基於 Hadoop 雲端運算架構之平行基因演算最佳化學習障礙診斷輔助系統

許晉瑜[1]　許貫傑[2]　高秀婷[3]　吳東光[4]　孟瑛如[5]
[1-4]國立彰化師範大學　資訊管理學系
[5]國立新竹教育大學 特殊教育學系
[1] sw9856@gmail.com、[2] madjump2012@gmail.com、[4] tkwu@im.ncue.edu.tw

摘要

學習障礙的鑑定過程是相當耗費時間與人力的，因此學者們為減低鑑定人員的負荷，使用類神經網路等演算法來開發學習障礙輔助系統，研究結果顯示，透過分散式平行基因演算法搭配類神經網路，可縮短運算時間並提高準確率。

以 MPI 為基礎的分散式平行基因演化存在分散式電腦間的溝通與資料儲存問題。本研究選用雲端運算中完全開放原始碼的 Hadoop 架構作為建構學習障礙輔助系統的平台，透過 HDFS 分散式檔案系統儲存並傳遞資料，並修改平行基因演算法使其對應至 MapReduce 程式框架之上。此外，由於本研究所採用的學習障礙輔助系統資料量較小，因此自行設計一種切割檔案區塊的方式，另外透過 HDFS 作為節點間互相溝通的檔案伺服器，並儲存精英染色體供節點存取。實驗結果顯示，資料量相對較小之情況下，運算速度相較於網格運算並無提升，但當 MapReduce 處理的資料集數量較大時，Hadoop 環境的學習障礙輔助系統其平均學習障礙辨識準確率將會優於網格運算。

關鍵詞：Hadoop、MapReduce、雲端運算、平行基因演算、學習障礙

Abstract

Procedure in the identification of students with Learning Disabilities (LD) requires a lot of time and manpower. In order to reduce the evaluation personnel's workload, researchers use Artificial Neural Network (ANN) to develop an assisted LD students identification system. The results show that through MPI-based Parallel Distributed Genetic Algorithm (PDGA) with ANN, the operation time and identification accuracy may both be improved.

However, the MPI-based solution may have the communication and data storage issues in the distributed environment. In this research, we port the assisted LD students' identification system to the open-source Hadoop framework. We redesign the parallel genetic algorithm and map it to the MapReduce program model, and store corresponding data in the Haddop Data File System. To fit our problem into the Hadoop environment, we design a data file segmentation approach. In addition, we use HDFS to store the elite chromosome, and to act as the communication media among the computation nodes. By comparing results to previous study using MPI-based solution, we have found that our solution may have edge in the case of larger data set.

Keywords: Hadoop 、 MapReduce 、 Cloud Computing、Parallel Genetic Algorithm、Learning Disabilities

1. 緒論

1963 年美國教育學者 Kirk 首度提出學習障礙（Learning Disabilities, LD）一詞，顧名思義即是學習發展上產生異狀，在聽、說、讀、寫、算等方面有較顯著困難，導致在學習過程中產生問題 [15]。學習障礙不同於一般常見的身心障礙，而是屬於一種隱性障礙，學習障礙者幾乎沒有明顯的外貌或外表特徵，常被大眾誤解為不用功的學生，以致於學習障礙者常發生延誤就醫或是從未就醫的狀況，直到 2002 年 5 月 9 日教育部才於『身心障礙及資賦優異學生鑑定標準』的條文中公佈學習障礙定義：「指統稱因神經心理功能異常而顯現出注意、記憶、理解、推理、表達、知覺或知覺動作協調等能力有顯著問題，以致在聽、說、讀、寫、算等學習上有顯著困難者；其障礙並非因感官、智能、情緒等障礙因素或文化刺激不足、教學不當等環境因素所直接造成之結果。」[1] 由於學習障礙在診斷上面十分複雜，目前國內學習障礙鑑定流程大多是以 Discrepancy Model [15]為基礎，該流程主要分為以下四步驟[17]：

初步篩選：大多採用學習行為特徵檢核表（Learning Characteristics Checklist, LCC）作為篩選的工具，此階段通常是由普通班教師、家長、醫護人員、社工人員、學校行政人員或其他與個案相關之人員進行篩選。**成就測驗（Achievement Testing, AT）：**對疑似學習障礙的學生施測成就測驗，更進一步的過濾出在讀、寫、算等基本技能有困難者。

