，影響台灣地區一般投資人投資決策最主要的前三大變數依序為：財經媒體 (包含新聞雜誌、電視節目)、投資標的基本面(財務報表分析數據)與技術指標(KD、RSI、MACD 等技術指標所呈現之買進或是賣出訊號)。由此可見，台灣地區一般投資人的投資決策多數仍是受到傳播媒體的影響，而非進行客觀的系統化交易。但是台灣的傳播媒體素質良莠不齊，許多傳播媒體缺乏深度探討的能力，時常大漲就翻出利多消息報導，而大跌時就找出利空消息報導，更有甚者會配合炒作大

戶發表所謂「出貨文」，讓許多盲目追隨的投資大眾暴露在高度風險之中。

為了使演算法交易普及，讓一般投資者進行交易有所依歸，本研究計畫發展出一套包含篩選金融市場交易策略及模擬交易回測策略績效功能之策略交易平台。為滿足大量使用者的運算需求，本平台包括以下幾點特性：

1. 為了使本平台擁有良好的分散式運算機制及可擴充性，並提供高速處理海量資料的運算能力，本平台將採用 Apache Hadoop 雲端運算架構。
2. 包含各項技術分析指標計算與 K 線型態辨識能力，提供使用者多樣化的交易策略。
3. 提供基因演算法模擬環境與模擬交易回測平台，讓使用者評估各式交易策略績效，進而建構自己所需之交易模型。



## 第二章 文獻探討

### 第一節 雲端運算

根據美國國家標準局(NIST)的定義：雲端運算是運用無所不在、便利、隨需應變的網路，共享廣大的運算資源，如網路、伺服器、儲存、應用程式以及服務等，可透過最少的管理工作及服務供應者互動，快速提供各項服務。簡言之，雲端運算的精隨就是運用網路共享廣大的資源做分散式的儲存與運算。

除了基本的定義，NIST 就雲端運算服務模式的不同，將目前常見的雲端運算服務分為三大類，依序為軟體即服務 (Software as a Service, SaaS)、平台即服務 (Platform as a Service, PaaS)、架構即服務 (Infrastructure as a Service, IaaS)。

#### 一、軟體即服務 (SaaS)：

消費者使用應用程式，但並不掌控作業系統、硬體或運作的網路基礎架構。即代表透過網路提供商業應用軟體的一種新興服務模式，對於使用者來說，SaaS 讓軟體取得的成本與使用方式都跟以往不同，過去使用商業應用軟體必須先購買使用權，在 SaaS 模式下，軟體是存放在提供者端，使用者透過網路使用應用軟體。

#### 二、平台即服務 (PaaS)：

消費者使用主機操作應用程式。消費者掌控運作應用程式的環境 (也擁有主機部分掌控權)，但並不掌控作業系統、硬體或運作的網路基礎架構。平台通常是應用程式基礎架構。就是提供平台的服務，讓客戶將開發的應用程式很容易部署到雲端如 Google App Engine 或微軟的 Azure 平台，給開發人員更大的方便與彈性。

### 三、基礎架構即服務 (IaaS)：

消費者使用「基礎運算資源」，如處理能力、儲存空間、網絡元件或中介軟體。消費者能掌控作業系統、儲存空間、已部署的應用程式及網絡元件（如防火牆、負載平衡器等），但並不掌控雲端基礎架構。

本研究所提出之策略交易與回測模擬平台即為軟體即服務(SaaS)的一種，為求同一時間能夠供給大量使用者使用，並進行大量的市場交易資料運算，具有高度彈性、可擴充性且能分散式處理大量資料的雲端運算架構將為本平台最佳解決方案。



## 第二節 Apache Hadoop

Hadoop 是 Apache 軟體基金會的一個開放原始碼專案，它是由 Apache Lucene 的創立者 Doug Cutting 所建立的，由 Java 寫成，主要用途為建構大量資料的分散式儲存與運算環境。Hadoop 專案包含了兩個最主要的元素，Hadoop Distributed File System(HDFS)與 MapReduce。

Hadoop 最初是 Apache Lucene 這個專案中的搜尋引擎—Nutch 的一部分。起初的 Nutch 實現了一套可以抓取網頁並搜尋資料的系統，但是開發人員很快地發現它能處理的網頁數量是有極限的，在當時的架構之下，無法擴充到處理數十億以上的網頁。之後，Google 發表了有關於 Google File System 的論文，這篇論文中所提出的架構可以解決抓取大量網頁和建立索引時所產生的大量檔案，這對 Nutch 的開發團隊來說很有幫助。於是，Nutch 的開發團隊就開始將這個觀念在 Nutch 專案中實作，並將其稱為 Nutch Distributed File System(NDFS)，也就是 Hadoop HDFS 的前身。

2004 年，Google 又發表了一篇介紹 MapReduce 概念的論文。MapReduce 是一種用於處理資料的分散式程式框架，用平行運算的方式，將大量需要運算的資料，切割成許多區塊，並將這些區塊分給叢集中的每個運算節點再同時對這些區塊分別作運算，最後再將運算的結果結合起來，產生出最終的運算結果。Nutch 的開發團隊同樣地將這個概念在 Nutch 專案中實作出了開放原始碼的版本。

在 Nutch 計畫中的 NDFS 與 MapReduce 原本是設計來處理搜尋領域的相關運算，但人們發現分散式儲存與運算的架構在別的領域上也有優異的表現，它們的應用慢慢地超出了原本計畫使用的搜尋領域，於是開發人員於 2006 年將 NDFS 與 MapReduce 從 Nutch 計畫中抽出，成為一個獨立的 Lucene 子計畫，並將其命名為 Hadoop。在 2008 年 1 月，Hadoop 又從 Lucene 計畫中獨立出來，成為 Apache 最頂級的計畫(top level project)之一。以下將會依序介紹 MapReduce 與 HDFS 的架構：

## 一、MapReduce：

MapReduce 將運算程序分為兩個階段 Map 和 Reduce 階段。每個階段的輸入及輸出都採用 <key, value>pair 的方式。在 Hadoop 中，一個工作(job)是客戶端要執行的一個工作單位，它包含了輸入資料、MapReduce 程式與組態資訊。而一個工作又會被分為數個任務(task)。

在 Hadoop 叢集中，節點分為兩種不同身分，一種是 jobtracker，另一種是 tasktracker。Jobtracker 在一個叢集中只有一個，它對所有的 tasktracker 進行任務的分配與排程，tasktracker 會執行任務並且將執行情況報告給 jobtracker，如果任務執行失敗，jobtracker 會將失敗的任務分派給其他的 tasktracker 執行。

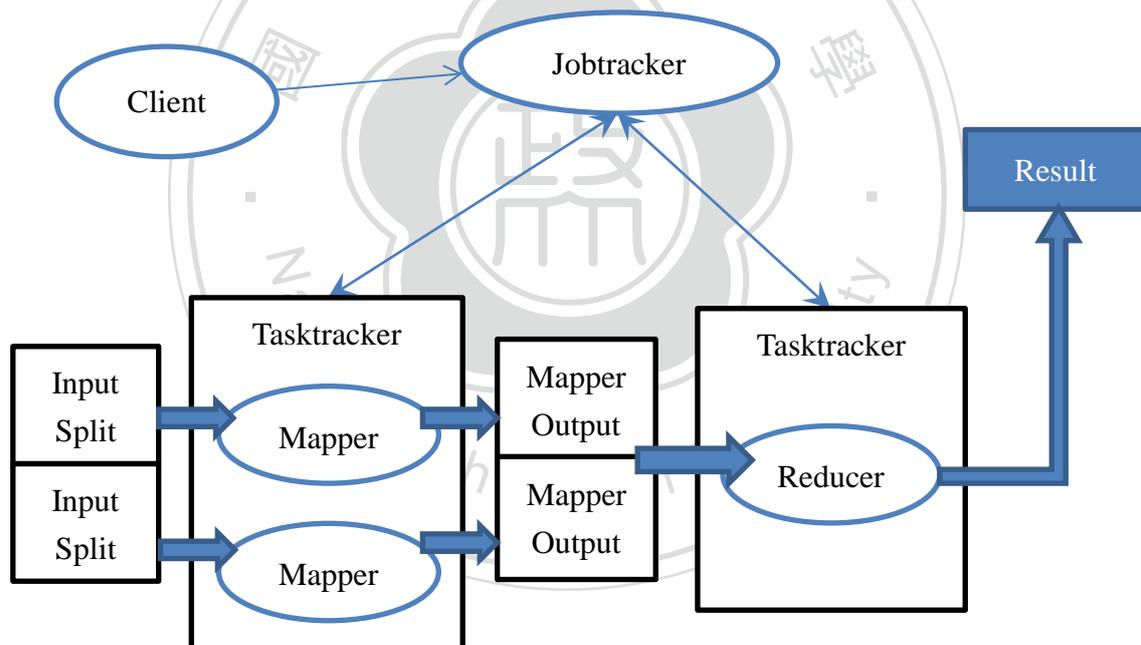


圖 1、Jobtracker 與 tasktracker 示意圖

(資料來源：本研究整理)

任務的類型分為兩種，map 任務與 reduce 任務。Hadoop 會將輸入資料分割成固定大小後再開始做 MapReduce，map 任務就是每個分割後的輸入資料區塊都會執行的運算過程，交給 mapper 執行。Mapper 運算完之後的輸出一樣會用 <key, value>pair 的形式送給處理 reduce 任務的 Reducer 作為輸入。最後 reducer 再依照開發人員撰寫的程式將資料結合化簡，產生最後的輸出。

## 二、HDFS：

一個 HDFS 叢集裡面會有兩種類型的節點，一個 namenode 與一群 datanode。Namenode 管理著整個檔案系統的命名空間，他會維護檔案系統的結構和所有檔案與目錄的 metadata。而 datanode 則是實際儲存檔案的地方，一個檔案在 HDFS 中可能會被分為一個或多個資料區塊，這些區塊會儲存在一組 datanode 上。為了提升容錯能力而達到高可用性(availability)，資料區塊通常會有多個複本儲存在不同的 datanode 上，namenode 也負責進行資料區塊的建立、刪除與複製。

當客戶端應用程式要讀寫資料時，會先向 namenode 進行 metadata 操作，在 namenode 的安排下，客戶端應用程式會知道該直接向哪一個 datanode 存取資料。實際在進行資料讀寫時，資料流並不會經過 namenode，而是由 datanode 直接處理客戶端應用程式的讀寫請求。

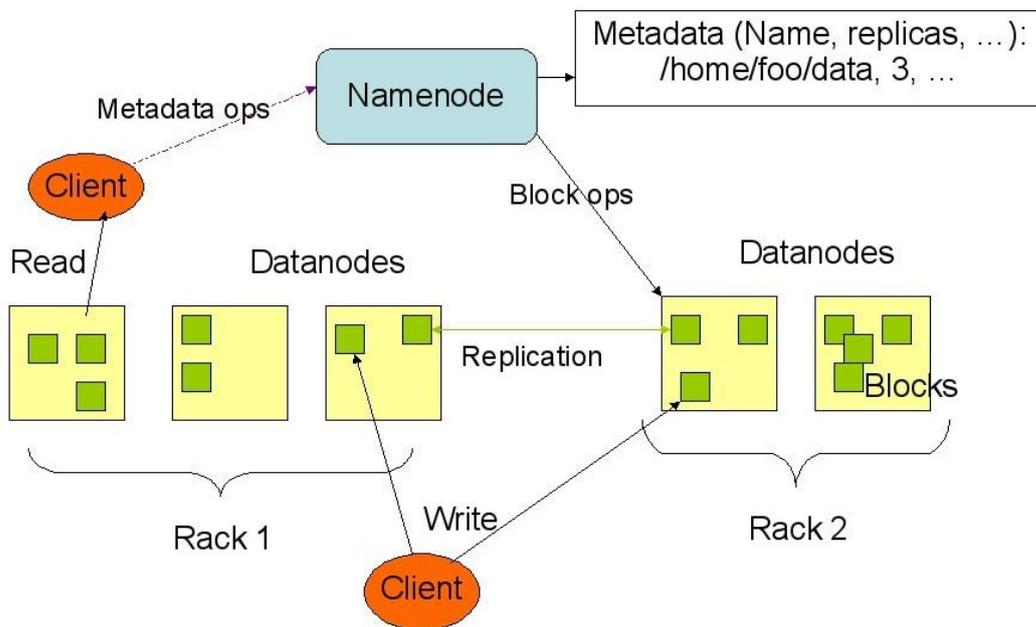


圖 2、HDFS 架構圖

(資料來源：apache.org)

### 第三節 HBase

HBase 是一個非關聯式、欄導向 (column-oriented) 的分散式資料庫，建構於 HDFS 之上。當開發者需要即時讀寫或是隨機存取大量資料時，就適合使用 HBase。HBase 專案起源於 2006 年，由 Chad Walters 與 Jim Kellerman 創立。它參考了 Google 發表的 Bigtable 概念，並加以實作。HBase 原本是 Hadoop 專案的子計畫，到了 2010 年 5 月，HBase 正式成為 Apache 的首層專案。

在系統架構上，HBase 如同 MapReduce 由 jobtracker 和 tasktracker 組成、HDFS 由 namenode 和 datanode 組成一般，HBase 也是採用 master/slave 架構，由 HBase master 和多個 regionserver 所組成。在 HBase 中，區域 (region) 代表的是一個資料表的水平分割。HBase master 負責分配區域給已經註冊的 regionserver、監控 regionserver 的運作情形、維護 regionserver 的負載平衡以及資料表結構管理等工作。Regionserver 負責儲存零到多個區域並處理客戶端的讀寫操作，同時 regionserver 也會定期向 HBase master 回報自己的運作情形。

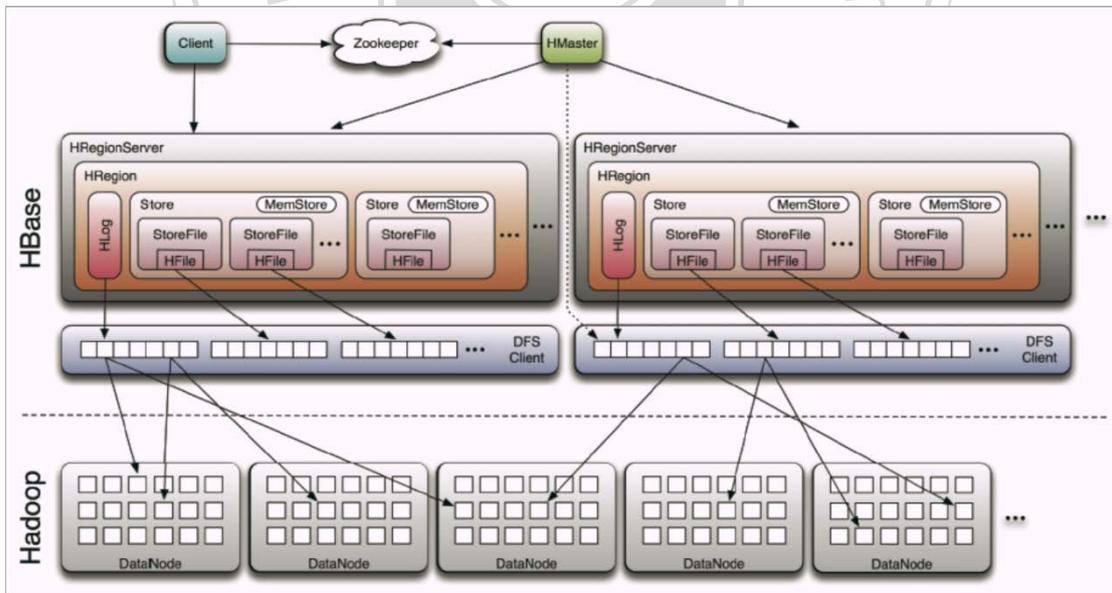


圖 3、HBase 架構圖

(資料來源：James Chin, Zikai Wang, 2011)

