

雲端計算環境中使用粒子群演算法優化參加校園活動排程系統

尹邦嚴

國立暨南國際大學

pyyin@nunu.edu.tw

林聖婷

國立暨南國際大學

kelly2xxx@gmail.com

摘要

隨著智慧型手機的普及，手機應用程式(App)被靈活運用在許多領域上，開啟了雲端計算的新時代。本研究提出一種以雲端計算為架構的開放式校園 App Cloud 環境，鼓勵教職員工生以使用者角度開發他們所喜歡的校園 Android App，並上傳 App 到校園 App Cloud，校園內的每一個成員都可以到此環境下載與生活學習相關的 App，並利用平台的下載次數推薦最實用的 App。透過平台的設計機制，學校也可節省可觀的資訊系統購置及維護費，回收實用的 App，繼續維護並永續使用。

本研究在上述雲端計算架構中實作一個透過粒子群演算法最佳化個人最佳參加學校活動排程系統，使用者可以設定優化目標及限制條件，例如根據使用者過去參加活動的紀錄，找出某時段最感興趣的活動排程建議，並透過此實例證明此雲端計算架構的運在校園推動的可行性。本研究同時以基因演算法及粒子群演算法應用在排程校園活動的最佳化問題上，實驗結果顯示粒子群演算法求得的解較基因演算法優異。

關鍵詞：雲端計算、智慧型手機、手機應用程式、基因演算法、粒子群演算法、最佳化

雲端計算環境中使用粒子群演算法優化參加校園活動排程系統

壹、簡介(Introduction)

在各大專院校的各類校園資訊系統中，大多是以特定目的為導向的平台，開發的方式不外乎兩類：外包給資訊平台開發公司或由校內資訊人員自行開發。不論是以那一種方式，都需要編列預算，長期的建置與維護，才能持續提供優質的服務。以國立暨南國際大學為例，校園內的平台即有：暨大校園網頁、暨大教務系統、暨大徵才資訊系統、暨大租屋資訊系統、暨大活動報名系統等等(如圖 1)。

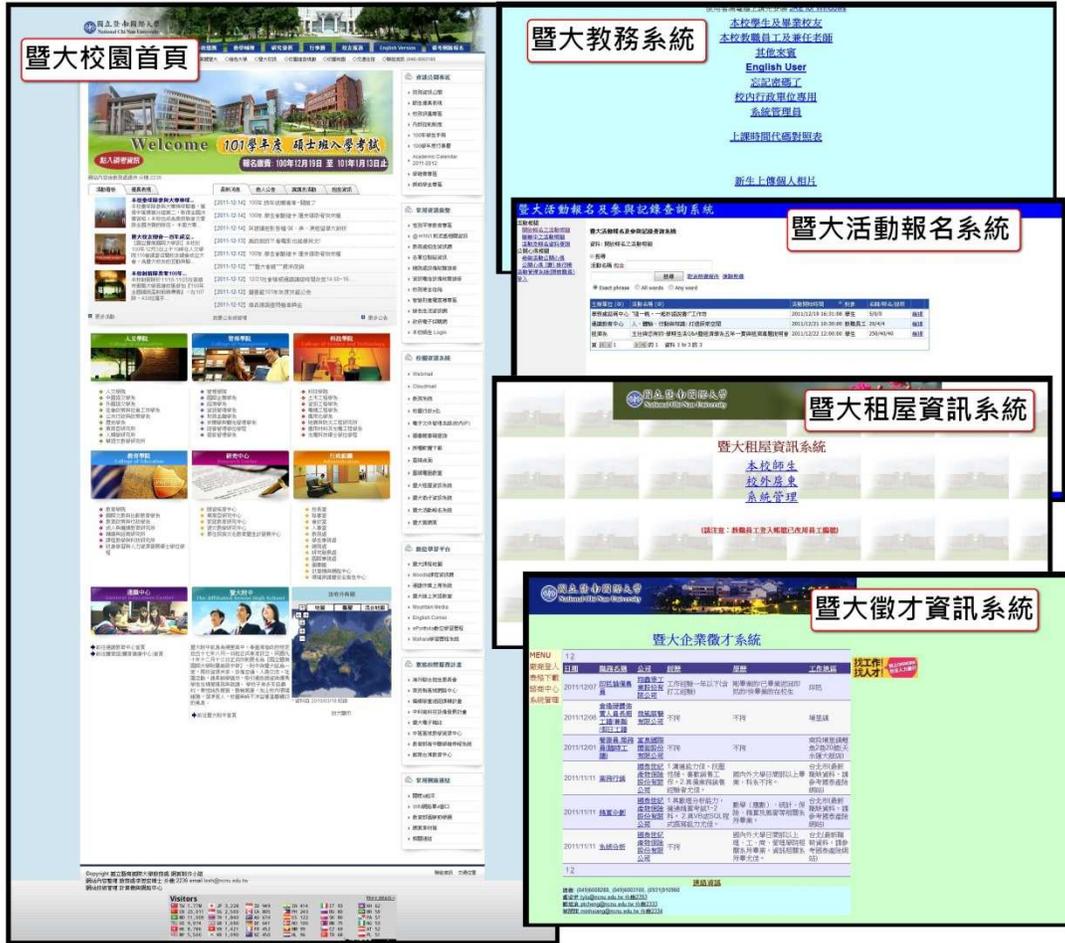


圖 1 暨大部份校園資訊系統

由於網際網路應用的快速發展使得雲端計算概念盛行，加上智慧型手機的日益普及，多樣目的功能的手機應用程式(App)相繼產出，越來越多人有能力開發自己所期望功能的 App，來實現他們對於手機功能的需求，使得系統服務更貼近行動化與客製化。另一方面，使用者透過平常的系統使用經驗，更加深入了解需要的服務內容及提供方式，由使用者自行開發 App 並由其它使用者來評比實用性的模式(Prosumer)，創造出一