魏氏兒童智力量表第三版（The Wechsler Intelligence Scale for Children-Third Edition, WISC-III）： WISC-III 共分為十三項分測驗，並藉十三項分測驗推算出四個因素指數，三種智商分數，這些分數可做為六歲至十六歲兒童智力鑑定與教學診斷之依據，根據分數未達各年級標準進行評量，篩選出有顯著內在差異的學生，並排除疑似智

能障礙的狀況。**最後以內在能力與心理歷程分析：**由心評人員針對學生內在能力與心理歷程進行分析，來判斷學生是否為學障生。

根據表 1 對於北部地區某縣 91 至 93 學年度學習障礙鑑定人力資源統計表，其 91 學年度學習障礙鑑定所投入之人力包括：第一階段 150 人、第二階段 90 人、第三階段 119 人、第四階段 12 人，這對於各縣市特教單位及特教教師而言，可謂是相當沉重的負擔。為減輕學習障礙鑑定人員的負擔，近年來有學者開始使用人工智慧的方法進行學習障礙鑑定，如類神經網路（Artificial Neural Network，ANN）[3]支援向量機（Support Vector Machine）[4]等分類演算法來進行學習障礙學生診斷之相關研究。

表 1 北部地區某縣91至93學年度學習障礙鑑定耗費之人力資源統計表

階段 / 學年度	校內調查疑似學生	進行校內初步鑑定	鑑輔會複查	最後統整
	人數×天數	人數×天數	人數×天數	人數×天數
91 學年度	25×6	15×6	17×7	3×4
92 學年度	11×2	34×2	14×5	3×4
93 學年度	24×4	20×4	16×6	3×4

資料來源： 吳東光與孟瑛如 [2]

根據過去研究可發現，運用基因演算法（Genetic Algorithms, GA）結合支援向量機，進行資料集屬性之特徵選取，並搭配類神經網路來做預測，可獲得相當不錯的結果，但由於類神經網路在建立的過程相當耗時，因此近年來有學者利用網格運算（Grid Computing）結合多台電腦，加速類神經網路分類模型的建立 [5]，亦有學者利用基因演算法對類神經網路之訓練參數進行校估，並搭配網格運算環境對參數進行分散式平行基因演算（Parallel Distributed Genetic Algorithm, PDGA）搜尋 [6]。

雲端運算（Cloud Computing）與網格運算並沒有嚴格的區隔，兩者皆為平行運算（Parallel Computing）衍伸出來的概念 [7]。網格運算重點在於讓任何伺服器都能加入以提供龐大運算；而雲端運算則是使用大量相同的電腦來執行，在管理非異質性電腦之間的溝通、任務分配及分散式儲存的方法皆較網格運算來得出色。因此，本研究藉以實作出一個以雲端運算為基礎的學習障礙輔助系統，探討基因演化如何運作在雲端運算平台上。

2. 文獻探討

2.1 基因演算法

基因演算法是由 Holland 在 1975 年首度提出，其概念是模擬達爾文進化論中「物競天擇、適者生存」的觀念，創造出一種最佳化搜尋方式。其演化流程如下:

基因編碼:基因編碼之目的在於將所面臨的問題以編碼的方式來呈現，編碼方式可分為「二元編碼」及「實數編碼」兩種。**初始族群:**使用者視需求決定染色體族群的大小，即問題解集合的大小。**適應性函數:**適應性函數為評估染色體適應能力的指標。**複製:**其目的是根據每條染色體的適應值挑選出較佳的染色體進行複製，以提高族群整體的素質。**演化策略:**在基因演算法中，演化分為兩個部份：交配（Crossover）及突變（Mutation），目的即是產生適應力更強的子代。**產生新族群:**經由複製、交配及突變等過程，適應值較差的染色體會被演化過後的新染色體所取代，產生新族群當作下一代，而新族群必須再重覆以上步驟繼續演化，直到滿足終止條件為止。