## 第四節 技術分析

技術分析 (Technical Analysis) 是基於金融商品過去的供需變動，使用歷史的股價資料，嘗試透過各項圖形或量化指標，以及統計方法來分析市場的價跟量、大盤指數，藉由分析過去的趨勢來預測個股未來的走勢。這些歷史資料包括過去的成交量、成交價等等。

Iati 指出技術分析的基本假設及應用如下列幾點 (The Economist, 2006)：

1. 市場上證券價格純粹由供需決定，與內含價值無關。
2. 歷史會一再重演，而投資者也會一再重複相同的行為。
3. 不論股價有任何的變動或訊息，均可由圖表或歷史數據察覺。
4. 投資者有時理性，有時非理性，這些因素不是基本分析所能察覺的，而只會反映於市場供需上。
5. 股價雖有些日常的小波動，但仍有其趨勢可循。
6. 技術分析的主要目的，在於研判股票的買賣時機。
7. 技術分析關切的是股價的波動與變化，而不是股價應有的水準。

技術分析主要可分為技術指標與圖形分析兩大類型 (Robert D., 1998)。技術指標是利用統計、計量方法等數理計算模型，將市場上描述市場行為的各種公開資料，轉換成具有分析未來走勢意義的參考數值，並能產生機械式買賣訊號，提供投資人判斷買賣股票時機的一種工具 (溫豐全, 2008)。

除了技術指標之外，另外一種常見的技術分析方法則是圖形分析。圖形分析是研究走勢圖支撐壓力與盤勢變化的分析方法。主要使用的工具為價格走勢圖。圖形分析可以再細分為為 K 線型態 (短線) 與價格型態 (中長線) 分析。

K 線型態是以日本德川幕府時代的米市交易分析方法為濫觴，它可提供短期交易心理的簡明輪廓，反映市場力量與心理狀態，顯示買、賣雙方之間的互動關係。K 線型態的組成是由以一到數根 (通常為五根以下) 不同種類之 K 線所組成的特定組合模式來分析標的的市場交易現況，是一種篇短線的技术分析方法。

本研究整理了 62 種 K 線型態（詳見附錄一），並以量化方法進行辨識，作為交易買賣之參考。

另外一種圖形分析方式為價格型態分析，以一段時間的價格走勢圖來做分析，主要注重的是大方向的走勢與價格的支撐和壓力區間，其中大部分時間走勢延續時的 K 線變化重要性降低，但關鍵時點具有特殊意義的 K 線變化依舊十分重要，例如長紅 K 線突破壓力區間。因此，若能搭配良好的基礎 K 線型態分析能力，將可以提高價格型態分析的品質。

圖形分析具有相當程度的主觀色彩，尤其是價格型態分析，分析的準確與否，絕大部分取決於分析者本身的經驗與技巧。因此，本研究暫時不將價格型態分析列入運用。



## 第五節 基因演算法

基因演算法是由密西根大學學者 Holland (1975) 根據自然界生物演化過程所設計的一套演算法。其主要概念在於模擬自然界中「物競天擇，適者生存」的演化法則，藉由一代接著一代的模擬演化，來找出目標問題的最佳解。以下將簡單介紹基因演算法之運作流程：

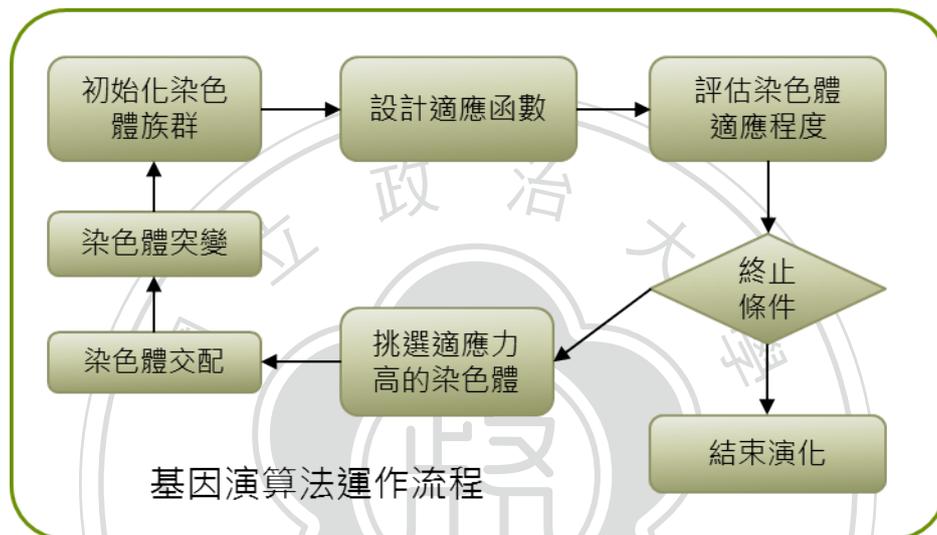


圖 4、基因演算法運作流程圖

(資料來源：本研究整理)

1. 染色體編碼 (Chromosome encoding)：將目標問題的所有可能解答編碼成方便進行演化運算的格式。
2. 適應函數 (Fitness function)：設計一套用來評估染色體適應程度的指標。
3. 根據適應函數計算所有染色體的適應程度。
4. 評估所有染色體的適應程度，如果滿足終止條件，則停止演化，如果尚未滿足，便繼續進行演化。
5. 挑出演化中適應力高的染色體族群。
6. 交配 (Crossover)：將高適應力的染色體相互交換基因以產生下一代染色體。

7. 突變 (Mutation)：隨機抽選部分染色體基因進行調整。

8. 重新計算交配與突變後的染色體適應程度，並重複步驟 4。

由於基因演算法具有高度解釋力，目前廣泛的被應用於半導體產業、航太工業、生技產業與財務金融等領域。關於基因演算法在本研究中之實際應用方式將於第三章做詳細說明。



### 第三章 研究流程與系統架構

傳統的系統化交易策略設計方法是由具有經驗的交易策略設計者，根據本身對各項量化技術指標的了解，將各指標互相搭配組合成一套的程式交易策略，再透過代入過往交易的歷史資料來進行回測。在多次的反覆調整策略與參數之後，最終產生出優秀的交易策略。傳統的策略設計方法產生出高品質的交易策略，但相對的此模式所需的進入門檻較高，交易策略設計者必須具備相當充足的金融市場相關知識與程式撰寫能力。

本研究的交易策略發展模式係由一套內建大量交易規則的基因庫中選取若干交易規則組成一套交易策略，再將交易策略視為染色體，運用基因演算法「物競天擇」的特性（詳細運作流程會在後面篇章介紹）從廣大的染色體族群中選出優秀的交易策略。除了內建的基因庫之外，使用者亦可以進階設定的方式將優秀的交易策略加入基因庫，進一步提升最終產出的交易策略品質。此模式除了能產生優秀的交易策略外，亦大幅降低了使用者的進入門檻。

## 第一節 研究流程



圖 5、本研究之研究流程圖

(資料來源：本研究整理)

步驟一：定義問題。

本研究將針對臺灣上市上櫃股票與臺灣指數期貨建議買賣時點的擇時交易策略。配合資金控管模式之輔助，利用基因演算法，藉由歷史資料的訓練找出最適當的交易策略。

步驟二：文獻探討。

整理建構基於 Hadoop 雲端運算架構之策略交易與回測模擬平台所需之各項技術與理論，包含雲端運算、Apache Hadoop、技術分析與基因演算法等相關知識。

步驟三：建立策略交易與回測模擬平台。

基於文獻探討中所討論之各項技術與理論，設計出適宜本研究使用之模型與架構，進而實作開發。

步驟四：找出績效良好之交易策略。

在策略交易與回測模擬平台開發完成後，將臺灣上市上櫃股票與臺灣指數期貨之歷史資料匯入，並運用本研究所提出之模型，測試各種不同的基因演算法參

數，找出績效良好之交易策略。

步驟五：系統架構調整與效能調校。

重新檢討系統架構並進行調整以改善系統運作效率。

步驟六：結論與未來研究方向。

探討本研究的發現、貢獻，對於不足或缺失的部分，提供可行的改善方案及未來研究方向。



## 第二節 系統架構

本研究之系統主要可分為三大區塊，分別為市場狀態計算模組、基因演算法模組及前端網頁使用者介面，如下圖。

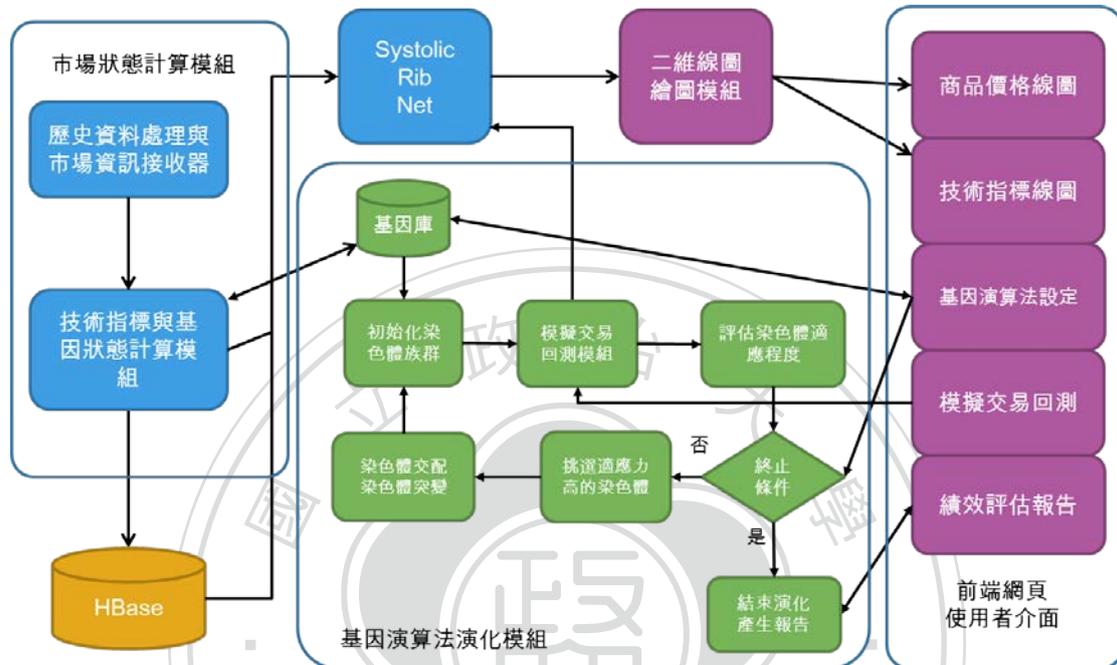


圖 6、系統架構圖

(資料來源：本研究整理)

以下章節將對此三大區塊逐一進行詳細介紹：

## 一、市場狀態計算模組

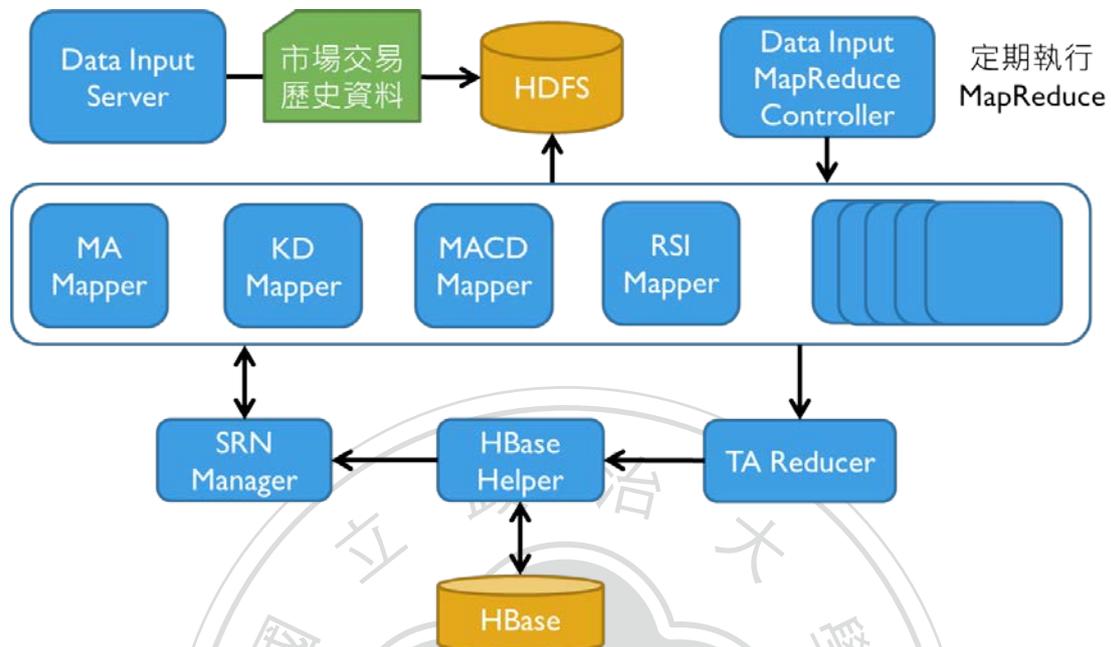


圖 7、市場狀態計算模組架構圖

(資料來源：本研究整理)

本研究中所稱之市場狀態係在某一時間點上，某項金融商品的所有技術分析指標值與該商品於本研究之基因庫中所有基因的狀態。在市場狀態計算模組的部分，先由 Data Input Server 負責介接市場交易資訊，並將市場交易資訊轉化為可用的輸入格式，存入 HDFS，作為市場狀態計算模組的輸入資料。

Data Input MapReduce Controller 會定期檢查 HDFS 上是否有未處理的市場交易資訊，如果有的話，就會啟動 Data Input MapReduce job，將 HDFS 未處理的市場交易資料，交給各個 Mapper 進行技術指標與基因狀態的計算。由於技術指標之間存在著相依性，例如 BIAS 指標的計算需要用到 MA 指標的值，MACD 指標的計算需要用到 EMA 指標。因此 Data Input MapReduce Controller 會控制各個指標 Mapper 運算的先後順序。各個 Mapper 在計算完成後會將結果寫入 Systolic Rib Net (將於下一段詳細介紹)，供其他相依的技術指標 Mapper 運算時使用。技術指標 Mapper 運算完成之後，TA Reducer 會將計算完成的結果蒐集之後存入