種新經濟。本研究探討在校園環境中，開發一個雲端環境平台式服務(Platform as a Service)，讓校園內的每一個成員都可以開發自己認為實用的 App，集中上傳到 NCNU App Cloud 手機應用程式資源平台中，豐富校園手機應用程式資源。

現有的校園系統因目的導向不同，於是一個校園內發展出多種不同功能的平台，分散在不同主機上，管理不易之外，對於使用者來說，必須從不同入口處去尋找他們所需要的平台，十分不便。本研究以國立暨南國際大學為例，建構一個暨大校園 App 雲(NCNU App Cloud)，讓校園中的教職員工生可以自由開發 Android App，並上傳到此校園 App 雲中，鼓勵師生共同參與學校的事物，並以使用者的角度來開發他們所需要的 App，突破以往學校系統皆是由校方資訊人員開發，不僅 App 更符合使用者期待，此空間更名為校園帶來新的服務選擇以及使用方向。此外，我們在此平台設計 App 下載排名機制，因而可以得到推薦較多人下載的 App 且較具實用性質的 App，透過這樣的機制，學校也可節省可觀的資訊系統購置及維護費。

本研究以參加校園活動排程推薦系統 App 為例，提出一個使用粒子群演算法的參加校園活動個人化最佳排程的 App，使其運行於 NCNU App Cloud 中，除了確實可以過濾使用者的排程條件取得某日最佳活動排程之外，我們亦透過此例實證雲端計算應用於開放式校園 App 平台雲的可行性，也透過實驗比較粒子群演算法與基因演算法的效能，顯示前者更適用於參加校園活動個人最佳化排成問題，所求得的活動建議排程最接近最佳解，計算時間也比基因演算法更快速。

貳、文獻探討(Literature Review)

一、雲端計算

雲端計算一般的說法是，在網路技術剛剛興起的時候，人們畫圖時習慣用一朵雲在表示，雲端計算是一種網際網路運算的方式，共享軟硬體資源和資訊可以按需求提供給電腦和其他裝置。現今，行動裝置的網路頻寬速度正在加速，在不久的未來一定可以實現好幾倍的無線通訊速度，可能某天就可以達到與現今 ADSL 等固網同等速度的傳輸環境。雲端計算的介面目前是以執行完整瀏覽器的 PC 為主流，但是今後會逐漸轉變為個人用的手機、平板電腦、PDA 等行動裝置。

雲端計算的定義中，若按服務類型可分為三類(鄧瑋敦 民國 99)，第一類是『基礎設施雲 Infrastructure as a Service (IaaS)』，此層級中是以「服務」的形式提供基礎架構(虛擬機器或儲存設備)。第二類為『平台雲 Platform as a Service (PaaS)』，此層級中是以「服務」的形式提供平台。第三類是『軟體雲 Software as a Service (SaaS)』，此層級中是以「服務」的形式提供軟體。

本研究規劃一個雲端框架，稱作『暨大校園 App 平台雲(NCNU App Cloud)』，在此框架中我們集合了所有跟學校相關的 Android App，提供生活在此校園中的任何人使用跟學校相關的 Android App，更加凝聚在此校園生活的所有人的凝聚力、對事件關心的聚焦力。於是這樣的雲框架符合雲端服務類型定義的第二種『平台雲』，使用者可以將他們所開發的應用程式托管到平台中，這個層級中，應用程式的開發和佈署必須遵守特

定的規則和限制，例如程式語言、程式設計框架、數據儲存模型等(陳滢 民國 99)。本研究之暨大平台雲(NCNU App Cloud)所規定的限制就是只限 Android App 的應用程式上傳，而這樣子的一個手機應用程式平台雲可提供服務給所有的暨大校園成員使用。

二、智慧型手機

智慧型行動電話技術(王英育 民國 93)以「應用邏輯」來區分的話，可分為「Voice Phone」、「Feature Phone」，以及「Smart Phone」。Smart Phone 與 Feature Phone 的差異在於前者內建開放式作業系統能讓使用者自行新增應用程式，後者則是由廠商自行開發封閉式專屬作業系統，而本研究鎖定的 Android 即是開放式手機作業系統，任何人皆可開發 Android App 新增應用在 Android 手機上運行。

簡單的說，「智慧型手機」就是一台可以隨意安裝和移除應用軟體(App)的手機，就像電腦一樣。而「功能手機」是被手機廠限制平台，不能隨意安裝和移除軟體的。因此智慧型手機可以定義為：擁有開放系統環境，並允許第三方自行研發 App，並提供使用者自行運用的手機(陳宜伶 民國 95)。

現今智慧型手機的兩大家作業系統是 Apple 公司的 iOS 和 Google 公司的 Android，是目前使用人數最多的智慧型手機系統，兩家各有優缺點。根據 NetMarketshare (<http://www.netmarketshare.com/>)指出，截至 2012 年 3 月手機作業系統的市佔率，iOS 系統佔 61%，Android 系統佔 19%(見圖 2)，以開發手機應用程式開發條件而言(何孟翰 民國 99; 何孟翰 民國 100; 蓋索林(gasolin) 民國 100)，兩種手機作業系統之開發手比較見表 1。

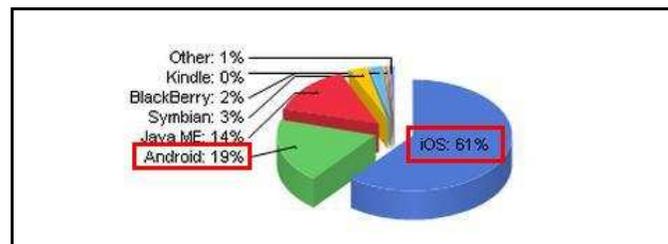


圖 2 NetMarketshare 公布之 2012 年 3 月手機作業系統市佔率示意圖

表 1 開發 iOS 與 Android 之比較圖

	Android (Google)	iOS (Apple)
開發語言	Java	Objective-C, Cocoa
開發工具	Eclipse	XCode
作業系統	不拘(Windows, MacOSX, Linux)	MacOSX
加入開發者身份	終身 25 美金	每年 99 美金
實機測試	無條件	必須是開發者身份
軟體上架審核	無(幾乎)	必要
軟體上架時間	快速	最快一周左右，審核不通過得重新排隊送審。