本研究中適應性函數為使用類神經網路實作的分類模型，其分類準確率即為染色體的適應值。而複製方法本研究將延續以往學者的經驗 [6, 8, 9]使用計算量較少的競爭選擇法。在演化策略中，從交配池中隨機挑選出兩條染色體，並根據使用者事先設定的交配率（Crossover Rate）來決定此兩條染色體是否進行交互相換，在二元編碼的交配方式有單點交配、雙點交配與遮罩交配 [10]。本研究染色體長度僅只有 4 個基因，由類神經網路中的**學習率、動量、隱藏層神經元個數及初始化連結權重相關的亂數種子**所組成，為考量處理效率及基因長度，選擇使用單點交配方式處理較為快速，本研究中基因演化的染色體編碼方式使用實數編碼。在基因值互相交換時，將使用公式 2-1 進行計算[6]。

$$\begin{cases} x_i = a + y_i + (1-a) \times x_i \\ y_i = a \times x_i + (1-a) \times y_i \end{cases} \qquad \text{(公式 2-1)}$$

其中，α 為 0 ~ 1 間的隨機數，x_i 與 y_i 分別為兩條要進行交換的染色體基因值。

突變的使用可幫助演化結果跳脫至其它空間，增加產生適應力更強的染色體。但須預先設立一個門檻值即突變率（Mutation Rate），判斷是否進行突變，當突變率大於隨機機率時，則該基因會產生突變。本研究選用單點突變，選取染色體上的一個基因進行突變，而基因突變時的基因值計算則採用公式 2-2 及公式 2-3。

$$x_j = \begin{cases} x_j + \Delta(t, d_j - x_j) & if \quad z = 0 \\ x_j + \Delta(t, x_j - e_j) & if \quad z = 1 \end{cases} \qquad \text{(公式 2-2)}$$

$$\Delta(t, y) = y \times (1 - r^{(1-t/T)^q}) \qquad \text{(公式 2-3)}$$

其中，d_j 與 e_j 分別為 x_j 的上限值與下限值，z 隨機二元值 0 或 1，r 為 0 ~ 1 之間的隨機亂數，t 為當前的演化世代數，q 為公式 2-2 與演化世代數之相依度。

2.2 Hadoop

「Hadoop」是由 Apache 基金會 Nutch 的創始人 - Doug Cutting 參考 Google 所發表 MapReduce 和 GFS 的公開文件資料，將系統架構實作開發一套與 Google 類似的分散式運算系統軟體平台。Hadoop 雲端運算平台包含三個部份，即 HDFS 分散式檔案系統、MapReduce 分散式運算及 Hbase 分散式資料庫。

1. HDFS 分散式檔案系統

HDFS 全名為 Hadoop Distributed File System ，是一個分散式的檔案系統架構，提供單一的目錄系統(Single Namespace)，能夠有效的處理大量檔案，並提供安全的儲存環境。

HDFS 是採用主從關係 (Master/Slave) 的架構，由 NameNode 和 DataNode 組成，Master 是指一個 NameNode，而 Slave 是多個 DataNode。HDFS 會將檔案切割成相同大小(64MB) 的區塊，DataNode 則以區塊為單位，實際儲存檔案區塊並管理資料內容，每個區塊可以獨立被建立、複製或刪除。而這些區塊為提供容錯的機制，系統會自動將每個區塊複製成自訂的副本數，分散儲存於不同的 Datanode 上。系統預設預設副本數為三份，使用者亦可自訂，本研究使用預設檔案副本數三份。Namenode 主要是管理檔案系統、管理檔案存取的權限或是儲存將切割的檔案存放至哪個 DataNode 的 Metadata，Namenode 本身並不存放資料。Datanode 會執行 Namenode 的要求指令，還會根據使用者對資料的要求進行讀取與寫入，因檔案被分割成多個，並存放在多台不同的機器上，因此可以加快資料的讀取速度。[11, 17]

2. MapReduce 分散式運算

MapReduce 是一種適合處理大量資料的程式模型，撰寫的程式可簡易地同時讓成千上萬台的 Hadoop 叢集同時運算 TB 以上的資料量。開發分散式運算時，最大的問題即為電腦在運算時所產生複雜的溝通問題，因此，此程式模型最大的益處在於程式開發者不需要詳細地了解底層叢集的架構為何，即可開發程式，進而大大減輕程式開發者開發分散式運算的負擔。MapReduce 其工作流程主要分 Map 階段和 Reduce 階段。藉由 Hadoop 平台所提供的容錯機制，開發人員可不需理會當任務失敗時所產生的錯誤，因為該平台會自動地將發生錯誤的任務重新分配給其他的機器來執行。使用者可以專心定義 Map 和 Reduce 函數的解決方案。Hadoop 透過 MapReduce 作業（Job）將整個任務分成若干個小任務（Task），接著再將分割完若干個小任務交給 Map 處理，這些小任務處理完的結果由 Reduce 統合成最後結果，而最後輸出結果的數量則視 Reduce 數量而定。