HBase。

Systolic Rib Net (以下簡稱為 SRN)，中文譯為「收縮式肋骨網絡」，是一種利用人類的脊椎構成概念來儲存金融商品交易資料的資料結構。在 SRN 中，將每一個單位時間的交易資料存成一個 Vertebra (脊椎骨)，由多個連續的 Vertebra 組成一條 Spine (脊椎)，一條 Spine 為一金融商品在某個時間粒度之交易資料的主軸。Spine 為一時間數列，而由於資料具有連續性的特性，每筆最新的原始資料會隨著時間的推移，一筆一筆加入至 Spine 的最前端，也就是說，Spine 會隨著時間的推移而生長。由於 Spine 為一時間數列，在 Spine 中的一個 Vertebra 可視為此時間數列的一個時間點。在這個時間點下的所有市場狀態資料將形成多條 Rib (肋骨)，連接在所屬的 Vertebra 之上。在 Rib 上所儲存之資料，依取用方式上的不同可分為 Horizontal 與 Vertical 兩種形式。Horizontal Data 是將 Rib 中的某一項技術指標或是基因狀態，以橫向時間序列的方式取出；Vertical Data 則是在某一個時間點下包含 Vertebra 本身的所有市場狀態資料。

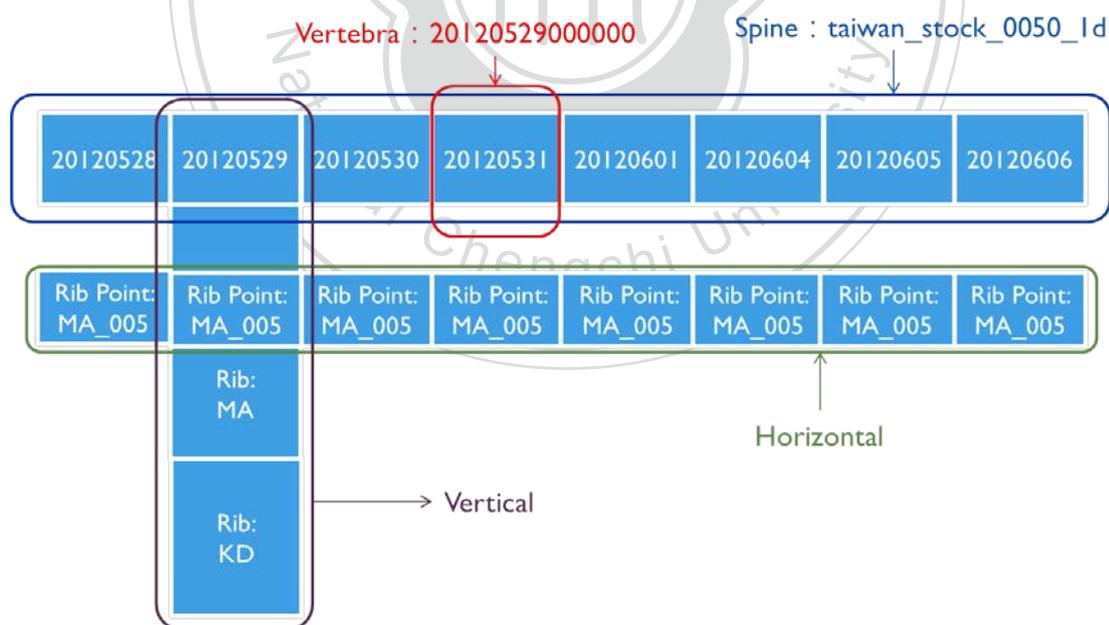


圖 8、SRN Model 示意圖

(資料來源：本研究整理)

SRN 能夠支援不同的時間數列，並包含在一個 SRN 群內，也提供不同目標

的資料運算。以期貨為例，使用者對於一支期貨做回測時，可選擇不同的時間精度，以秒、分鐘、小時做運算時，不同的時間精度需要不同的資料，因此能夠使用時間粒度為以秒為單位之 SRN，以分鐘為單位之 SRN，及以小時為單位之 SRN，這三個 SRN 將會被包含在同一商品的 SRN 群內。若使用者需要對以小時為單位之 SRN 做 MACD 指標之運算，則 SRN 中便有為了計算 MACD 之資料的 Rib，為了不同的計算目標，SRN 提供不同的 Rib 來儲存。

本研究為了降低回應時間 (Response Time)，採用預先計算的方式，在進入基因演算法演算階段前，就先將所有市場狀態計算完成並存放在 SRN 結構之中。

## 二、基因演算法模組

根據應用問題的不同，套用基因演算法也需要進行一些調整，以下將介紹基因演算法在本研究中應用之設計概念與系統架構。

### (一) 基因演算法模組設計概念

#### 1. 基因庫

基因是由一個或多個技術分析值所組成的一組判別式，是組成染色體且可抽換的最小單位，而此判別式的值則稱為基因狀態。基因庫是本研究中所有基因的集合。本研究整理了常用之技術分析指標與 K 線型態，並依其不同之特性，設計成若干不同之基因，詳細整理請參閱附錄二。

#### 2. 染色體編碼

一條染色體代表一個交易策略，由多組前述定義之基因所組成，並分為兩大部分—買進訊號與賣出訊號。每組基因具有一個權重值，當買進（賣出）訊號內基因狀態為真的所有基因權重值相加之後達到或超過買進（賣出）訊號門檻值，

則發生對應的買進（賣出）訊號。

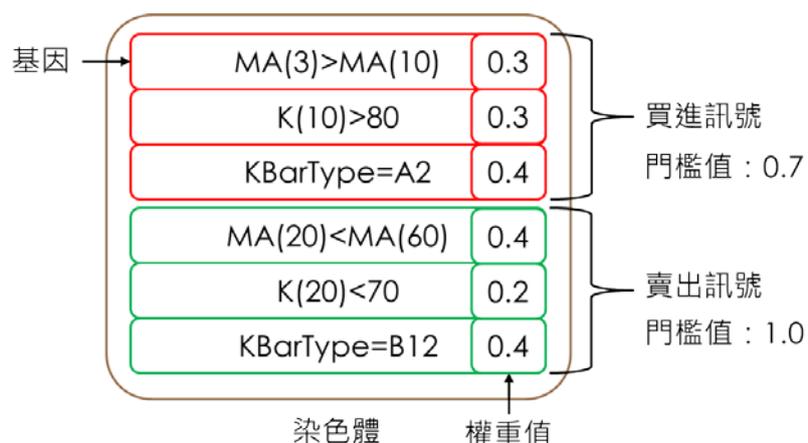


圖 9、染色體編碼示意圖

(資料來源：本研究整理)

### 3. 適應函數

適應函數所代表的是染色體的生存規則，亦是用來評估每個染色體代表問題解的優劣，不良的適應函數將會導致染色體錯誤的演化方向及結果。本研究以交易策略之報酬率來做為適應程度衡量的依據。回測後，系統將評估染色體的適應值，適應值越大表示該染色體內之基因越有機會被保留繼續演化。

### 4. 交配

交配指的是在演化過程中，具有高適應力的染色體交換彼此基因產生下一代染色體的過程。在本研究的設計中，染色體之間的交配有兩種模式，一種模式為買賣訊號的整組基因交配，另一種模式為買賣訊號的部分基因交配。

買賣訊號的整組基因交配指的是在交配時，前代的染色體將買進或賣出訊號以不切割的方式完整地移轉至下一代染色體上。

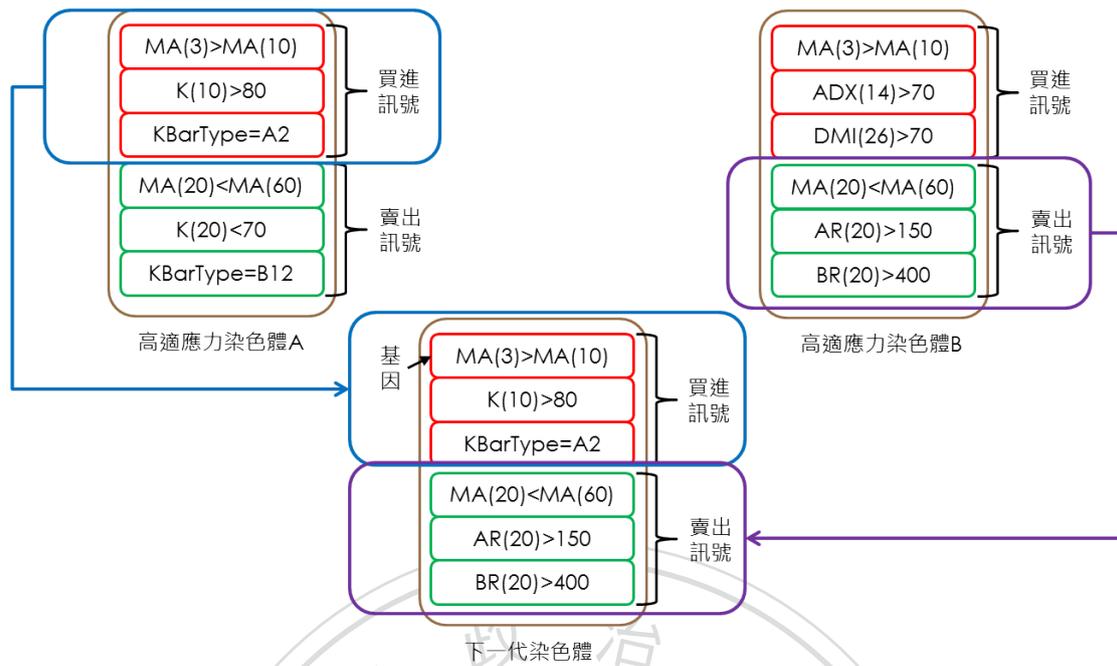


圖 10、整組基因交配示意圖

(資料來源：本研究整理)

買賣訊號的部分基因交配則是前代的染色體只將部分的買賣訊號基因移轉至下一代染色體上。

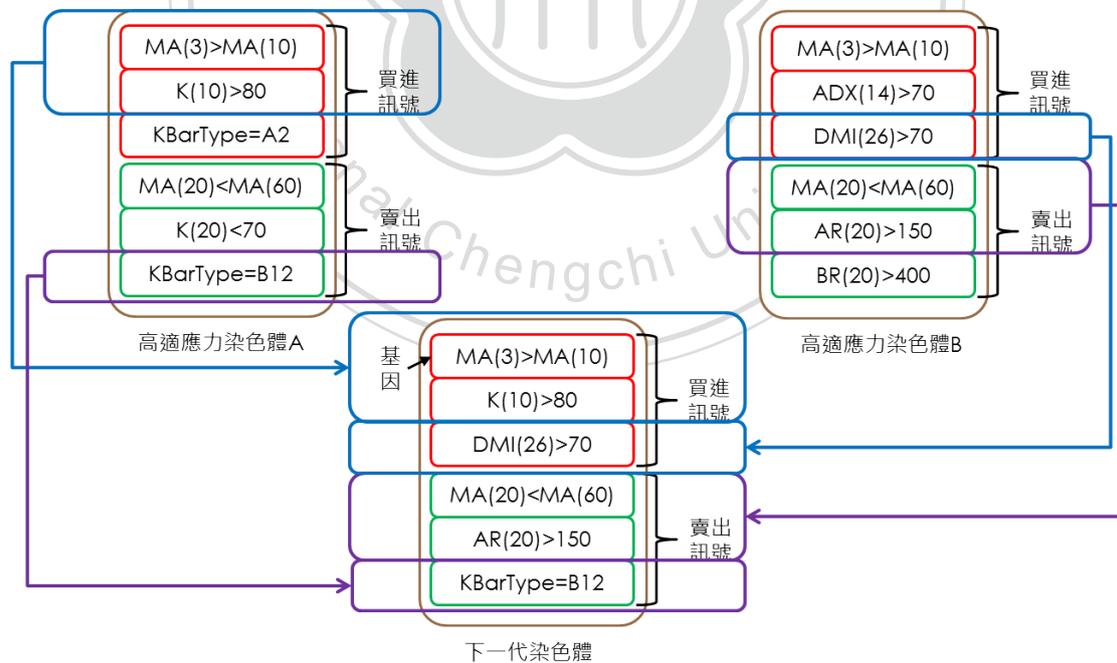


圖 11、部分基因交配示意圖

(資料來源：本研究整理)

## 5. 突變

突變係指在演化過程中，染色體中的基因隨機產生變化的情況。本研究中基因突變的方式有兩種，一種是基因位置交換突變，一種是基因參數突變。

基因位置突變是以交換買進訊號基因與賣出訊號基因位置的方式來產生突變。

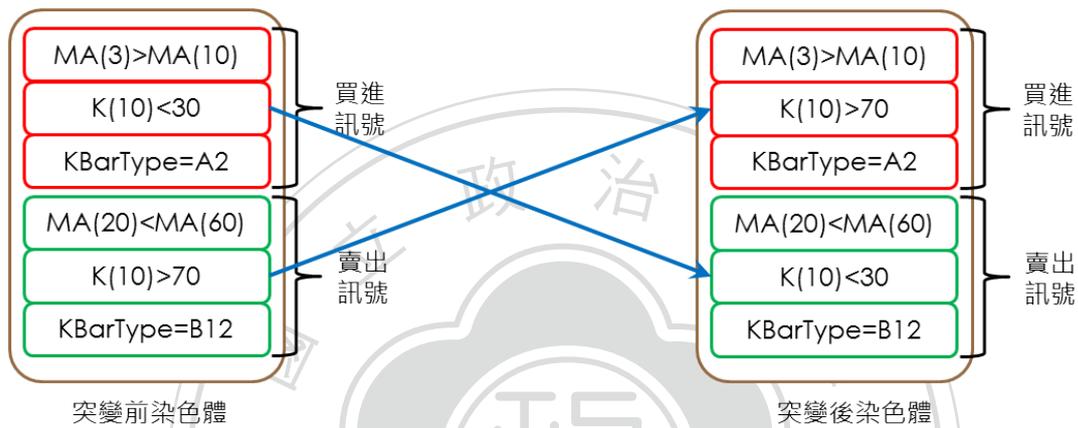


圖 12、基因位置突變示意圖

(資料來源：本研究整理)

基因參數突變則是以隨機抽換基因的方式來產生突變。

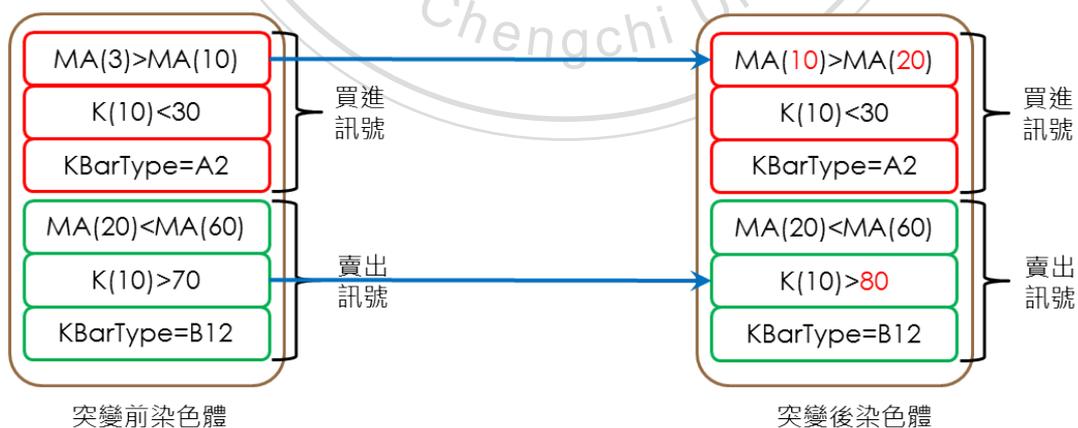


圖 13、基因參數突變示意圖

(資料來源：本研究整理)