本研究主要以開發條件難易來評估，決定以 Android 系統 App 為主，將 NCNU App Cloud 設定為只允許 Android App 上傳的平台，一方面 Android 系統已經有廣大的使用人數了，但最主要的考量是 Android App 較為容易上手開發、測試與使用。

三、國內之校園活動系統調查

以國內各大學來說，普遍都存在多個獨立的系統，例如：學校官方網站、選課系統、會計系統、活動公告系統、博碩士論文上傳系統、數位學習平台等。多數校園的選課系統、博碩士論文上傳系統以及會計系統等，因牽涉到學生資料安全問題還有連線人數上線問題，幾乎都是委外資訊公司開發，當然也有校園自己的資訊人員所開發，例如暨南大學與逢甲大學。雖然不是外包，但是各資訊子系統皆為獨立操作，缺乏整合性介面。

目前已經蠻多學校漸漸將一些系統導入雲端校園的概念，像是雲端信箱、雲端桌面、雲端電腦教室等。於是本研究也結合雲端計算概念，建置一個平台可提供所有使用者開發或分享 Android App，並規劃將這些 App 資源集中於某平台中，讓尋找資源能有效率。又以校園系統開放性開發角度來說，某些系統可能牽涉到資安問題，不適合一開始就開放給全校師生開發，本研究以校園活動排程來做問題開發，問題內容不僅可以讓使用者發揮創意，問題本身也比較沒有涉及到資安問題。

許多學校已經有校園活動公告系統，不過這些系統並未特別以使用者角度做最佳活動排程規劃，也並沒有發展 Android 系統介面，但其實此校園活動跟全校師生息息相關，當使用者臨時想查詢或參與活動，可能並不見得身處 PC 環境中，但其實現在科技發達，多數人已經持有手持通信裝備可以隨時隨地上網查詢，雖然透過手持通訊裝備上網也可以查得活動資訊，但本研究希望更進一步開發 Android 應用程式，方便校園使用者更方便查詢或規劃校園活動。

四、次經驗演算法

次經驗演算法近來被廣泛的應用到各種複雜的問題，像是生物系統模擬、作業研究、組合最佳化的 NP-Complete 問題等，使用次經驗演算法就可以在合理時間內得到近似最佳解，是非常有效率的計算方式來解決複雜困難的問題。經典的次經驗演算法例如基因演算法(Chen et. al. 2005)、粒子族群最佳化(Yin et. al. 2006)、螞蟻演算法(Yin et. al. 2006)、模擬退火演算法(Bryan et. al. 2006)、禁忌搜尋演算法(Lee et. al. 2000)等。

本研究之作一個手機應用程式 App，應用粒子群演算法(Particle Swarm Optimization)，優化參加學校活動個人最佳排程，使用者可以設定優化目標及限制條件，例如根據使用者過去參加活動的紀錄，找出某時段最感興趣的活動排程建議。

參、研究方法(Research Methods)

一、系統架構

本研究設計一個暨大校園 App 平台雲(NCNU App Cloud)，提供給在暨大生活的師生上傳 Android App，在這個框架中，使用者可以按照自己所希望的 App，自由設計開

發 Android App，並上傳到此平台中，此平台設計概念類似 Android Market，提供給校園中的所有人下載 Android App 使用，此 NCNU App Cloud 設計架構呈現如圖 3。



圖 3 暨大校園 App 平台雲(NCNU App Cloud) 設計架構

二、個人化活動排程最佳化

本研究在暨大校園 App 雲平台，設計一個實例的 App，但舉凡教務系統、會計系統諸如此類系統涉及資料安全問題，如何開放使用者設計必須有更多的分析與設計。因此，本研究先以一個資料安全層級較低的參加校園活動排程系統為例，讓使用者自由發揮設計之創意，也實證本研究構想之可行性。

目前校園活動之相關系統，在國立暨南國際大學是以公告活動的方式發佈，如圖 4，國內大多數大專院校皆是如此。除了不是從使用者角度開發之外，也沒有參加活動排程的推薦系統。

本研究將粒子群演算法(PSO)應用於尋找使用者在校園中的最佳活動安排建議，程式實作於 Web App Server 以及 Database Server，讓使用者可以透過任何 web 介面來存取 App 結果，亦可透過 Android App 介面來操作，圖 5 為此手機應用程式(App)之設計架構。

更新日期	類別	單位	首頁等級	人氣	標題
2012/3/17	活動資訊	校首頁	普通	335	101學年大學儲備軍官訓練團(ROT)甄選招生中
2012/3/17	活動資訊	校首頁	普通	779	2012 Foreign Student Admission Brochure Announcement
2012/3/17	活動資訊	校首頁	普通	1484	香港教育大學2012學年(秋季)交換學生計畫
2012/3/17	最新公告	校首頁	普通	569	2012年度韓國政府海外國人研究所獎學金
2012/3/14	活動資訊	校首頁	普通	502	放榜囉~~101年大專程度義務役預備軍官預備士官考選成績於3/8日公告
2012/3/14	活動資訊	校首頁	普通	982	公告本校101學年度碩士班暨碩士在職專班入學考試有口試系所錄取名單
2012/1/6	活動資訊	校首頁	普通	991	2012 Foreign Student Admission Brochure Announcement
2012/3/17	活動資訊	校首頁	普通	250	公告2012外國學生申請入學簡章-中、英文版
2012/1/2	活動資訊	校首頁	普通	37	公告2012外國學生申請入學簡章-中、英文版