3. 研究方法

本研究的主要目的是實作出一個以 Hadoop 雲端運算平台為基礎的學習障礙輔助系統，透過 Hadoop 架構的 HDFS 儲存資料，利用基因演算法搭配類神經網路至 MapReduce 的框架上，並且探討如何將基因演算法對應至 Map 與 Reduce 任務上來縮短運算時間並提高分類準確率。另外，本研究亦與先前研究 [9] 之平行基因演算法做比較，探討學習障礙輔助系統在網格環境與 Hadoop 環境下運行的成效。

3.1 樣本資料與前處理

本研究使用的樣本資料為台灣地區三個縣市特殊教育中心提供之學習障礙鑑定資料做為樣本資料集，各資料集內的資料特徵屬性沿用以往學者 [6] 進行過篩選或整理的屬性組合，其內容整理如表 2。

表 2 樣本資料集內容表

資料集	資料筆數	學習障礙筆數	資料屬性 x 個數
1	652	148	WISC-III x7
2	125	31	WISC-III x6、AT x1
3	159	76	WISC-III x5、AT x2、LCC x3

資料來源：林雅莉 [6]

註: WISC-III x7 包括 3 項智力量表與 4 項因數指數; LCC 為學習行為特徵檢核表;AT 為成就測驗，資料集 2 包含認字、閱讀與數學，資料集 3 包含國文、英文與數學[6]

由於樣本資料集內的屬性值皆不相同，若數據範圍差距過大會導致類神經網路的訓練失真。因此，所有樣本資料皆需先經過正規化的處理，將所有資料的屬性值轉換成 0 至 1 之間的浮點數。而樣本資料實際分類的結果與學習障礙輔助診斷系統預測之結果, 存在四種比較關係，如表 3 所示。其中，True 表示預測分類正確，False 表示預測分類錯誤；而 Positive 表示具學習障礙，Negative 表示不具學習障礙。

表 3 實際分類結果與預測分類結果關係表

預測分類 / 實際分類	學習障礙	非學習障礙
學習障礙	True Positive（TP）	False Positive（FP）
非學習障礙	False Negative（FN）	True Negative（TN）

本研究類神經網路接收學習率、動量、隱藏層神經元個數及與初始化網路連結權重相關的亂數種子使用五等份交叉驗證法（5-Fold Cross-Validation）將樣本資料集平均切割成五份，每次使用其中四份進行類神經網路訓練，再使用剩餘的一份進行分類預測，進而得到一份 CIR 值，重複五次循環後取其平均值即為該次交叉驗證的

CIR 值。類神經網路的績效指標為學習障礙的鑑定準確率(Correct Identification Rate, CIR)，計算公式如下：

$$\textbf{CIR} = \frac{\text{TP} + \text{TN}\,(\text{判斷正確的學障與非學障生})}{\text{TP} + \text{TN} + \text{FP} + \text{FN}\,(\text{資料筆數})}$$

3.2 基於 MapReduce 之平行基因演化模式

MapReduce 程式模型可以讓使用者簡易的開發分散式運算，但有許多應用是無法對應至 MapReduce 中 [12]，因此本研究參考先前學者之研究 [12, 16]，並以 MapReduce 程式模型開發本研究之學習障礙輔助系統。本研究將基因演算法每一世代的演化當成一個獨立的 MapReduce 作業，由 4 個基因以亂數的方式產生輸入檔案當作初始群體，在 Map 與 Reduce 兩個階段，當 MapReduce 接收檔案後，會依據區塊大小自動切割成 M 個檔案片段，若檔案量小於當初設定的區塊大小，則不會做切割的動作。由於本實驗輸入的檔案量非常小，當族群總數為 100 時，檔案大小僅為 11KB，因此檔案在上傳時並不會被自動分割，為解決此情況有兩個方法：**1.設定檔案區塊大小。2.自行透過程式將檔案大小切割成 M 個檔案片段，以上兩種方式皆於實驗一時驗證。**