## (二) 基因演算法模組系統架構

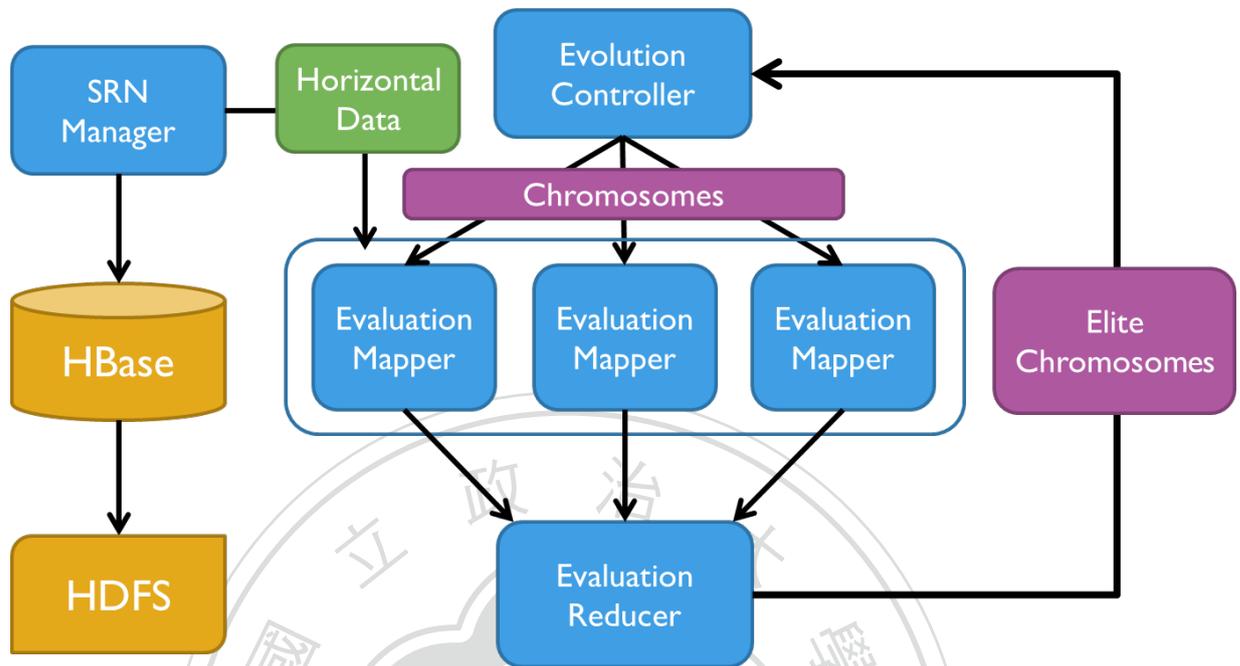


圖 14、基因演算法模組系統架構圖-Horizontal Design

(資料來源：本研究整理)

上圖為基因演算法模組之原始系統架構圖，主要是由 Evolution Controller 來控制整個演化的進行。Evolution Controller 在每一代演化之初，先準備當代演化所需的染色體族群，再將染色體族群分成若干個 split，交由 Evaluation Mapper 進行適應值計算。Evaluation Mapper 接收到 Chromosome 後，會檢視分配到的 Chromosome 需要那些基因狀態資料，再從 SRN Manager 調出所需的 Horizontal Data 進行適應值的計算。

此架構為原本 Mahout (與 Hadoop 搭配的 Machine Learning 函式庫) 的標準做法，但是實際進行基因演算法模擬回測時，此架構卻出現了運算效率低落的情形。在重新檢測每階段作業所花費的時間後，發現主要效能的瓶頸是 Evaluation Mapper 在計算適應值時，所進行大量的磁碟 I/O。每一次計算新的 Chromosome 都要重新讀出不同的 Horizontal Data，就算將讀出的 Horizontal Data 存入記憶體中的快取也因為擊中率 (Hit Rate) 太低而效益不彰。

因此，本研究重新調整了基因演算法模組之系統架構如下：

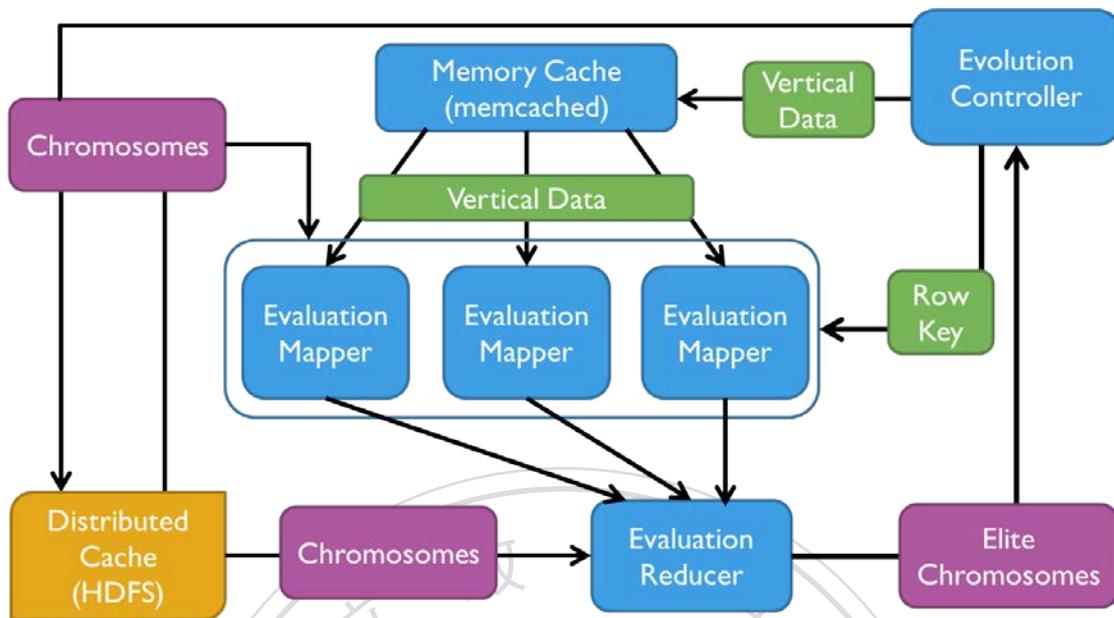


圖 15、基因演算法模組系統架構圖-Vertical Design

(資料來源：本研究整理)

在修正後的基因演算法模組系統架構中，仍是由 Evolution Controller 來控制整個基因演化的模擬過程。在一個世代的模擬演化開始進行之前，Evolution Controller 會先將該世代要參與模擬演化的染色體族群全部儲存到 Hadoop 所提供的 Distributed Cache 上面，再將模擬演化期間要用到的所有 Vertical Data 存放在 Memory Cache 當中。模擬演化開始之初，Evolution Controller 會將切割好的 Row Keys 傳給各個負責的 Evaluation Mapper，各 Evaluation Mapper 接到 Row Key 後會從 Distributed Cache 中讀出所有的 Chromosomes，再根據分配到的 Row Keys 到 Memory Cache 中取得相對應的 Vertical Data，最後計算出每個時間點所產生出的買賣訊號。Evaluation Mapper 計算出買賣訊號後，將結果交給 Evaluation Reducer，由 Evaluation Reducer 統整，計算出 Chromosomes 適應值後挑出優秀的 Chromosomes 交給 Evolution Controller 以進行下一代的演化。

Horizontal Design 與 Vertical Design 主要的差異分為以下三點：

### 1. MapReduce 的主要輸入資料差異

在 Horizontal Design 中，MapReduce 的主要輸入是染色體；在 Vertical Design 中，MapReduce 的主要輸入是市場狀態資料。Mahout 原先的架構主要是應用在包含大量染色體族群資料所進行的基因演算法模擬，但在本研究所應用的問題上，染色體族群的資料量大小是遠低於演化所需的環境資料 (Context)。故我們應該要調整需要被分散處理的主體，從染色體族群調整成市場狀態資料。

### 2. 市場狀態資料取用方式差異

在 Horizontal Design 中，市場狀態資料是以橫向時間序列的方式取用，雖然讀取完所需基因狀態之後即可完成整個演化期間的回測模擬，但是每次計算不同的染色體基因就必須重新讀取 Horizontal Data，即使相同的基因在不同的染色體中有重複出現仍必須重新讀取；在 Vertical Design 中，市場狀態資料是以縱向的方式取用，一次讀出該時間點所有的市場狀態資料。這個調整的優點在於讀取完一次即可完成所有染色體在該時間點的買賣訊號判斷，並保證同樣的資料未來不需再用到，可避免 Horizontal Design 中，相同資料會重新讀取的情形。

### 3. 市場狀態資料準備時機與記憶體快取的使用

有別於 Horizontal Design 當中，等到需要使用市場狀態資料時才到 HBase 讀取；Vertical Design 在模擬演化開始前，就先將需要用到的資料一次讀取完存入 Memory Cache 之中。這樣的作法可以減少磁碟資料存取次數，並用較快速的記憶體資料存取來取代磁碟資料存取。

## 三、前端網頁介面

本研究為提供跨平台、容易上手且可供大量使用者同時使用之策略交易平台，

在使用者介面上選擇了 web-based 的前端網頁介面。本研究之前端網頁介面包含以下區塊：

#### **(一) 市場狀態線圖瀏覽：**

以二維線圖的方式，展示出市場上所有交易相關資訊，包括價格、成交量與技術指標等時間序列資料。

#### **(二) 基因演算法設定：**

提供基因演算法模組所需之相關組態設定與基因庫擴充介面。使用者可以依自身需求切換基本設定與進階設定兩種版本，除了讓希望快速上手的使用者不需透過繁雜設定即可使用此平台，亦可讓進階使用者產生更有潛力的交易策略。

#### **(三) 模擬交易回測：**

模擬交易回測頁面用以啟動與顯示整個演化與交易回測過程。使用者可透過此頁面操作模擬交易回測模組、調整資金控管模式。

#### **(四) 績效評估報告：**

顯示基因演算法模組所產生的優秀交易策略結果報告，使用者可在此檢視各優秀交易策略之策略詳細內容、浮動淨值走勢、詳細買賣時點、最終損益、年化報酬率...等資訊。

# 第四章 實驗數據分析

## 第一節 模擬速度測試與分析

本研究在第三章時曾就 Horizontal Design 與 Vertical Design 兩種不同的模擬演化架構進行比較，本章節將實際測試兩種架構在三種不同測試環境中運算速度上的差異。模擬速度測試使用之基因演算法參數如下：

族群大小 (Population)：300

染色體長度 (Chromosome Length)：8

演化代數 (Generation)：100

回測期間：120 個交易日資料

以上參數設定會在三種不同測試環境中分別進行十次模擬，模擬結束後挑出最佳時間、最差時間並計算平均時間。

第一個測試環境為 Hadoop 叢集模式，包含一個 Jobtracker 與三個 Tasktracker。測試結果如下表：

	Horizontal Design	Vertical Design
最佳時間	14361.473	5813.361
最差時間	16057.031	7014.909
平均時間	15062.713	6475.719

(單位：秒)

表 1、模擬演化時間表(叢集模式-1 個 Jobtracker 和 3 個 Tasktracker)

(資料來源：本研究整理)

測試結果顯示 Vertical Design 的確能大幅度提升模擬演化速度，平均花費時間減少了 57%。縱使如此，本研究仍希望能再縮短模擬演化時間，在增加分擔運算工作的 Tasktracker 應可減少模擬演化時間的假設下，決定在第二個測試環境

中增加三個 Tasktracker。

第二個測試環境一樣是 Hadoop 叢集模式，包含一個 Jobtracker，Tasktracker 數目從三個增加到六個。測試結果如下表：

	<b>Horizontal Design</b>	<b>Vertical Design</b>
<b>最佳時間</b>	17652.592	7782.469
<b>最差時間</b>	20319.712	9480.042
<b>平均時間</b>	18727.631	8835.368

(單位：秒)

表 2、模擬演化時間表(叢集模式-1 個 Jobtracker 和 6 個 Tasktracker)

(資料來源：本研究整理)

在第二個測試環境中，雖然 Vertical Design 的表現依舊大幅優於 Horizontal Design，但是整體模擬演化的時間卻沒有如預期的因為加入三個 Tasktracker 而縮短，反而花費了更多時間。

為了找尋原因，本研究在執行模擬演化的同時，觀測各節點的處理器負載，發現各節點的處理器多數時間都是處於低負載的程度，只有少數時間到達中等負載。基於兩次測試與觀測各節點負載情況的結果，初步推斷縮短模擬演化時間主要的關鍵在於資料輸入輸出時間，而非運算時間。

為了印證這個推斷，本研究設計了第三個測試環境。第三個測試環境不再使用 Hadoop 叢集模式，改用 Hadoop 單機虛擬分散模式，並將原本的硬碟(Hard Disk Drive)更換成固態硬碟(Solid State Drive)。這樣的測試環境就可以將資料輸入輸出時間最小化，預期模擬回測速度會有更好的表現。實際執行第三個測試環境的測試數據如下；

	Horizontal Design	Vertical Design
最佳時間	5128.325	782.219
最差時間	6647.891	1500.242
平均時間	5855.723	1135.378

(單位：秒)

表 3、模擬演化時間表(單機虛擬分散模式)

(資料來源：本研究整理)

在第三個測試環境中，模擬演化的速度大幅改善。Vertical Design 的表現仍舊優於 Horizontal Design。跟叢集模式相比，單點模式的 Horizontal Design 縮短了約 61% 的模擬演化時間；Vertical Design 更是大幅縮短了約 82% 的模擬演化時間。

這樣的結果印證了模擬演化運算速度改善的主要瓶頸在於等待資料輸入輸出時間的假設，但也說明了本研究的模擬演化運算其實並不適合採用 Hadoop 的運算架構。主要關鍵在於本研究模擬演化運算所需時間跟資料輸入輸出與傳輸時間之間的比例。本研究的模擬演化運算在設計之初就已經將大部份的市場狀態預先計算好，並以基因的方式儲存在基因庫當中，故實際在模擬演化時所需要的運算量其實並不大。Hadoop 的平行運算架構適合用在運算時間大幅超過資料輸入輸出時間與平行運算準備時間的運算工作，與本研究模擬演化運算工作的特性並不相符。如此一來，也可以合理解釋為何在第二次測試中加入更多 Tasktracker 反而需要花費更久的時間。因為加入更多 Tasktracker 需要花費更多的網路傳輸與資料準備時間，而實際執行運算時間所占比例太小，導致分散式運算省下的運算時間遠不及為了達成分散所花費的時間。

## 第二節 策略回測實驗數據

染色體長度、染色體族群數量與模擬演化代數是影響本研究策略回測績效之三個關鍵的參數。本節將討論不同的染色體長度、染色體族群數量與模擬演化代數，對策略回測績效的影響，讓未來的使用者在設定參數上有所參考。

本研究對染色體長度選取了五個參數，分別為 2、4、8、16 與 32；染色體族群數量亦選取了五個參數，分別為 30、60、100、300 與 500；模擬演化代數則選取了六個參數，分別為 1、5、10、30、50 與 100。本研究將這些參數之所有可能組合進行測試，回測期間為三個月，測試標的為台灣五十指數股票型基金，測試結束後記錄各參數組產生之最佳染色體的獲利率，測試結果整理如下：