圖 4 目前暨大校園活動公告形式

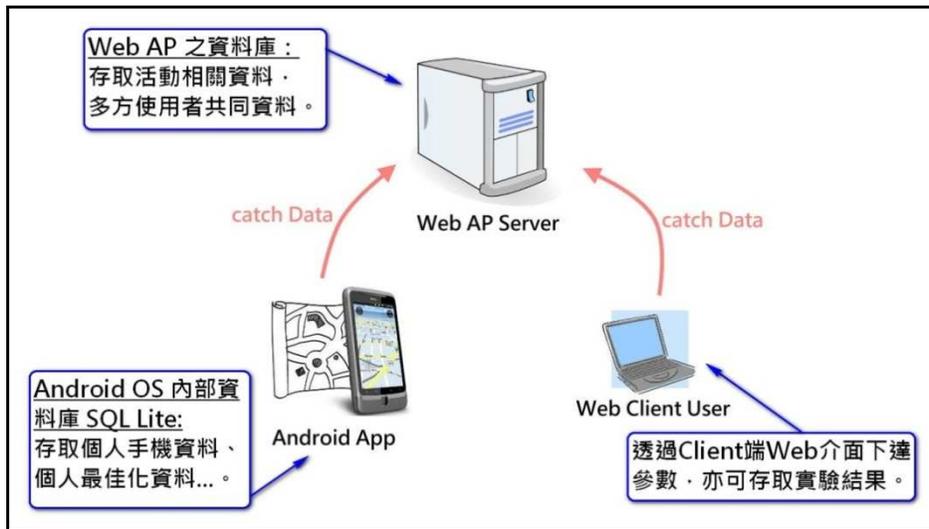


圖 5 活動最佳化之設計架構

本研究提出兩種校園活動最佳化的客製化使用者查詢模式，在最佳化模式一中，針對使用者參加有興趣的活動，整體喜好程度越高越好，如下所示：

$$\text{Maximize: } \quad \text{Max } \sum_{i=1}^n X_i W_i \quad (1)$$

$$\text{Subject to: } \quad X_i X_j |t_i - t_j| \geq t, \forall i, j \quad (2)$$

$$X_i X_j |t_i - t_j| \leq T, \forall i, j \quad (3)$$

其中，我們將 X_i 用來表示使用者是否要參加，1 表示要參加，0 表示不參加。 W_i 則是表示使用者的喜好在第 i 個活動的權重，在這邊的權重是針對活動的類別。本研究以使用者曾參與過的活動類別來計算喜好類別之權重比重。假設使用者未曾參與過的活動類型之活動，我們給予這個活動喜好分數為使用者參加過最少類型比率 $\times 0.7$ 當做活動類型喜好機制之加分分數，若該活動為使用者曾參與過的活動類型，我們給予這個活動分數為該類型參與比率當做活動類型喜好機制之加分分數。若使用者沒有任何活動參與記錄，則適合度分數計算不加入活動類型活動類型喜好分數。

目標式(1)中，將編碼過後每個要參加的活動乘以使用者的喜好程度加總，加總之後的分數，越大表示該活動組合越好。在限制式(2)中，針對所有要參加的活動，前後安排的活動起始時間差需大於等於一個使用者設定的常數 t ，該使用者才能有充份的時間到下一個活動現場。而在限制式(3)中，則是表示活動的最早起始時間與活動的最後活動的結束時間的時間差要小於使用者設定的常數 T 以內，以滿足該使用者的最長參與期間 (duration) 時間規劃。

在最佳化模式二中，使用者可限定參加的活動一定要超過興趣門檻值 d ，這樣的活動參加場次越多高越好，因此數學模式可延用前述最佳化模式一之外，需再新增一個限制式(4)，如下所示：

$$X_i W_i \geq d, \forall i \quad (4)$$

在限制式(4)中，針對每個所要參加的活動，活動類型的喜好程度一定要大於使用者設定的門檻值 d 以上，才可安排該活動。

以上兩個模式下，使用者在系統中先選擇某一天日期及要挑選的活動時間區間，我們即可透過資料庫程式語言過濾出該天時間範圍內所有的活動，假設該時段內總共有 n 個活動可供挑選，一個排程候選解可編碼為 d 長度的二元字串，其中 0 代表「不參加該活動」，1 代表「參加該活動」。因此，以下所述基因演算法及粒子群演算法的母體，都是以此候選解編碼方式來產生。

三、基因演算法

基因演算法的基本流程如圖 6 所示，首先，初始母體包含固定人口數量的染色體，產生方法通常以隨機方式或依據問題特性產生，再來依據問題的目標式(1)來對各個染色體計算其適合度，再根據其適合度大小來挑選合適的染色體，並藉此淘汰不合適者，常見的天擇(Selection)方法有輪盤法(Roulette-wheel)及競爭法(Tournament)，我們實作中則採用輪盤法。之後再對染色體進行交配(Crossover)以及突變(Mutation)的運算來產生新一代的人口，我們實作單點交配(Single-Point Crossover)及位元翻轉突變(Bit-Flip Mutation)，交配率及突變率分別設定為 0.8 及 0.2。然後重複執行前述四個步驟：計算適合度、天擇、交配、突變，直到達到最大演化代數或指定終止條件為止，最後輸出整個演化過程中適合度最高的染色體做為最終解。

在處理不合理的時候，採行給予懲罰分數機制。我們以活動 1 和活動 2 來說明，活動 1 的結束時間與活動 2 開始時間之間的時間差當做一個單位的懲罰分數，當活動與活動的時間相互衝突的時候，給予三個單位的懲罰分數，在適合度分數中扣除。若活動與活動之間的時間差大於使用者所設定的時間時，給予一個單位 + 50 當做懲罰分數，在適合度分數中扣除。