計算適應性函數為整個實驗過程中最花費時間的階段，且每個染色體計算適應性函數的作業是可獨立計算的，因此本研究將計算適應性函數的步驟交給多台電腦做分散式運算，即 Map 階段；而複製、交配、突變、取代最低適應值之染色體與挑選精英染色體這些步驟，必須運算完族群中所有染色體的適應值後才可以執行，因此本研究將這些步驟安排在 Reduce 階段，其流程如圖 1 所示。

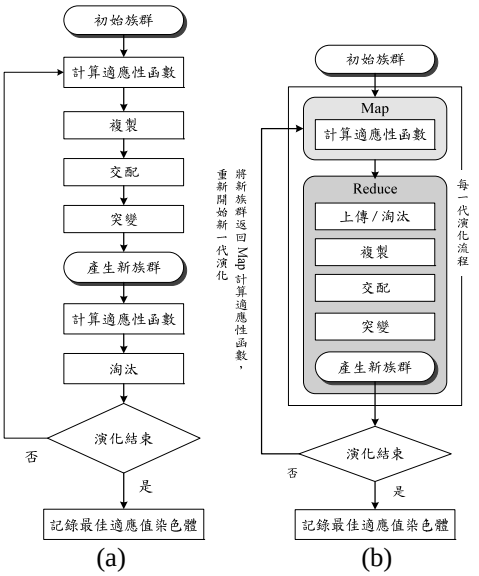

圖 1 (a)一般基因演算法整體運作流程 (b)對映至 MapReduce 之平行基因演算法執行過程

此外，在 HDFS 中的精英染色體對於新一代族群的演化是關鍵的因素，當 HDFS 的染色體適應值越高，則演化出來的染色體就越有機會演化成較佳的適應值，因此本研究實驗二將探討如何配置能讓下一代演化能達到較佳的適應值。

從 HDFS 中隨機取得五個精英染色體後，則取代每一代適應值表現最差的五個染色體，並繼續運算複製、交配與突變步驟，最終產生新一代族群，完成一代 MapReduce 平行基因演算法，每一代 MapReduce 平行基因演算步驟與 HDFS 之示意圖如圖 2 所示

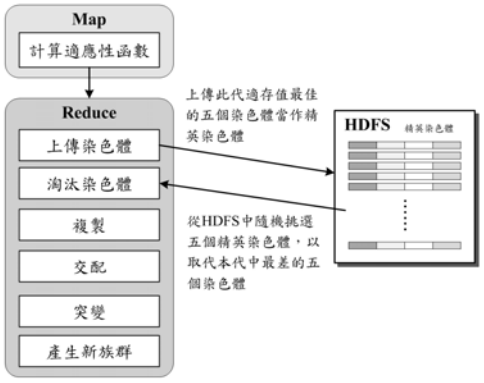

圖 2 MapReduce 平行基因演算法與 HDFS 示意圖

3.3 實驗環境

本研究使用 2 部伺服器主機，以 Ubuntu 10.04 作業系統架設 Hadoop 環境，將 2 部伺服器主機以 KVM 方式分別讓 PC1、PC2 產生 7 台和 5 台虛擬機器，記憶體大小皆為 1GB。伺服器主機規格如表 4。

表 4 伺服器主機規格

編號	處理器型號	核心數	記憶體容量
PC 1	Intel(R)Core(TM) i7	8	12 GB
PC 2	AMD Phenom(tm) II	6	8 GB

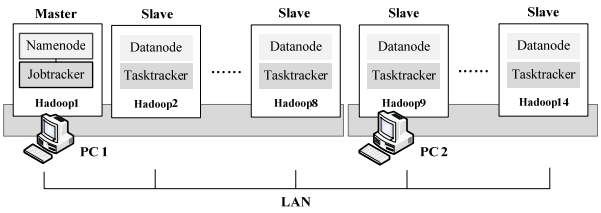

圖 3 Hadoop 電腦角色配置圖

4. 實驗結果與分析

本研究實驗共分為四部份，實驗一至實驗四之實驗結果數據皆為運行二十次之平均值，分別概述實驗設計如下。

4.1 實驗一

在 MapReduce 作業中，分散式運算的單位為檔案片段，一個檔案片段即一個檔案區塊，一個檔案區塊為一個 Map 任務，系統預設切割檔案區塊大小為 64MB，當文件的大小遠遠小於 HDFS 的區塊大