一、染色體長度：2

族群數量 \ 演化代數	30	60	100	300	500
1	0.00%	0.37%	3.37%	3.82%	3.67%
5	1.50%	1.12%	10.12%	7.65%	11.02%
10	3.30%	6.52%	11.25%	9.00%	19.80%
30	3.67%	14.17%	17.55%	26.55%	24.30%
50	1.48%	17.10%	19.57%	26.55%	26.77%
100	1.50%	22.50%	24.30%	26.77%	26.77%

表 4、染色體長度=2 時，不同染色體族群數量在不同演化代數之獲利率

(資料來源：本研究整理)

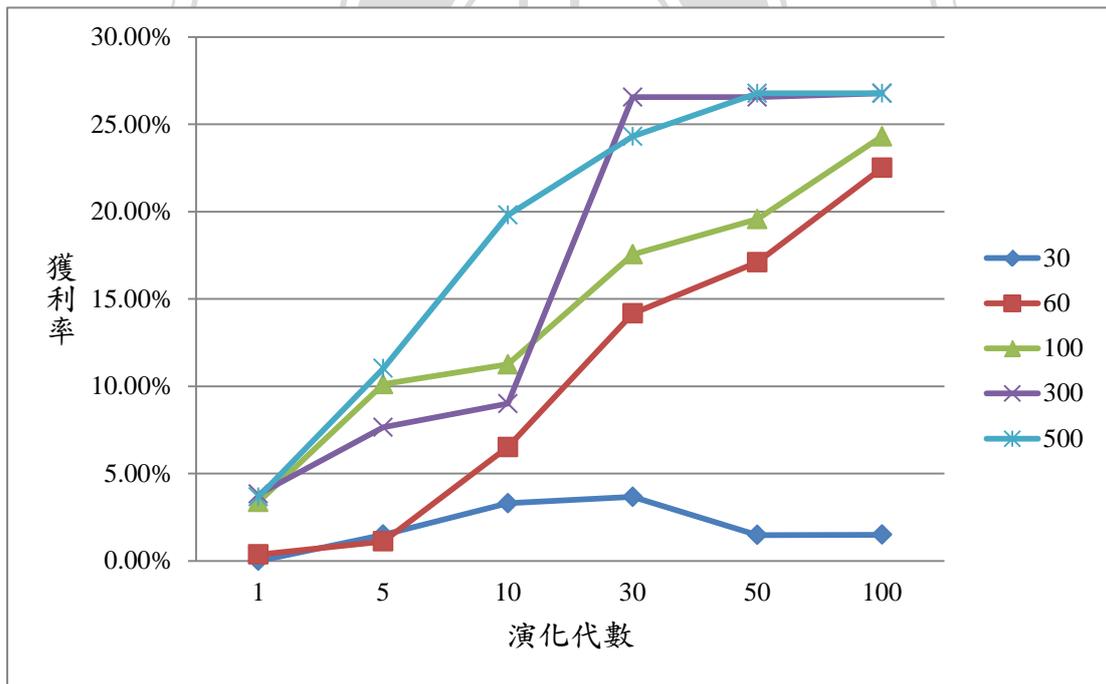


圖 16、染色體長度=2 時，不同染色體族群數量在不同演化代數之獲利率折線圖

(資料來源：本研究整理)

## 二、染色體長度：4

族群數量 \ 演化代數	30	60	100	300	500
1	0.07%	3.15%	2.47%	4.50%	3.37%
5	0.15%	9.45%	11.47%	13.50%	10.12%
10	0.18%	13.27%	16.87%	13.50%	10.35%
30	7.50%	20.02%	19.35%	19.80%	26.77%
50	11.25%	22.95%	20.70%	24.07%	27.45%
100	24.07%	22.95%	22.75%	24.75%	28.12%

表 5、染色體長度=4 時，不同染色體族群數量在不同演化代數之獲利率

(資料來源：本研究整理)

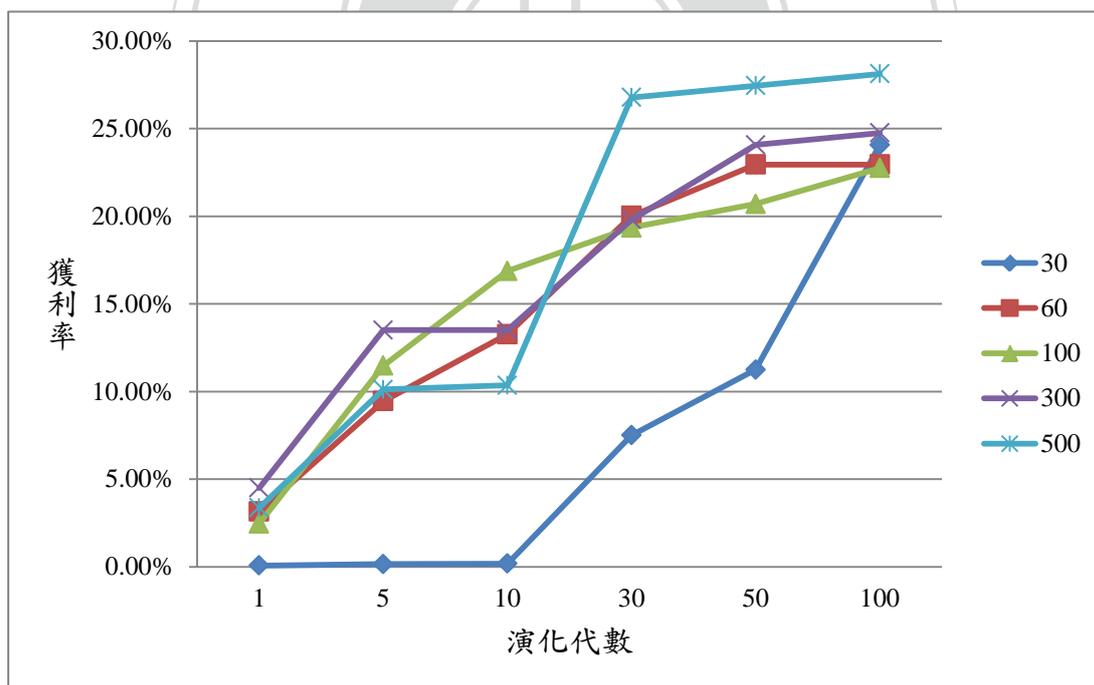


圖 17、染色體長度=4 時，不同染色體族群數量在不同演化代數之獲利率折線圖

(資料來源：本研究整理)

### 三、染色體長度：8

族群數量 \ 演化代數	30	60	100	300	500
1	1.50%	1.20%	2.25%	3.52%	3.07%
5	4.05%	8.70%	5.40%	20.70%	17.77%
10	9.60%	9.60%	6.97%	16.20%	17.77%
30	13.80%	17.32%	15.30%	26.50%	26.32%
50	15.30%	26.10%	21.82%	26.55%	26.55%
100	22.95%	28.35%	24.52%	26.32%	29.02%

表 6、染色體長度=8 時，不同染色體族群數量在不同演化代數之獲利率

(資料來源：本研究整理)

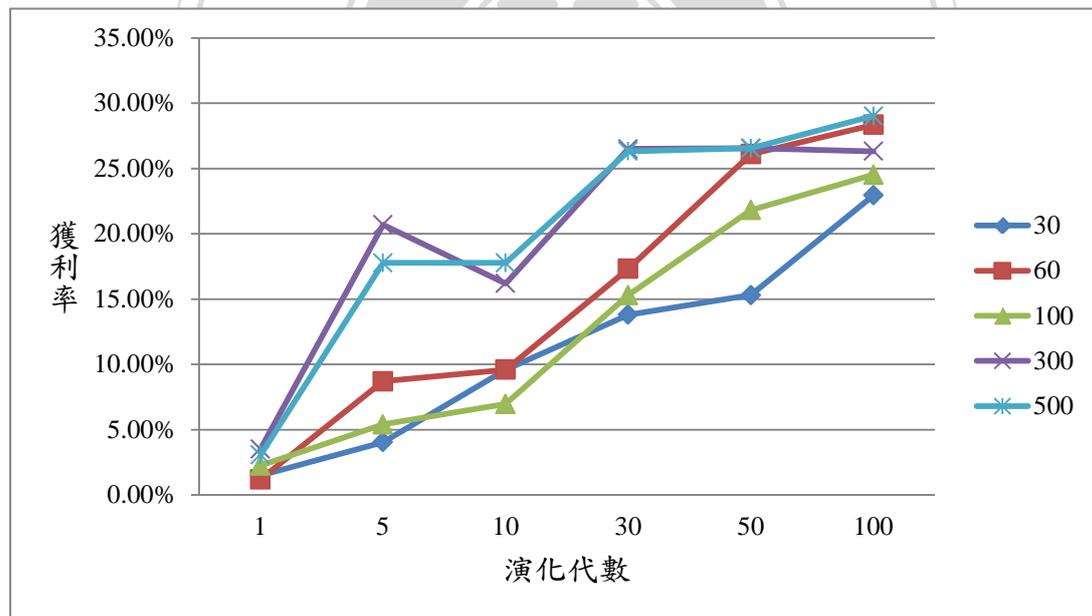


圖 18、染色體長度=8 時，不同染色體族群數量在不同演化代數之獲利率折線圖

(資料來源：本研究整理)

#### 四、染色體長度：16

族群數量 \ 演化代數	30	60	100	300	500
1	0.45%	1.95%	2.32%	3.60%	3.52%
5	0.60%	3.60%	1.80%	6.97%	13.05%
10	1.90%	9.00%	7.20%	9.75%	15.52%
30	5.80%	15.97%	15.90%	17.10%	14.62%
50	8.25%	11.55%	17.10%	14.62%	17.10%
100	6.42%	14.70%	13.50%	17.75%	19.80%

表 7、染色體長度=16 時，不同染色體族群數量在不同演化代數之獲利率

(資料來源：本研究整理)

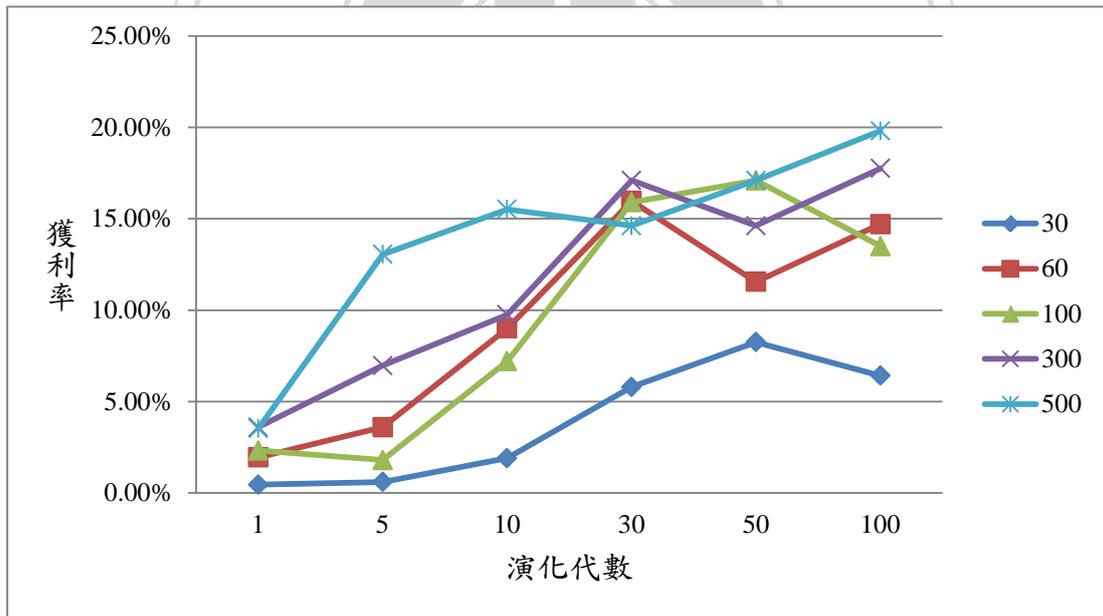


圖 19、染色體長度=16 時，不同染色體族群數量在不同演化代數之獲利率折線圖

(資料來源：本研究整理)

五、染色體長度：32

族群數量 \ 演化代數	30	60	100	300	500
1	0.00%	1.50%	0.75%	1.27%	2.77%
5	0.00%	0.00%	5.70%	6.75%	3.82%
10	0.00%	0.00%	8.50%	13.95%	13.15%
30	0.00%	0.00%	15.75%	17.10%	14.40%
50	0.00%	0.00%	13.05%	19.57%	18.67%
100	0.00%	0.00%	16.20%	16.42%	14.85%

表 8、染色體長度=32 時，不同染色體族群數量在不同演化代數之獲利率

(資料來源：本研究整理)

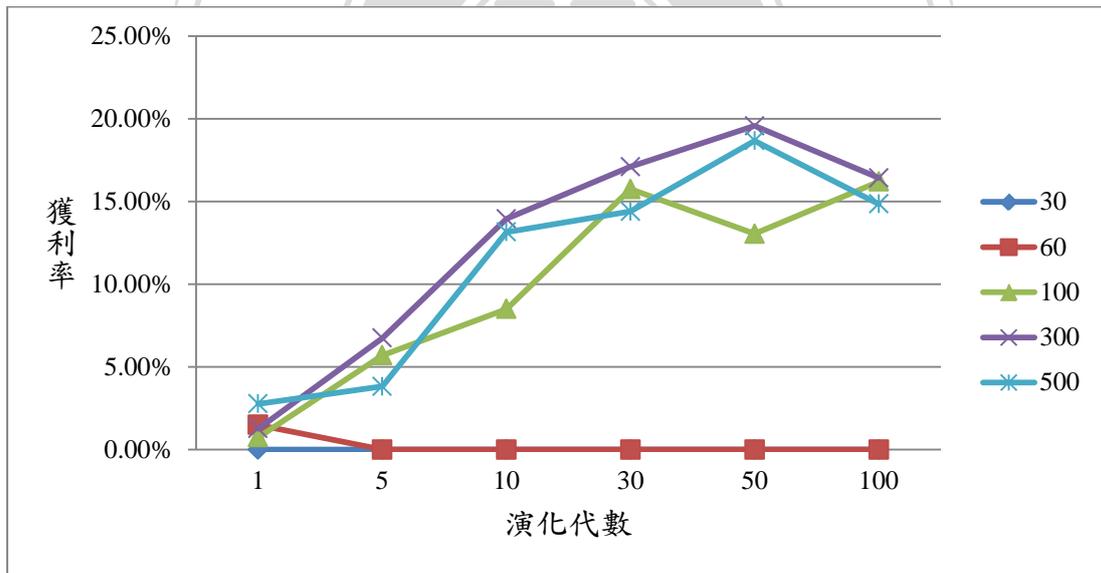


圖 20、染色體長度=32 時，不同染色體族群數量在不同演化代數之獲利率折線圖

(資料來源：本研究整理)

### 第三節 基因演算法參數設定

根據上一節的實測結果，本研究歸納出幾個參數設定重點，供使用者進行模擬演化時參考：

#### 一、染色體的適應程度會隨著演化代數上升，但有其極限。

在一段過去的時間區間內，金融商品價格具有有限的波動幅度，因此透過買賣操作該金融商品能獲得之超額報酬有其極限。在本研究中，染色體的適應程度即為各染色體在一段期間內對某項金融商品買賣之獲利能力，故染色體的適應程度也應該會有極限值存在。