四、粒子群演算法

粒子群演算法流程圖如圖 7 所示，首先，會以隨機的方式產生初始候選解，每個候選解即代表一個粒子(Particle)，如同基因演算法中的染色體。粒子藉由自身的最佳經驗和全體粒子的最佳經驗交互演化，每個粒子都往最佳經驗位置附近靠近，逐步找到問題的近似最佳解。

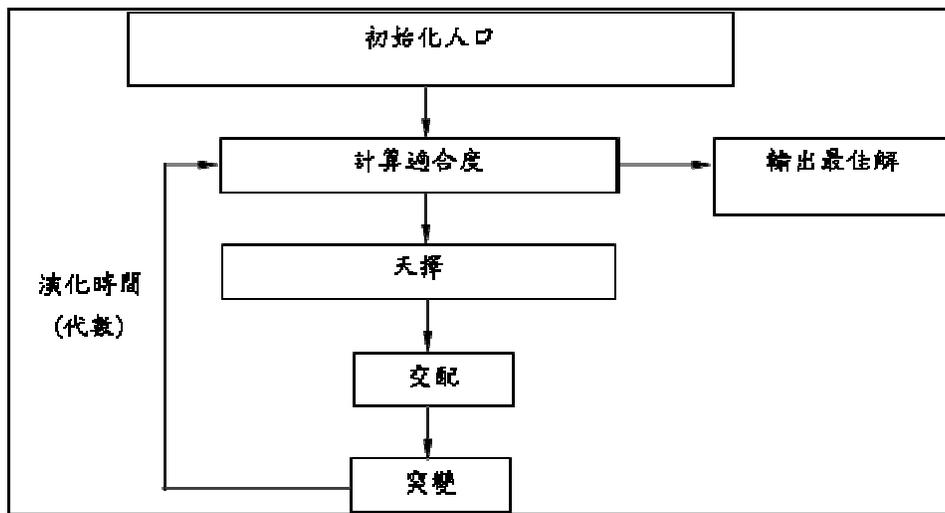


圖 6 基因演算法流程圖

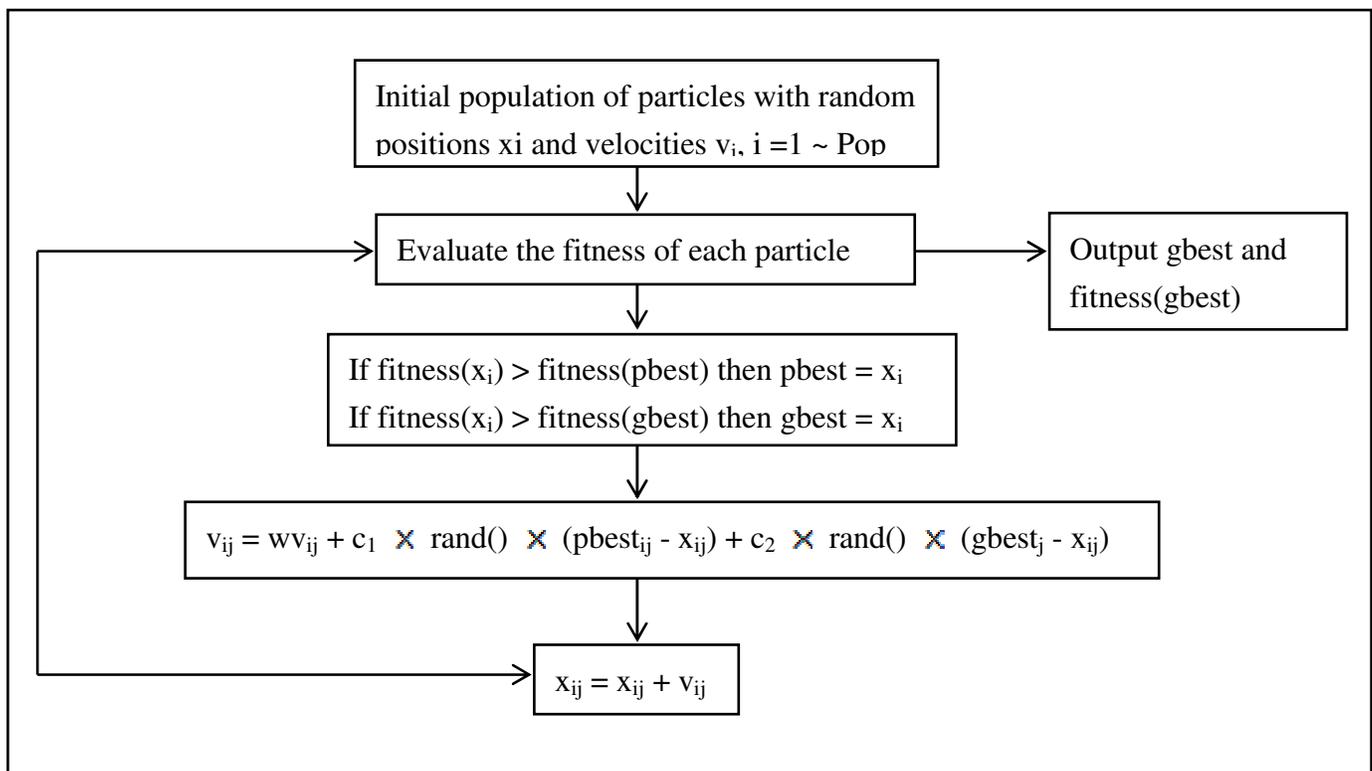


圖 7 粒子群演算法流程圖

圖 7 當中 Pop 為粒子數目， x_i 為粒子 i 所在的位置向量，也就是其編碼形式， V_i 為移動向量。接下來使用適合度評估函數針對每個粒子答案品質好壞做評估，然後設定每個粒子的自身最佳經驗值(pbest)和全部粒子的全體最佳經驗值(gbest)，自身最佳經驗值是每個粒子在逐代演化中，到目前為止所尋找到最好品質的答案，粒子在每一代的演化中，若能找到更佳品質的解，則進行自身最佳經驗值的更新，全體最佳經驗值則為全部粒子的歷史最佳解，此解具有目前最高品質的答案。

w ， c_1 和 c_2 為加權值，控制不同參數的影響程度， $\text{rand}()$ 為介於 0~1 之間的實數變數， v_{ij} 為粒子 i 在第 j 個維度上的位移向量值， v_{ij} 的更新除了參考原本的位移向量慣性之外，同時也參照 $pbest$ 和 $gbest$ 。更新完後再評估新得的解是否有比原本的 $gbest$ 和 $pbest$ 佳，若有則進行更新。