小，即稱為小檔案 [18]。Hadoop 架構中對於小檔案的應用，在啟動作業排程時會產生 Overhead，導致每個工作皆須多花一段時間等待 [14]，而本研究實驗的檔案遠低於 64MB，僅只有幾 KB 而已，因此本實驗設計以手動切割檔案的方式，藉此降低小檔案在 Hadoop 架構上之 Overhead。

實驗採取兩種方式進行比較，方法一為原先 Hadoop 架構之分割檔案片段的方法，透過設定檔案區塊大小，由系統自行切割成檔案片段；方法二由本實驗設計，透過執行程式的方式，手動切割檔案成檔案片段。實驗結果，方法一的運作時間在 400 到 470 秒間，而方法二的運作時間在 58 秒至 72 秒間，由此得知兩種方法在運作時間方面相差約 6.8 倍。本研究建議，當進行 MapReduce 的檔案遠小於預設的 64MB 檔案區塊大小時，透過手動切割可使 MapReduce 整體運作的效能較佳。

4.2 實驗二

本實驗即探討 HDFS 中精英染色體應該如何配置，才可讓下一代演化有較高的機會達到較佳的適應值。本實驗實作兩種不同的方式保留 HDFS 中的精英染色體，方法一為前先研究 [6] 的保留方式，當一代計算完適應值後，上傳最佳的五個染色體作為精英染色體；方法二則為本研究設計的方法，為過濾掉較差的染色體作為精英染色體，因此在上傳前設立門檻，只上傳大於平均值之染色體作為精英染色體。本研究實作以上兩種方法來比較染色體的適應值，並皆運行 20 次取其平均 CIR 值。

在本實驗中得知，若不捨去較差之精英染色體，其 CIR 值則在 83.2 至 84.8 間震盪，反之，其 CIR 值在 84.8 至 86.4 間震盪，其最高之適應值亦可達到 87.58。當精英染色體的 CIR 值皆較高時，有較高的機會演化出較佳的適應值，因此本研究使用方法二的方式保留精英染色體。

4.3 實驗三

本實驗亦與先前研究 [9] 的學習障礙輔助系統進行比較，以下呈現染色體總族群為 100 及 200 時，各資料集分別運行在網格與 Hadoop 環境下的結果，如表 5 所示，在平均 CIR 值方面，當總族群數增加時，其平均 CIR 值亦有遞增的趨勢。在運算時間方面，學習障礙輔助系統在 Hadoop 環境的運算時間皆高於網格運算，但透過表 5 可觀察出當總族群數增加時，其運算時間增加之倍數越來越少 (相較於網格運算總族群數從 100 增加至 200 時，其時間增加之倍數將近為 2)，換句話說，當運算資料量越大時，將越能逐漸凸顯 Hadoop 平台的優勢。

由結果得知，無論學習障礙輔助系統在何種環境下，當族群總體數增加時，其 CIR 值有較大的機會能演化成較好之 CIR 值。特別在資料量最大的資料集 1 中，Hadoop 其 CIR 值都優於網格環境。另外在運算時間部分，Hadoop 環境皆高於網格環境，其主要原因可能為 Reduce 階段尚未平行化所致。

表 5　資料集在 Hadoop 環境與 Grid 環境之比較

資料集	族群數	Hadoop			Grid		
		CIR(%)	運作時間	增加時間倍數	CIR(%)	運作時間	增加時間倍數
1	**100**	**87.89**	3935	-	87.45	2572	-
	200	**88.02**	5600	1.4	87.57	4707	1.8
	300	**88.05**	6809	1.2	-		
2	**100**	85.29	2225	-	85.4	919	-
	200	85.96	2763	1.2	86.6	1599	1.7
	300	86.4	3128	1.1	-		
3	**100**	86.62	3581	-	87.1	1301	-
	200	86.85	4974	1.4	87.2	2533	1.9
	300	87.2	5889	1.2	-		