在本研究的實驗中也印證了這樣的推論，染色體的適應程度（即獲利率），在演化到 24%~26% 左右的水平之後，就開始呈現穩定波動，不再持續上升。若是再持續增加演化代數，只會增加模擬時間，無法找到優秀的染色體。

#### 二、染色體族群數量越大，就能以越少的演化代數找到高適應力之染色體。

染色體族群數量代表的是每一代參與競爭的染色體總數。染色體族群數量若是太小，代表演化過程中缺乏競爭，不易產生優秀的染色體後代。反之，若是染色體族群數量夠大，整個染色體族群的複雜度高，在模擬演化前期（較少的演化代數）就容易出現優秀的染色體後代。在染色體長度為 4 和 8 的實驗中，當染色體族群數量為 30 或 60 時，都必須經過長期的演化，染色體的適應程度才慢慢改善。特別是在染色體長度為 2 的實驗中，染色體族群數量為 30 的實驗甚至無法產生出優秀的染色體後代。反觀染色體族群數量為 300、500 的實驗，幾乎都在 30 代的演化之後，就接近適應程度之極限。

#### 三、在本研究的基因編碼基礎之下，染色體長度不宜過長，建議低於 16。

在本研究的基因編碼基礎之下，染色體所包含的基因如果太多，各基因之間

容易產生互斥現象，不易產生交易訊號，獲利能力也較差。從實驗數據來看，在染色體長度為 2、4 和 8 的實驗當中產生出的優秀染色體適應程度遠高於染色體長度為 16、32 的實驗所產生出的優秀染色體。在染色體長度為 32 且染色體族群數量只有 30 或 60 的實驗中，甚至無法產生出能夠獲利的染色體。

綜合以上幾點，建議使用者在設定基因演算法模擬回測參數時，染色體長度應低於 16，染色體族群總數至少大於 100，演化代數則是最少大於 30。



# 第五章 結論

## 第一節 研究結論

傳統的系統化交易策略設計方法是由具有經驗的交易策略設計者，根據本身對各項量化技術指標的了解，將各指標互相搭配組合成一套的程式交易策略，再透過代入過往歷史交易資料來進行回測。在多次的反覆調整策略與參數之後，最終產生出優秀的交易策略。傳統的策略設計方法產生出高品質的交易策略，但相對的，此模式所需之進入門檻較高，交易策略設計者必須具備相當充足的金融市場相關知識與程式撰寫能力。

本研究以 Hadoop 雲端運算架構建立金融交易策略發展模式，由一套內建大量交易規則的基因庫中選取若干交易規則組成一套交易策略，再將交易策略視為染色體，運用基因演算法「物競天擇」的特性從廣大的染色體族群中選出優秀的交易策略。經過驗證，此模式的確能找出過往績效優秀的交易策略，除此之外，亦大幅降低了使用者在程式化交易上的進入門檻。使用者不再需要撰寫策略程式，只要選定標的、指定模擬演化區間，就能找出在該區間內表現良好的交易策略。

在基因演算法模組架構方面，本研究從 Horizontal Design 調整至 Vertical Design 的過程中，大幅度縮減了模擬演化的時間。雖然在模擬速度測試時發現，本研究基因演算法模擬演化的部分所需的運算時間並沒有遠遠超過等待資料輸入輸出的時間，使得 Hadoop 分散式運算架構於本研究之基因演算法模擬演化部分無法發揮應有的效益；但在批次處理大量歷史交易資料、計算技術指標與基因狀態時，Hadoop 架構仍是十分合適的解決方案。

最後，本研究對基因演算法在找尋優秀策略時所設定之參數進行討論，得到了以下三點結論：

一、染色體的適應程度會隨著演化代數上升，但有其極限。只要有足夠的染

色體族群數量（建議 300 以上），50 代的模擬演化就足以找到十分優秀之染色體。

二、染色體族群數量越大，就能以越少演化代數找到高適應力之染色體，以 300 到 500 之間的染色體族群數量最為合適。

三、在本研究的基因編碼基礎之下，染色體長度不宜過長，建議低於 16。

未來使用者透過本系統進行模擬演化時，可以基於這三點要領選擇所需之參數，即可以最有效率的方式，找到優秀的交易策略。

## 第二節 未來展望

本研究主要聚焦於基於 Hadoop 雲端運算架構之策略交易與回測模擬平台之架構設計與建置、探討調整架構所帶來的效能改善與找出優秀交易策略所需之基因演算法參數設定。但本研究所找出之優秀交易策略，乃是過去一段時間區間內之優秀交易策略，本研究並未進行這些優秀交易策略未來是否能夠持續帶來超額報酬之實證研究，後續研究者可針對此問題進行實證研究。除此之外，亦可針對各項不同之指標組合進行績效探討，為使用者提供更多參考依據。

# 參考文獻

## 中文部分

1. 王耀聰、辜文元、魏綸毅譯(2011)。Tom White 著。Hadoop 技術手冊(Hadoop: The Definitive Guide, Second Edition)。台北市：基峯資訊。
2. 陸嘉恆(2012)。Hadoop 實戰技術手冊。台北市：佳魁資訊。
3. 賴育漣(2009, 10月1日)。演算法交易漸成市場熱點。經濟日報。取自 [http://itrader168.blogspot.com/2009/09/blog-post\\_23.html](http://itrader168.blogspot.com/2009/09/blog-post_23.html)。
4. 杜金龍(2006)。最新技術指標－在臺灣股市應用的訣竅。台北市：財訊。
5. 林昇甫、徐永吉(2009)。遺傳演算法及其應用。台北市：五南。
6. 江吉雄(2002)。遺傳演算法於股市選股與擇時策略之研究(未出版碩士論文)。國立中央大學，桃園縣。
7. 姜林杰祐(2012)。「演算法交易與高頻交易」－台灣期交所課程教學資源。取自 <http://www.programtrading.tw/viewtopic.php?f=24&t=18840>。
8. 藍子軒譯(2011)。Ernest P. Chan 著。計量交易(Quantitative Trading)。台北市：寰宇。
9. 寰宇財務顧問公司譯(2006)。Gregory L. Morris 著。強力陰陽線(Candle Power)。台北市：寰宇。
10. 溫豐全(2008)。運用技術指標建構投資決策之知識架構(未出版碩士論文)。國立政治大學，台北市。
11. 楊雅菱(2011)。基於雲端環境與服務導向架構之交易策略評估平台框架(未出版碩士論文)。國立政治大學，台北市。

## 英文部分

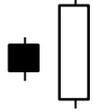
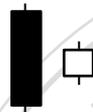
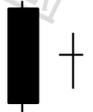
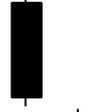
1. Kevin Kelly (2006) . Moving markets Shifts in trading patterns are making technology ever more important. *The Economist*. Retrived from [http://www.economist.com/node/5475381?story\\_id=E1\\_VQSVPR](http://www.economist.com/node/5475381?story_id=E1_VQSVPR).
2. National Institute of Standards and Technology(2010). *Cloud Computing Forum & Workshop*. Retrieved from <http://www.nist.gov/itl/cloud.cfm>.
3. Dean J, Ghemawat S (2008) . *MapReduce: Simplified data processing on large clusters*. Communications of the ACM, 2008,51(1): 107-113.
4. Apache (2009) . *Apache Hadoop 0.20 Documentation - HDFS Architecture*. Retrieved from [http://hadoop.apache.org/common/docs/r0.20.0/hdfs\\_design.html](http://hadoop.apache.org/common/docs/r0.20.0/hdfs_design.html)
5. Chang F, Dean J, Ghemawat S, et al (2006) . *Bigtable: A distributed structured data storage system*. The 7th OSDI. Seattle, Washington.
6. Robert D. Edwards, & John Magee (1998) , *Technical Analysis of Stock Trends*. J. Magee, Inc.
7. Rob Iati (2009) , *The Real Story of Trading Software Espionage*. Retrieved from <http://AdvancedTrading.com> .
8. Gregory L. Morris, Ryan Litchfield (2006) . *Candlestick Charting Explained: Timeless Techniques for Trading Stocks And Futures*. McGraw-Hill Professional.
9. J.H.Holland (2009) . *Adaption in Natural and Artificial Systems*. University of Michigan, United States.

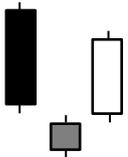
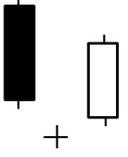
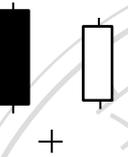
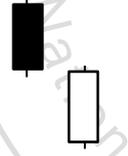
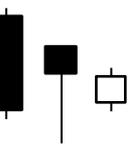
# 附錄一

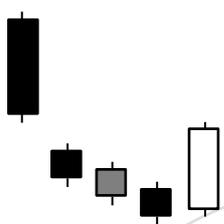
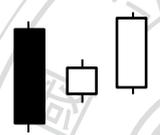
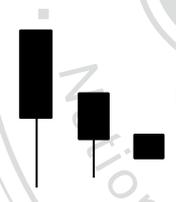
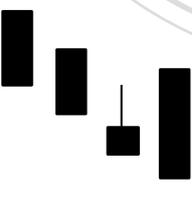
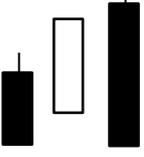
單一 K 線型態-共 22 種

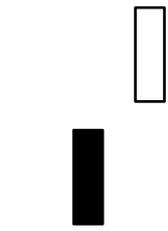
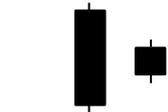
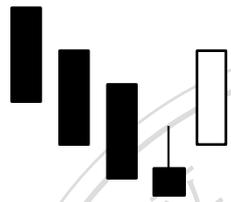
長紅線		長黑線		長腳十字	
短紅線		短黑線		長下影十字	
長紅實線		長黑實線		長上影十字	
紅收盤實線		黑收盤實線		墓碑十字	
紅開盤實線		黑開盤實線		蜻蜓十字	
紅紡錘線		黑紡錘線		星形十字	
長下影線紅線		長下影線黑線			
長上影線紅線		長上影線黑線			

K 線型態彙整-多頭反轉型態(出現於下跌趨勢中)-共 24 種

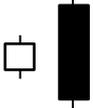
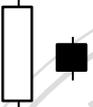
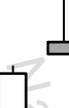
名稱	型態圖例	辨識重點
槌子 A1		<ol style="list-style-type: none"> <li>1. 下影線長度遠較實體為長。</li> <li>2. 很短或是沒有上影線。</li> <li>3. 實體的顏色並不重要。</li> </ol>
吞噬 A2		<ol style="list-style-type: none"> <li>1. 第二天的實體必須完全包含第一天的實體。</li> <li>2. 第二天的實體長度若超過第一天實體長度的 30% 以上，則此型態更為強而有力。</li> </ol>
母子 A3		<ol style="list-style-type: none"> <li>1. 第一天的實體必須完全包含第二天的實體。</li> <li>2. 第一天的實體長度若超過第二天實體長度的 30% 以上，則此型態更為強而有力。</li> </ol>
母子十字 A4		<ol style="list-style-type: none"> <li>1. 第二天的十字線處於先前長黑線的範圍內。</li> </ol>
倒狀槌子 A5		<ol style="list-style-type: none"> <li>1. 第二天的價格區間下端附近形成小實體。</li> <li>2. 第二天的上影線通常會超過實體長度的兩倍。</li> <li>3. 第二天下影線幾乎不存在。</li> </ol>
貫穿線 A6		<ol style="list-style-type: none"> <li>1. 第二天為長紅線，但其開盤價低於前一天的最低價。</li> <li>2. 第二天收盤價高於前一天實體的中點。</li> </ol>
星形十字 A7		<ol style="list-style-type: none"> <li>1. 第二天順勢跳空。</li> <li>2. 第二天為十字。</li> <li>3. 十字的影線不應該太長。</li> </ol>

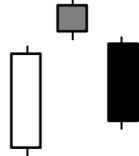
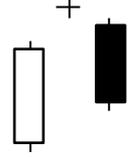
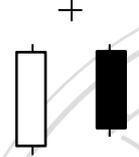
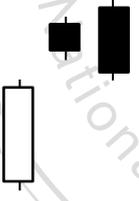
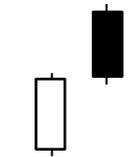
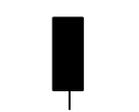
<p>晨星 A8</p>		<ol style="list-style-type: none"> <li>1. 第二天的星形與第一天的實體之間有跳空現象，且第三天的實體超過第一天實體的一半。</li> <li>2. 星形的顏色不重要。</li> <li>3. 第三天若成交量放大，則未來趨勢會更強烈。</li> </ol>
<p>晨星十字 A9</p>		<ol style="list-style-type: none"> <li>1. 型態與晨星類似，但第二天必須是跳空的星形十字。</li> <li>2. 比晨星更具反轉意義。</li> </ol>
<p>棄嬰 A10</p>		<ol style="list-style-type: none"> <li>1. 型態與晨星十字類似，但第二天的十字和第三天的下影線之間必須存在跳空。</li> </ol>
<p>三星 A11</p>		<ol style="list-style-type: none"> <li>1. 存在下降趨勢中</li> <li>2. 所有三天均收十字線。</li> <li>3. 第二天與前後兩天之間都有跳空。</li> </ol>
<p>遭遇線 A12</p>		<ol style="list-style-type: none"> <li>1. 第一天為長黑線，第二天為長紅線。</li> <li>2. 兩天收盤價相同。</li> <li>3. 兩天線形若都是收盤實體則效果更佳。</li> </ol>
<p>執帶 A13</p>		<ol style="list-style-type: none"> <li>1. 下降趨勢中，以最低價開盤，一路逆勢上漲，有長實體，無下影線。</li> <li>2. 若周圍出現很多類似型態，重要性便喪失了。</li> </ol>
<p>獨特三河床 A14</p>		<ol style="list-style-type: none"> <li>1. 第一天是長黑線。</li> <li>2. 第二天開盤價較第一天收盤價高，盤中創新低價，收盤則在最高價附近作收，造成短小實體的黑線。</li> <li>3. 第三天為短紅線，其實體位於第二天實體的下方。</li> </ol>