Shi&Eberhart (1998) 在 1998 年另外提出一個位移向量更新的方式，在原本標準的更新方式中加入可動態調整權重值的運用。更新方式如公式(5)所示：

$$w = 0.9 - \frac{0.9-0.4}{\text{iter}_{\max}} \times \text{iter} \quad (5)$$

w 為一慣性因子，用以調整每次粒子移動時的距離大小， w 值會隨著演化代數從最高 0.9 一直降到 0.4， iter_{\max} 為粒子最大演化代數， iter 為當代的演化代數。 w 值由大降到低的目的是讓粒子群在一開始的時候有較大的移動距離，隨著代數的增加，粒子群的移動範圍也縮小，逐漸往最佳解的位置聚集，通常可以避免太早收斂，得到較好的結果。

在處理不合理的時候，同上段基因演算法的懲罰分數機制，計算於粒子群演算法中。

五、介面設計

若使用 web 介面，在 client 端網頁上，輸入本實驗所架設的 server IP 之後，即可透過 web 介面執行本實驗。若使用 Android 介面(如智慧型手機)，則執行我們的參加校園活動最佳化 App 亦可進行本系統的操作，如圖 9 所示。



圖 9 暨大校園最佳化 Android App 介面示意圖

肆、實驗結果(Experimental Results)

一、校園活動最佳化模式一

本實驗透過 web client 端的介面來測試此校園活動最佳化 App 之運算結果是否為最佳解。在活動最佳化模式一中，我們設定以下查詢參數：使用者代號為『kelly』，針對『2011 年 10 月 5 日』去做當日活動的最佳排程，時間區間為早上 8 點到晚上 20 點，前

一個參加活動結束與下一個參加活動開始時間之間隔設定為必須大於 10 分鐘以上，如圖 10。



針對使用者

計算 這一天

活動最佳化安排。

設定想參加活動的時間範圍 ~ 之間。

設定所參加的活動與活動之間，時間 t 需大於 分鐘。

圖 10 使用者改變參數，改查詢 2011 年 10 月 05 日可參與的活動。

第二個問題使用者一樣為『kelly』，活動日期改為『2011 年 10 月 5 日』，活動時間一樣設為早上 8 點到晚上 20 點，活動與活動的時間區間一樣設為『10 分鐘』以上，如圖 14。查詢該時段區間包含 36 筆活動（總共有 2^{36} 種排列組合），如圖 11，我們再透兩種演算法去求最佳活動排程建議。

基因演算法和粒子群演算法求得的活動建議如圖 12，兩種演算法求得出來的解皆為合理解，沒有重複安排的活動，活動與活動之時間間隔皆大於 10 分鐘以上。另外，我們可明顯看出粒子群演算法求得的解適合度遠高於基因演算法所得的解適合度，安排的活動數也較多。

二、校園活動最佳化模式二

我們依照前節兩個問題例子的參數設定在校園活動最佳化模式二做測試，但另外指定喜好活動類型參數值需達 0.5 以上，如圖 13。

由於使用者所設定活動類型的比重一定要大於 0.5，由圖 11 得知，過去使用者所參加過的活動只有活動類別 4 的權重大於 0.5，所以，模式二的實驗結果必須只能挑出活動類型 4 的活動，一樣以排列最多活動為最佳。本研究分別以基因演算法以及粒子群演算法求得最佳化活動排列建議，如圖 14。兩種演算法所求得的建議活動皆屬合理解，同時，粒子群演算法求得的解優於基因演算法。

使用者 **kelly** 參與活動種類之記錄：

活動種類代號	參與活動種類	次數	所佔百分比
2	藝文類	1	0.3333333333
4	休閒類	2	0.6666666667

(符合挑選日期的活動數為：36筆。)

以下為有包含 2011/10/05 的所有活動：

活動序號	活動名稱	活動類別	活動起始時間	活動結束時間
2	活動二	4	20111005080000	20111005083000
20	活動二十	2	20111005080000	20111005090000
24	活動二十三	4	20111005080000	20111005090000
37	活動三十六	9	20111005084900	20111005085100
38	活動三十七	9	20111005084900	20111005095100
22	活動二十二	4	20111005090000	20111005110000
21	活動二十一	4	20111005110000	20111005120000
7	活動七	2	20111005110000	20111005120000
11	活動十一	4	20111005110000	20111005120000
39	活動三十八	4	20111005114900	20111005125000
40	活動三十九	8	20111005114900	20111005125000
41	活動四十	4	20111005114900	20111005125000
31	活動三十	3	20111005120000	20111005130000
42	活動四十一	10	20111005130000	20111005133000
43	活動四十二	7	20111005130100	20111005132100
44	活動四十三	4	20111005140000	20111005145200
15	活動十五	2	20111005140000	20111005160000
10	活動十	4	20111005140000	20111005150000
45	活動四十四	7	20111005141100	20111005145100
19	活動十九	2	20111005150000	20111005160000
14	活動十四	2	20111005150000	20111005160000
47	活動四十六	2	20111005153000	20111005183200
46	活動四十五	4	20111005153300	20111005195100
32	活動三十一	4	20111005161000	20111005164000
49	活動四十八	5	20111005161800	20111005195300
48	活動四十七	4	20111005161900	20111005195000
30	活動二十九	4	20111005170000	20111005180000
9	活動九	5	20111005170000	20111005180000
6	活動六	4	20111005180000	20111005190000
25	活動二十四	3	20111005181500	20111005184500
26	活動二十五	4	20111005183500	20111005191000
27	活動二十六	7	20111005185000	20111005193000
33	活動三十二	2	20111005192000	20111005195000
34	活動三十三	4	20111005192000	20111005193000
35	活動三十四	10	20111005192900	20111005195100
36	活動三十五	10	20111005194900	20111005195100