4.4 實驗四

實驗(a)：為了觀察染色體族群大小與演化節點個數的關係，實驗時設定每個節點皆運算 20 個族群數，當節點數為 1、2、4、8、12 時，其族群數亦為 20、40、80、160 與 240 個，其三個資料集之運算結果如圖 4 所示。可發現，當染色體總族群數增加時，平均 CIR 值亦會有增加的趨勢。

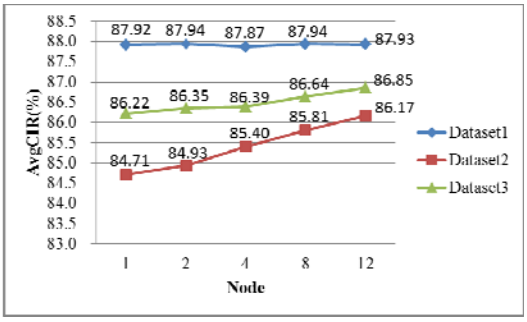

圖 4　實驗四(a)各資料集分別運行之平均 CIR

實驗(b)：為了觀察當染色體固定總族群數時，其演化節點個數與染色體間的關係，實驗將固定總族群數設定為 200，而節點個數設定為 1、2、4、8 與 12 個本實驗使用三個資料集，並分別運行 20 次，如圖 5 所示。

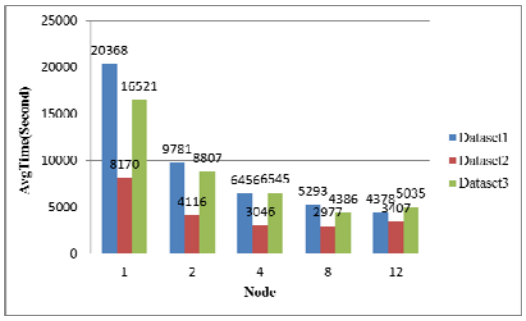

圖 5　實驗四(b)各資料集分別運行之平均運行時間

在平均運算時間方面，可發現當總族群數為 200 時，每個資料集皆有各自最適合的 Map 數量。

在筆數最多的 Dataset1，圖中顯示節點為 12 時其運算時間最短，可得知此資料集最適合的 Map 數為 12 個。因此本研究建議在 MapReduce 作業時，當每個 Map 作業處理每筆 Mapper 的時間需較久時，應設定較多個 Map 數量，透過此設定即可達到較佳的 MapReduce 性能。

先前研究 [9] 顯示當總族群數固定時，投入更多的節點可縮短計算時間，但每個節點演化時需保持一定的染色體規模，否則會導致整體演化陷入區域最大或最小化的問題；但從本研究實驗結果顯示，學習障礙輔助系統在 MapReduce 平行基因演算法中可避免此問題；可能原因在於學習障礙輔助系統在網格環境下，每個節點皆需進行基因演化的所有步驟，直至達到收斂標準為止，因此當每個節點處理的族群數少於 20 時，會導致前述整體演化陷入區域最大或最小化之情況；但在我們的 MapReduce 平行運算實作中，分散式運算處理的運算僅只於計算適應值的步驟，基因演化的後續運作皆交由一個 Reduce 作業統一運算，因此並不會因為每個節點處理的資料量規模小就導致整體演化陷入區域最大或最小化。

5. 結論與建議

本研究使用雲端運算平台的 Hadoop 架構來實作學習障礙輔助系統，在分散式運算方面，設計平行基因演算法使其對應至 MapReduce 程式框架上。研究發現當欲進行 MapReduce 作業的檔案遠小於系統預設的檔案區塊大小 64MB 時，自行透過程式將檔案切割為檔案區塊，亦可使 Hadoop 架構中的 MapReduce 整體運作的效能較佳。此外，在 HDFS 分散式檔案系統的精英染色體應設立門檻值，藉以捨去適應值較差的染色體，如此亦可演化出較佳的適應值。當總族群數呈現遞增時，其學習障礙辨識率亦呈現遞增；但當總族群數增加時，可發現每一個 Map 作業的負擔便因此加重，此時增加 Slave 的數量會使 Map 作業的運算速度加快，但過多的 Map 作業卻會導致 Reduce 作業時間加長，因此設定適當的 Map 數量，有益於提升 MapReduce 效能。