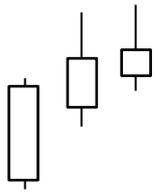
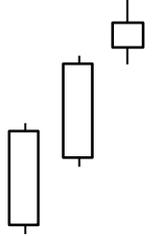
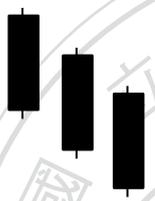
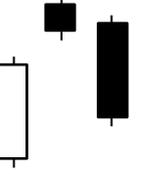
<p>三白兵 A15</p>		<ol style="list-style-type: none"> <li>1. 連續出現三天紅線，每天的收盤價均較前一天為高。</li> <li>2. 每天的開盤價均位於前一天實體之內。</li> <li>3. 每天的收盤價應該是最高價或是位於最高價附近。</li> </ol>
<p>起跑 A16</p>		<ol style="list-style-type: none"> <li>1. 第一天為長黑線。</li> <li>2. 第二天跳空下跌。</li> <li>3. 第三天、四天持續既定趨勢，第三天實體可為紅色或黑色。</li> <li>4. 第五天出現逆勢長紅，收盤價位於第一天和第二天的跳空缺口內。</li> </ol>
<p>內困三日 翻紅 A17</p>		<ol style="list-style-type: none"> <li>1. 前兩天為「母子」型態。</li> <li>2. 第三天收盤走高，確認反轉。</li> </ol>
<p>南方三星 A18</p>		<ol style="list-style-type: none"> <li>1. 第一天為長黑線，並具有長下影線。</li> <li>2. 第二天的線形基本上與前一天相同，但比較短。其最低價高於前一天的最低價。</li> <li>3. 第三天為短的黑色實線，其開、收盤價均位於前一天的價格區間內。</li> </ol>
<p>閨中乳燕 A19</p>		<ol style="list-style-type: none"> <li>1. 前兩天為連續的兩根黑色實線。</li> <li>2. 第三天的黑線於開盤時向下跳空。但是，當天的成交價曾經進入前一天的實體，形成很長的上影線。</li> <li>3. 第四根黑線完全吞噬第三天的線型，包括影線在內。</li> </ol>
<p>三明治 A20</p>		<ol style="list-style-type: none"> <li>1. 在下降趨勢中，一根黑線之後隨之出現一根紅線，其收盤價高於前一天黑線的收盤價。</li> <li>2. 第三天為黑線，其收盤價與第一天相同。</li> </ol>

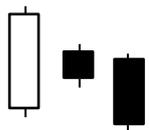
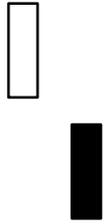
<p>反撲 A21</p>		<ol style="list-style-type: none"> <li>1. 在實線出現之後，立即出現相反顏色的實線。</li> <li>2. 兩根實線之間存在跳空缺口。</li> <li>3. 本型態所處之市場方向並不重要。</li> </ol>
<p>飛鴿歸巢 A22</p>		<ol style="list-style-type: none"> <li>1. 在下降趨勢中出現長黑線。</li> <li>2. 一根短黑線完全位於前一天的實體內。</li> </ol>
<p>梯底 A23</p>		<ol style="list-style-type: none"> <li>1. 連續三根黑線，其開、收盤價均不斷走低。</li> <li>2. 第四天為具有上影線的黑線。</li> <li>3. 最後一天為紅線，其開盤價位於前一天的實體之上。</li> </ol>
<p>低價配 A24</p>		<ol style="list-style-type: none"> <li>1. 下降趨勢中出現連續兩根黑線，兩天的收盤價相等。</li> </ol>

K 線型態彙整-空頭反轉型態(出現於上漲趨勢中)-共 22 種

名稱	型態圖例	辨認法則
吊人 B1		<ol style="list-style-type: none"> <li>1. 下影線長度遠較實體為長</li> <li>2. 很短或是沒有上影線</li> <li>3. 實體的顏色並不重要</li> </ol>
吞噬 B2		<ol style="list-style-type: none"> <li>1. 第二天的實體必須完全包含第一天之實體</li> <li>2. 第二天的實體長度若超過第一天實體長度的 30% 以上，則此型態更為強而有力</li> </ol>
母子 B3		<ol style="list-style-type: none"> <li>1. 第一天的實體必須完全包含第二天的實體</li> <li>2. 第一天的實體長度若超過第二天實體長度的 30% 以上，則此型態更為強而有力</li> </ol>
母子十字 B4		<ol style="list-style-type: none"> <li>1. 在上升趨勢中，出現一根長紅線</li> <li>2. 第二天的十字線處於先前長紅線的範圍內</li> </ol>
流星 B5		<ol style="list-style-type: none"> <li>1. 第二天實體必須有跳空的現象</li> <li>2. 第二天在價格區間下端附近形成小實體。</li> <li>3. 第二天的上影線通常會超過實體長度的兩倍</li> <li>4. 第二天下影線幾乎不存在</li> </ol>
烏雲罩頂 B6		<ol style="list-style-type: none"> <li>1. 第二天為長黑線，其開盤價高於前一天的最高價。</li> <li>2. 第二天的收盤價低於前一天長紅線實體中點。</li> </ol>
星形十字 B7		<ol style="list-style-type: none"> <li>1. 第二天順勢跳空。</li> <li>2. 第二天為十字。</li> <li>3. 十字的影線不應該太長。</li> </ol>

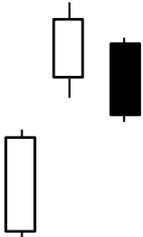
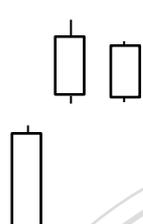
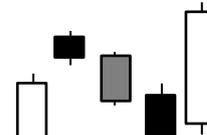
<p>夜星 B8</p>		<ol style="list-style-type: none"> <li>1. 第二天的星形與第一天的實體之間有跳空現象，且第三天的實體超過第一天實體的一半。</li> <li>2. 星形的顏色不重要。</li> <li>3. 第三天若成交量放大，則未來趨勢會更強烈。</li> </ol>
<p>夜星十字 B9</p>		<ol style="list-style-type: none"> <li>1. 型態與夜星類似，但第二天必須是跳空的星形十字。</li> <li>2. 比夜星更具反轉意義。</li> </ol>
<p>棄嬰 B10</p>		<ol style="list-style-type: none"> <li>1. 型態與夜星十字類似，但第二天的十字線和第三天的上影線之間必須存在跳空。</li> </ol>
<p>三星 B11</p>		<ol style="list-style-type: none"> <li>1. 存在上升趨勢中</li> <li>2. 所有三天均收十字線。</li> <li>3. 第二天與前後兩天之間都有跳空。</li> </ol>
<p>雙鴉躍空 B12</p>		<ol style="list-style-type: none"> <li>1. 第一天為順勢的長紅線</li> <li>2. 第二天出現向上跳空的黑線。</li> <li>3. 第三天開盤向上跳空，收盤卻低於第二天的黑線。其實體吞噬了第二天的黑線。</li> <li>4. 第三天黑線的收盤價仍高於第一天的收盤價。</li> </ol>
<p>遭遇線 B13</p>		<ol style="list-style-type: none"> <li>1. 第一天為長紅線，第二天為長黑線。</li> <li>2. 兩天收盤價相同。</li> <li>3. 兩天線形若都是收盤實體則效果更佳。</li> </ol>
<p>執帶 B14</p>		<ol style="list-style-type: none"> <li>1. 上升趨勢中，以最高價開盤，一路逆勢下跌，有長實體，無上影線。</li> <li>2. 若周圍出現很多類似型態，重要性便喪失了。</li> </ol>

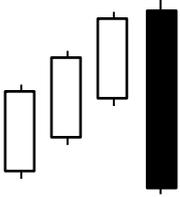
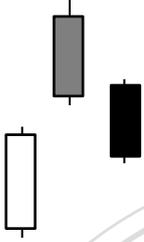
<p>大敵當前 B15</p>		<ol style="list-style-type: none"> <li>1. 連續三根紅線，每天的收盤價都比前一天為高。</li> <li>2. 每天的開盤價均位於前一天的實體內。</li> <li>3. 第二、三天出現相當長的上影線，顯示上升力道明確惡化。</li> </ol>
<p>步步為營 B16</p>		<ol style="list-style-type: none"> <li>1. 第二天為長紅實體。</li> <li>2. 第三天開盤價接近第二天收盤價。</li> <li>3. 第三天為紡錘線，也很可能是星形。</li> </ol>
<p>三烏鴉 B17</p>		<ol style="list-style-type: none"> <li>1. 連續三根黑線。</li> <li>2. 每天的收盤價都創新低。</li> <li>3. 每天的開盤價都在前一天的實體內。</li> <li>4. 每天的收盤價均以當天最低價或在其附近作收。</li> </ol>
<p>三胎鴨 B18</p>		<ol style="list-style-type: none"> <li>1. 三根連續的長黑線。</li> <li>2. 每天的開盤價約略等於前一天的收盤價。</li> </ol>
<p>起跑 B19</p>		<ol style="list-style-type: none"> <li>1. 第一天為長紅線。</li> <li>2. 第二天跳空上漲。</li> <li>3. 第三天、四天持續既定趨勢，第三天實體可為紅色或黑色。</li> <li>4. 第五天出現逆勢長黑，收盤價位於第一天和第二天的跳空缺口內。</li> </ol>
<p>雙鴨 B20</p>		<ol style="list-style-type: none"> <li>1. 上升趨勢中出現一根長紅線。</li> <li>2. 第二天收黑，但出現向上跳空缺口。</li> <li>3. 第三天為黑線，其開盤價位於第二天的實體內，收盤價則位於第一天的實體內。</li> </ol>

<p>內困三日 翻黑 B21</p>		<ol style="list-style-type: none"> <li>1. 前兩天為「母子」型態。</li> <li>2. 第三天收盤走低，確認反轉。</li> </ol>
<p>反撲 B22</p>		<ol style="list-style-type: none"> <li>1. 在實線出現之後，立即出現相反顏色的實線。</li> <li>2. 兩根實線之間存在跳空缺口。</li> <li>3. 本型態所處之市場方向並不重要。</li> </ol>



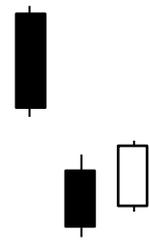
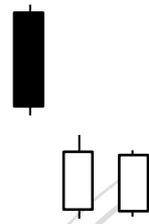
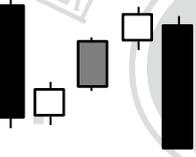
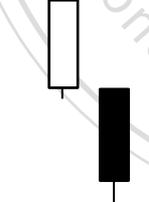
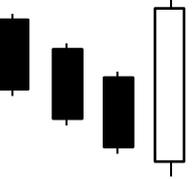
K 線型態彙整-多頭延續型態(出現於上漲趨勢中)-共 7 種

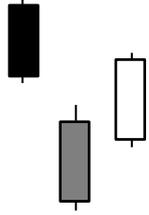
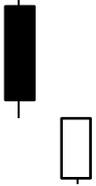
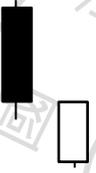
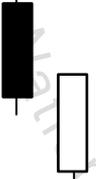
<p>上肩帶缺口 C1</p>		<ol style="list-style-type: none"> <li>1. 第一、二天都是上漲的紅 K 線。</li> <li>2. 第一、二天中間存在跳空缺口。</li> <li>3. 第三天為下跌的黑 K 線，開盤價位於前一天的實體內，收盤價位於跳空缺口內，但未完全填補該跳空缺口。</li> </ol>
<p>併肩白線 C2</p>		<ol style="list-style-type: none"> <li>1. 第一、二天都是上漲的紅 K 線。</li> <li>2. 第一、二天中間存在跳空缺口。</li> <li>3. 第三天跟第二天一樣也是一根類似長度的白線，且其開盤價大致與第二天之開盤價相同。</li> </ol>
<p>上升三法 C3</p>		<ol style="list-style-type: none"> <li>1. 第一天為上漲的長紅線。</li> <li>2. 在第一天後出現三根實體短小的陰陽線，這群短小陰陽線顏色不重要，但最好是黑色，並呈現下跌的走勢，而且必須位於第一天的最高價範圍之內。</li> <li>3. 最後一天出現強勁的漲勢，收盤價則超越第一天的收盤價。</li> </ol>
<p>隔離線 C4</p>		<ol style="list-style-type: none"> <li>1. 第一天為下跌的長黑 K 線。</li> <li>2. 第二天為紅開盤實線(執帶)。</li> <li>3. 兩天的開盤價大約相同。</li> </ol>
<p>執墊 C5</p>		<ol style="list-style-type: none"> <li>1. 第一天為上漲的長紅線。</li> <li>2. 第二天出現向上跳空的黑線。</li> <li>3. 隨後兩天為盤整走勢的短小 K 線，顏色並不重要，最低價不能低於第一天開盤價。</li> <li>4. 第五天為強勁的漲勢，收盤創新高。</li> </ol>

<p>三線反擊 C6</p>		<ol style="list-style-type: none"> <li>1. 前三天為不斷創新高的紅 K 線。</li> <li>2. 第四天開盤價創新高，隨後價格急挫，最低價較第一根紅線還低。</li> </ol>
<p>上缺口三 法 C7</p>		<ol style="list-style-type: none"> <li>1. 第一天為上漲的長紅線。</li> <li>2. 第二天亦為長 K 線，但顏色不重要，與第一天之間存在跳空缺口。</li> <li>3. 第三天為下跌的黑 K 線，開盤價位於第二天的實體之間，收盤價位於第一天的實體之間。</li> </ol>



K 線型態彙整-空頭延續型態(出現於下跌趨勢中)-共 9 種

<p>下肩帶缺口 D1</p>		<ol style="list-style-type: none"> <li>1. 第一、二天都是下跌的黑 K 線。</li> <li>2. 第一、二天中間存在跳空缺口。</li> <li>3. 第三天為上漲的紅 K 線，開盤價位於前一天的實體內，收盤價位於跳空缺口內，但未完全填補該跳空缺口。</li> </ol>
<p>併肩白線 D2</p>		<ol style="list-style-type: none"> <li>1. 第一天是下跌的黑 K 線。</li> <li>2. 第二天跳空開低，盤中上漲，收盤形成紅 K 線，但仍留有跳空缺口。</li> <li>3. 第三天跟第二天一樣也是一根類似長度的白線，且其開盤價大致與第二天之開盤價相同。</li> </ol>
<p>下降三法 D3</p>		<ol style="list-style-type: none"> <li>1. 第一天為下跌的長黑 K 線。</li> <li>2. 在第一天後出現三根實體短小的陰陽線，這群短小陰陽線顏色不重要，但最好是紅色，並呈現上漲的走勢，而且必須位於第一天的高低價範圍之內。</li> <li>3. 最後一天出現強勁的跌勢，收盤價則低於第一天的收盤價。</li> </ol>
<p>隔離線 D4</p>		<ol style="list-style-type: none"> <li>1. 第一天為上漲的長紅 K 線。</li> <li>2. 第二天為黑開盤實線(執帶)。</li> <li>3. 兩天的開盤價大約相同。</li> </ol>
<p>三線反擊 D5</p>		<ol style="list-style-type: none"> <li>1. 前三天為不斷創新低的黑 K 線。</li> <li>2. 第四天開盤價創新低，隨後價格大幅彈升，最高價高於第一根黑線的最高價。</li> </ol>

<p>下缺口三 法 D6</p>		<ol style="list-style-type: none"> <li>1. 第一天為下跌的長黑線。</li> <li>2. 第二天亦為長K線，但顏色不重要，與第一天之間存在跳空缺口。</li> <li>3. 第三天為上漲的黑K線，開盤價位於第二天的實體之間，收盤價位於第一天的實體之間。</li> </ol>
<p>頸上線 D7</p>		<ol style="list-style-type: none"> <li>1. 第一天為下跌的長黑線。</li> <li>2. 第二天為紅線但不為長紅線，其開盤價低於前一天的最低價。</li> <li>3. 第二天的收盤價約等於前一天的最低價。</li> <li>4. 第二天的成交量相對較大。</li> </ol>
<p>頸內線 D8</p>		<ol style="list-style-type: none"> <li>1. 第一天為下跌的長黑線。</li> <li>2. 第二天為紅線但不為長紅線，其開盤價低於前一天的最低價。</li> <li>3. 第二天的收盤價約等於前一天的收盤價。</li> <li>4. 第二天的成交量相對較大。</li> </ol>
<p>戮入線 D9</p>		<ol style="list-style-type: none"> <li>1. 第一天為下跌的長黑線。</li> <li>2. 第二天為長白線，其開盤價遠低於前一天的最低價。</li> <li>3. 第二天收盤價位於前一天的實體內，但低於實體中點。</li> </ol>