圖 11 為 2011 年 10 月 05 日可參與的活動比數。

【校園活動最佳化(一) GA WEB版, 建議活動如下:】					【校園活動最佳化(一) PSO WEB版, 建議活動如下:】				
參與排序	活動名稱	活動類別	活動開始時間	活動結束時間	參與排序	活動名稱	活動類別	活動開始時間	活動結束時間
1	活動二	4	0800	0830	1	活動二	4	0800	0830
2	活動二十二	4	0900	1100	2	活動二十二	4	0900	1100
3	活動四十	4	1149	1250	3	活動三十八	4	1149	1250
4	活動四十二	7	1301	1321	4	活動四十一	10	1300	1330
5	活動二十九	4	1700	1800	5	活動四十三	4	1400	1452
6	活動二十五	4	1835	1910	6	活動三十一	4	1610	1640
7	活動三十三	4	1920	1930	7	活動二十九	4	1700	1800
此組解適合度分數 fitness = 5.8666666668					此組解 fitness = 8.1333333332				

圖 12 GA 演算法與 PSO 演算法求得暨大校園活動最佳化模式一之結果畫面。

針對使用者

計算 這一天

活動最佳化安排。

設定想參加活動的時間範圍 ~ 之間。

設定所參加的活動與活動之間的時間 t 需大於 分鐘。

設定喜好的類型活動 d 大於 。

圖 13 活動最佳化模式二參數設定

【校園活動最佳化(二) GA WEB版, 建議活動如下:】					【校園活動最佳化(二) PSO WEB版, 建議活動如下:】				
參與排序	活動名稱	活動類別	活動開始時間	活動結束時間	參與排序	活動名稱	活動類別	活動開始時間	活動結束時間
1	活動二	4	0800	0830	1	活動二十三	4	0800	0900
2	活動三十八	4	1149	1250	2	活動十一	4	1100	1200
3	活動四十三	4	1400	1452	3	活動十	4	1400	1500
4	活動四十七	4	1619	1950	4	活動三十一	4	1610	1640
此組解 fitness = 2.672					5	活動二十九	4	1700	1800
					6	活動二十五	4	1835	1910
					7	活動三十三	4	1920	1930
					此組解 fitness = 4.676				

圖 14 GA 演算法與 PSO 演算法求得暨大校園活動最佳化模式二之結果畫面。

最後，我們進行強健性測試(robustness test)，重複執行 30 次上述兩個最佳化模式，分析其排程品質的穩定度，結果如圖 15 與圖 16。透過觀察得知，兩個最佳化模式在資料量較多(2011/10/05)的時候，粒子群演算法的適合度分數(fitness)明顯平均皆高於基因演算法，且穩定性也優於基因演算法，因此我們證實粒子群演算法比更基因演算法適合用於校園活動排程最佳化問題。