本研究亦與先前網格運算的學習障礙系統比較，發現當總族群數較小時，使用網格運算較 Hadoop 架構的效能高；但當資料量多與總族群越來越大時，本研究推論採用 Hadoop 架構的學習障礙輔助系統準確率將高於網格運算。由於本研究首次設計並實作平行基因演算法至 MapReduce 程式框架上，因此無詳細探索有關 Hadoop 及 MapReduce 效能優化的方法，未來研究可朝向優化 Hadoop 及 MapReduce 效能方面著手，如透過調整 Hadoop 參數進行優化。

參考文獻

[1] 全國法規資料庫(2006)。身心障礙及資賦優異學生鑑定標準。(http://law.moj.gov.tw/LawClass/LawSingle.aspx?Pcode=H0080065&FLNO=10)

[2] 吳東光、孟瑛如（2007）。資訊科技於輔助特教診斷暨支援特教行政與教學之應用。教育資料與研究，78 期，頁 205-226。

[3] 翟鴻榮（2007）。類神經網路搭配委員會機器於輔助學習障礙鑑定之研究。國立彰化師範大學資訊管理學研究所碩士論文。

[4] 吳明豐(2007)。應用基因演算法於優化 SVM 分類器模型－以學習障礙學生鑑定為例，國立彰化師範大學資訊管理學研究所碩士論文。

[5] 張文鴻（2008）。利用網格運算提升以類神經網路為基礎之學障輔助診斷系統準確度與效能。國立彰化師範大學資訊管理學研究所碩士論文。

[6] 林雅莉（2009）。應用分散式演化運算於提升類神經網路分類準確率之研究－以學習障礙鑑定為例。國立彰化師範大學資訊管理學研究所碩士論文。

[7] Mr. Saturday（2008）。Cloud Computing 雲端運算。(http://mmdays.com/2008/02/14/cloud-computing/)

[8] 許育誠（2009)。網格運算環境中工作分配方式對整體運算效能之影響-以學障輔助診斷系統之建構為例。國立彰化師範大學資訊管理學研究所碩士論文。

[9] 張旭（2010）。虛擬化對於平行基因演算程式之影響-以學習障礙輔助診斷系統為例。國立彰化師範大學數位內容科技與管理研究所碩士論文。

[10] 蘇木春、張孝德（2004）。機器學習：類神經網路、模糊系統以及基因演算法則(修訂二版)。全華圖書。

[11] 趨勢科技研發實驗室(2009)。Hadoop 與 Hadoop 分散式檔案系統。民 100 年 4 月 10 日，取自：(http://www.slideshare.net/trendfd/zh-tw-introduction-to-hadoop-and-hdfs zh_TW_Introduction_to_Hadoop and HDFS.pdf)

[12] Chao, J., Vecchiola, C., & Buyya, R. (2008). MRPGA: An Extension of MapReduce for Parallelizing Genetic Algorithms. Paper presented at the eScience, 2008. eScience '08. IEEE Fourth International Conference on, 214-221.

[13] Kirk, S. A. (1963). Behavioral diagnosis and remediation of learning disabilities. In Anonymous, Proceedings of the conference on exploration into problems of the perceptually handicapped child. Chicago: Perceptually Handicapped Children.

[14] Luo, M., & Yokota, H. (2010). Comparing Hadoop and Fat-Btree based access method for small file I/O applications. Paper presented at the Proceedings of the 11th international conference on Web-age information management.

[15] Schrag, J. A. (2000). Discrepancy approaches for identifying learning disabilities. National Association of State Directors of Special Education.

[16] Verma, A., Llora, X., Goldberg, D. E., & Campbell, R. H. (2009). Scaling Genetic Algorithms Using MapReduce. Paper presented at the Intelligent Systems Design and Applications, 2009. ISDA '09. Ninth International Conference on, 13-18.

[17] Wu, T. K., Huang, S. C., & Meng, Y. R. (2008). Evaluation of ANN and SVM classifiers as predictors to the diagnosis of students with learning disabilities. Expert Systems with Applications, 34(3), 1846-1856.

[18] White, T. (2009). Hadoop: The Definitive Guide. O' Reilly Media, Inc.