## 附錄二

	技術指標名稱	公式	相關參數	參數範圍	基因狀態
1	移動平均線(Moving Average, MA)	$N \text{ 日 MA} = \frac{N \text{ 日收盤價總和}}{N \text{ 日}}$	N 為正整數 short、long 為整數， short < long	short, long, N ∈ {3,5,10,15,20,30, 40,50,60,80,100,120, 140,160,180,200,220, 240,300,360}	1. $MA_{\text{short}} > MA_{\text{long}}$ 2. $MA_{\text{short}} < MA_{\text{long}}$ 基因總數：380
2	乖離率指標(BIAS) (相依於 MA)	$N \text{ 日 BIAS} = \frac{(\text{當日股價收盤價} - N \text{ 日內股價的移動平均值})}{N \text{ 日內股價的移動平均值}} \times 100\%$	N 為正整數	N ∈ {3,5,8,12,16,20, 26,32,45,60,80,100, 120,140,160,180,200, 220,240,300,360} $\beta \in \{0.2,0.25,0.3,0.35,0.4,$ 0.45,0.5,0.55,0.6,0.65,0.7}	1. $BIAS(N) < -\beta$ 2. $BIAS(N) > \beta$ 基因總數：462
3	指數移動平均 (Exponential Moving Average, EMA)	N 日的 $EMA_t =$ $EMA_{t-1} + \left[ \left( \frac{2}{N+1} \times (Close_t - EMA_{t-1}) \right) \right]$	K 為平滑常數 $K = \frac{2}{N+1}$ $EMA_t$ 代表當日的 EMA $EMA_{t-1}$ 代表前一日 的 EMA	short, long, N ∈ {3,5,10,15,20,30, 40,50,60,80,100,120, 140,160,180,200,220, 240,300,360}	1. $EMA_{\text{short}} > EMA_{\text{long}}$ 2. $EMA_{\text{short}} < EMA_{\text{long}}$ 基因總數：380

4	指數平滑異同平均線 (Moving Average Convergence Divergence · MACD) (相依於 EMA)	$DIF = EMA(\text{short}) - EMA(\text{long})$ $MACD_t = MACD_{t-1} + K \times (DIF_t - MACD_{t-1})$	short、long 為整數 $\text{short} < \text{long}$ $K = \frac{2}{N+1}$	short, long, N $\in \{3,5,8,12,16,20,$ 26,32,45,60,80,100, 120,140,160,180,200, 220,240,300,360\}	<ol style="list-style-type: none"> <li>1. <math>DIF &gt; 0</math></li> <li>2. <math>MACD &gt; 0</math></li> <li>3. <math>MACD &gt; DIF</math></li> <li>4. <math>DIF &lt; 0</math></li> <li>5. <math>MACD &lt; 0</math></li> <li>6. <math>MACD &lt; DIF</math></li> </ol> 基因總數：252
5	隨機指標(Stochastic, KD Line)	未成熟隨機值(Raw Stochastic Value, RSV) $RSV_t = \frac{Close_t - Low_N}{High_N - Low_N} \times 100$ $K_t = \frac{2}{3} \times K_{t-1} + \frac{1}{3} \times RSV_t$ $D_t = \frac{2}{3} \times D_{t-1} + \frac{1}{3} \times K_t$	$Close_t$ 為當天收盤價 $Low_N$ 為 N 天內最低價 $High_N$ 為 N 天內最高價 $K_t$ 為當日 K 值 $D_t$ 為當日 D 值	$N \in \{3,5,8,12,16,20,$ 26,32,45,60,80,100, 120,140,160,180,200, 220,240,300,360\} $\alpha \in \{95,90,85,80,75,70\}$ $\beta \in \{5,10,15,20,25,30\}$	<ol style="list-style-type: none"> <li>1. <math>K &gt; D</math></li> <li>2. <math>K &lt; D</math></li> <li>3. <math>D &gt; \alpha</math></li> <li>4. <math>K &gt; \alpha</math></li> <li>5. <math>D &lt; \beta</math></li> <li>6. <math>K &lt; \beta</math></li> <li>7. <math>3K - 2D &lt; \beta</math></li> <li>8. <math>2D - 3K &gt; \beta</math></li> </ol> 基因總數：798

6	威廉 R 指標 (WMS%R)	$N \text{ 日 WMS\%R} = \frac{High_N - Close_t}{High_N - Low_N} \times 100\%$	N 為正整數	$N \in \{3,5,8,12,16,20,26,32,45,60,80,100,120,140,160,180,200,220,240,300,360\}$ $\alpha \in \{0.05,0.1,0.15,0.2,0.25,0.3\}$ $\beta \in \{0.7,0.75,0.8,0.85,0.9,0.95\}$	1. WMS%R < $\alpha$ 2. WMS%R > $\beta$ 基因總數：252
7	趨向指標(Directional Movement Index · DMI)	<p>Step1.趨向變動值(Directional Movement · DM)</p> <p>+DM = 本日最高價 - 昨日最高價</p> <p>-DM = 本日最低價 - 昨日最低價</p> <p>Step2.真實的波幅(True Range · TR)</p> $TR = \text{MAX}[ High_t - Low_t ,  High_t - Close_{t-1} ,  Low_t - Close_{t-1} ]$ <p>Step3.方向指標(Directional Indicator · DI)</p> $+DI_N = \frac{+DM_N}{TR_N} ; -DI_N = \frac{-DM_N}{TR_N}$ <p>Step4.平均趨向指標(Average Directional Movement Index · ADX)</p> $DX = \frac{ (+DI) - (-DI) }{ (+DI) + (-DI) } \times 100\%$ $ADX_N = \frac{(N-1) \times ADX_{N-1} + DX_t}{N}$	N 為正整數 +DI 表示上升方向指標	$N \in \{3,5,8,12,16,20,26,32,45,60,80,100,120,140,160,180,200,220,240,300,360\}$ $\alpha \in \{0.5,0.1,0.15,0.2,0.25,0.3,0.35,0.4\}$ $\beta \in \{0.6,0.65,0.7,0.75,0.8,0.85,0.9,0.95\}$ $\gamma \in \{0.2,0.25,0.3,0.35,0.4,0.45,0.5,0.55,0.6\}$	1. +DI > $\gamma$ 2. -DI > $\gamma$ 3. ADX < $\alpha$ (N) 4. ADX > $\beta$ (N) 基因總數：525

8	動量指標(Momentum Index , MTM)	$MTM_N = Close_t - Close_{t-N}$	N 為正整數 $Close_{t-N}$ 為 N 日前之收盤價	short, long, N $\in \{3,5,8,12,16,20,26\}$	1. $MTM_{short} > MTM_{long}$ 2. $MTM_{short} < MTM_{long}$ 基因總數：42
9	震盪量指標 (Oscillator , OSC)	$OSC_N = \frac{Close_t}{Close_{t-N}} \times 100\%$	N 為正整數 $Close_{t-N}$ 為 N 日前之收盤價	$N \in \{3,5,8,12,16,20,26\}$ $\beta \in \{5,10,15,20,25,30,35,40\}$	1. $OSC_N > 100 + \beta$ 2. $OSC_N < 100 - \beta$ 基因總數：112
10	相對強弱指數(Relative Strength Index , RSI)	$RSI_N = 100 - \frac{100}{1 + RS}$ $RS = \frac{N \text{ 日內股價上漲幅度總和之平均值}}{N \text{ 日內股價下跌幅度總和之平均}}$	N 為正整數 short、long 為整數 short < long	$N \in \{3,5,8,12,16,20,26,32,45,60,80,100,120,140,160,180,200,220,240,300,360\}$ $\alpha \in \{95,90,85,80,75,70\}$ $\beta \in \{5,10,15,20,25,30\}$	1. $RSI_{short} > RSI_{long}$ 2. $RSI_{short} < RSI_{long}$ 3. $RSI_N > \alpha$ 4. $RSI_N < \beta$ 基因總數：672

11	加權相對強弱指數 (Weighted Relative Strength Index · WRSI) (相仿於 RSI)	N 日的 $WRSI_t = \frac{2(1 * RSI_{t-N} + 2 * RSI_{t-N+1} + \dots + N * RSI_t)}{N(N+1)}$	N 為正整數 short、long 為整數 short < long	$N \in \{3,5,8,12,16,20,26,32,45,60,80,100,120,140,160,180,200,220,240,300,360\}$ $\alpha \in \{95,90,85,80,75,70\}$ $\beta \in \{5,10,15,20,25,30\}$	1. $WRSI_{short} > WRSI_{long}$ 2. $WRSI_{short} < WRSI_{long}$ 3. $WRSI_N > \alpha$ 4. $WRSI_N < \beta$ 基因總數：672
12	中間意願指標 (CR)	$CR = \frac{\text{多方強度}}{\text{空方強度}} \times 100$ a：CR 的 10 天平均線 b：CR 的 20 天平均線 c：CR 的 40 天平均線 d：CR 的 62 天平均線 a、b 兩線所合成的區域叫副地震帶 c、d 兩線所合成的區域叫主地震帶	中間價 = $\frac{(H+L)}{2}$ 上升值 = $H - \text{Last } L(\text{負值計 } 0)$ 下跌值 = $\text{昨天中間價} - L(\text{負計 } 0)$ 多方強度 = N 天的上升值的和 空方強度 = N 天的下跌值的和	$MA \in \{10,20,40,62\}$ $N \in \{3,5,8,12,16,20,26,40,62\}$ $\alpha \in \{300,325,350,375,400\}$ $\beta \in \{25,30,35,40,45,50,55\}$	1. $CR > CR_{MA}$ 2. $CR < CR_{MA}$ 3. $CR > \alpha$ 4. $CR < \beta$ 基因總數：181

13	累積/發散指標 (Accumulation Distribution Line) (相依於 MA)	$AD = \text{Last AD} + \frac{(2C - H - L)}{(H - L)} \times V,$ <p>若無 Last AD 就填 0，移動平均自第三日開始。</p> <p>如果 <math>H = L</math>，則 <math>AD = \text{Last AD} + \frac{(C - \text{Last C})}{\text{Last C}} \times V</math></p> $ADG_N = \frac{(\text{當日 } ADMA_N - N \text{ 日前 } ADMA_N)}{(\text{當日 } MAN - N \text{ 日前 } MAN)},$ <p>其中，<math>ADMA_N</math> 代表 AD 的 N 日移動平均，  <math>MAN</math> 代表收盤價的 N 日移動平均。</p>	V：成交量 H：最高價 L：最低價 C：收盤價 Last AD：前一日 AD	short, long, N $\in \{3, 5, 10, 15, 20, 30, 40, 50, 60, 80, 100, 120\}$	<ol style="list-style-type: none"> <li><math>ADMA_{short} &gt; ADMA_{long}</math></li> <li><math>ADMA_{short} &lt; ADMA_{long}</math></li> <li><math>ADG_N &gt; 0</math></li> <li><math>ADG_N &lt; 0</math></li> </ol> 基因總數：156
14	人氣指標(AR)	$AR_N = \frac{(N \text{ 日內}(H - O)\text{-之和})}{(N \text{ 日內}(O - L)\text{-之和})} \times 100$ <p>若分母為 0，<math>AR_N</math> 以 200 計。</p>	O：開盤價 H：最高價 L：最低價	$N \in \{3, 5, 8, 12, 16, 20, 26, 32, 45, 60, 80, 100, 120\}$ $\alpha \in \{130, 140, 150, 160, 170\}$ $\beta \in \{60, 65, 70, 75, 80\}$	<ol style="list-style-type: none"> <li><math>AR_N &gt; \alpha</math></li> <li><math>AR_N &lt; \beta</math></li> </ol> 基因總數：130
15	意願指標(BR) (相依於 AR)	$BR_N = \frac{(N \text{ 日內}(H - \text{Last C})\text{-之和})}{ N \text{ 日內}(\text{Last C} - L)\text{-之和} } \times 100$ <p>若分母為 0，<math>BR_N</math> 以 500 計。</p>	Last C：前一日收盤價 H：最高價 L：最低價	$N \in \{3, 5, 8, 12, 16, 20, 26, 32, 45, 60, 80, 100, 120\}$ $\alpha \in \{360, 380, 400, 420, 440\}$ $\beta \in \{40, 45, 50, 55, 60\}$	<ol style="list-style-type: none"> <li><math>BR_N &gt; \alpha</math></li> <li><math>BR_N &lt; \beta</math></li> <li><math>BR_N &gt; AR_N</math></li> <li><math>BR_N &lt; AR_N</math></li> </ol> 基因總數：286

16	簡易波動指標 (Ease of Movement Value, EMV)	先求當日價格中點， $MM_t = \frac{H_t + L_t}{2}$ 再計算當日價格終點與前日價格中點差值 $MID_t = MM_t - MM_{t-1}$ 計算每單位成交量， $VPU_t = \frac{V_t}{H_t - L_t}$ ， 計算當日 EMV， $EMV_t = \frac{MID_t}{VPU_t}$ 最後計算 EMV 的 N 日移動平均。	t：當日 H：最高價 L：最低價 V：成交量	short, long, N $\in \{1, 3, 5, 8, 12, 16, 20, 26, 32, 45, 60, 80, 100, 120\}$	1. $EMV_N > 0$ 2. $EMV_N < 0$ 3. $EMV_{short} > EMV_{long}$ 4. $EMV_{short} < EMV_{long}$ 基因總數：196
17	動量震盪指標 (Chande Momentum Oscillator, CMO)	單日上漲/下跌幅度：今日收盤價-前一日收盤價 $CMO_N = \frac{N \text{ 日內上漲總幅度} - N \text{ 日內下跌總幅度}}{N \text{ 日內上漲總幅度} + N \text{ 日內下跌總幅度}}$		$N \in \{3, 5, 8, 12, 16, 20, 26, 32, 45, 60, 80, 100, 120\}$ $\alpha \in \{30, 35, 40, 45, 50, 55, 60\}$	1. $CMO_N > \alpha$ 2. $CMO_N < -\alpha$ 基因總數：91
18	勁道指數 (Force Index, FI)	$FI = V_t * (C_t - C_{t-1})$ 再對 FI 計算 N 日移動平均	t：當日 V：成交量 C：收盤價	$N \in \{3, 5, 8, 12, 16, 20, 26, 32, 45, 60, 80, 100, 120\}$	1. $FI_N > 0$ 2. $FI_N < 0$ 基因總數：13
19	心理線 (Psychological Line, PSY)	$PSY_N = \frac{N \text{ 日內的上漲天數}}{N} * 100\%$		$N \in \{3, 5, 8, 12, 16, 20, 26, 32, 45, 60, 80, 100, 120\}$ $\alpha \in \{0.5, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4\}$ $\beta \in \{0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95\}$	1. $PSY_N < \alpha$ 2. $PSY_N > \beta$ 基因總數：208

總計 K 線型態共 84 種，技術指標共 19 種，技術指標基因狀態共 58 種，基因總數共 5886 種。