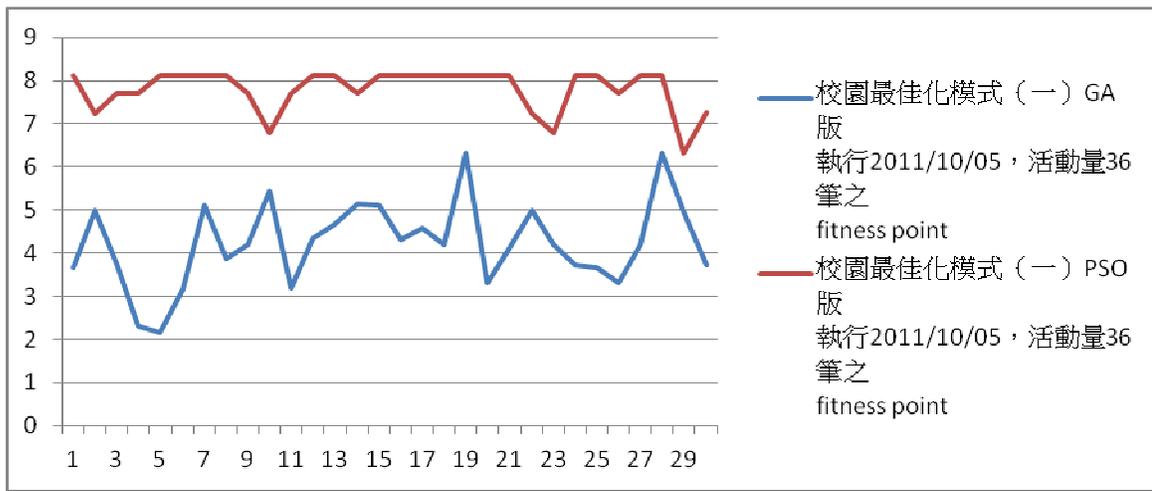


圖 15 校園活動最佳化模式一之強健性分析。

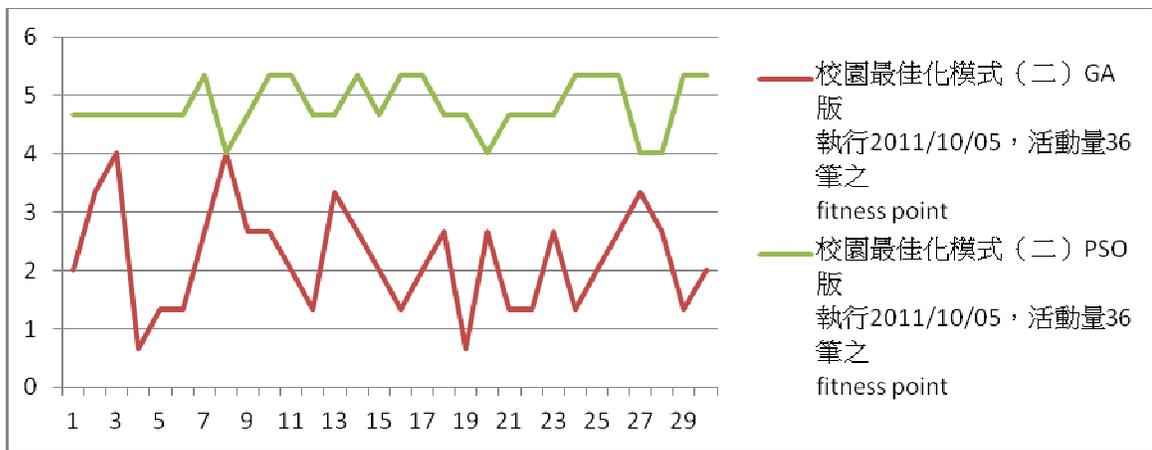


圖 16 校園活動最佳化模式二之強健性分析。

伍、結論(Conclusion)

本研究已展示校園活動最佳化 App 在暨大校園 App 平台雲(NCNU App Cloud)架構的運作狀態，一方面證實本研究所設計的 App 是可行的，另一方面也證實這樣校園 App 平台雲的流程完整性與存在的價值，的確可以幫助所有校園的師生共同參與學校的共同生活事物。

透過基因演算法以及粒子群演算法去應用校園活動最佳化的排列組合問題上，明顯觀察到粒子群演算法優於基因演算法，因此未來更可以將粒子群演算法擴充應用到更多校園雲端系統的應用問題上。

透過暨大校園 App 平台雲的下載次數排名機制，校方更可了解到許多更實用的校園 Android App，在未來行動裝置普及化的時代，此種平台雲服務將會是一種更符合校園使用者需求期待的新選擇。

陸、致謝(Acknowledgement)

本研究部份成果由國科會專題研究計畫(NSC 98-2410-H-260-018-MY3)所贊助。

柒、參考文獻(Reference)

1. 王英育，民國 93 年 12 月，智慧型行動電話發展技術藍圖，工業技術研究院。
2. 何孟翰，民國 99 年 11 月，Eclipse 工具建置 Android 開發環境，台北：Run! PC 雜誌 202 期。
3. 何孟翰，民國 99 年 10 月，iOS 軟體開發流程，台北：Run! PC 雜誌 201 期。
4. 何孟翰，民國 100 年 3 月，前進 Android Market! Google Android SDK 實戰演練，台北：精誠資訊。
5. 陳宜伶，民國 95 年 7 月，智慧型手機與高階相機之消費者行為分析，國立成功大學電信管理研究所碩士論文。
6. 陳澄 等，雲端策略，台北：天下雜誌，民國 99 年 3 月。
7. 蓋索林(gasolin) 編著，民國 100 年 8 月，Google! Android 3 手機應用程式設計入門第四版，台北：松崗資產管理。
8. 鄧瑋敦 譯，民國 99 年 1 月，雲端計算大解密，台北：城邦文化。
9. Bryan, K., Cunningham, P., and Bolshakova, N. "Application of Simulated Annealing to the Biclustering of Gene Expression Data," *Information Technology in Biomedicine, IEEE Transactions on* (10:3) 2006, pp. 519-525.
10. Chen, J. F., Lin, R. L., and Liu, Y. C. "Optimization of an MRT Train Schedule: Reducing Maximum Traction Power by Using Genetic Algorithms," *Power Systems, IEEE Transactions on* (20:3) 2005, pp. 1366-1372.
11. Lee, C. Y., and Kang, H. G. "Cell planning with capacity expansion in mobile communications: a tabu search approach," *Vehicular Technology, IEEE Transactions on* (49:5) 2000, pp. 1678-1691.
12. Shi, Y., and Eberhart, R. "A modified particle swarm optimizer," *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, 1998, pp. 69-73.
13. Yin, P.-Y., and Wang, J.-Y. "Ant colony optimization for the nonlinear resource allocation problem," *Applied Mathematics and Computation* (174:2) 2006a, pp. 1438-1453.
14. Yin, P.-Y., Yu, S.-S., Wang, P.-P., and Wang, Y.-T. "A hybrid particle swarm optimization algorithm for optimal task assignment in distributed systems," *Computer Standards & Interfaces* (28:4) 2006b, pp. 441-450.

Optimal Campus Activity-Attending Scheduling in the Cloud Computing Environment Based on Particle Swarm Optimization

Peng-Yeng Yin

Department of Information Management, NCNU

pyyin@nunu.edu.tw

Sheng-Ting Lin

Department of Information Management, NCNU

kelly2xxx@gmail.com

Abstract

With the popularity of smart mobile phones and many bundled program applications (app), we face a new era of cloud computing. This study proposes a framework based on cloud computing for developing a campus-wide App Cloud environment, encouraging faculties, staffs, and students to develop their favorite campus apps, and to upload their apps to the campus app cloud. The campus members can download these apps and get to learn more about their campus lives. In addition, the members can learn recommended popular apps by reference to the times of the apps downloads.

With the cloud computing framework, this study implements a personalized optimal campus activity-attending scheduling system by a particle swarm optimization algorithm. The users can set the optimization objectives and constraints. For example, the system can recommend the most interesting activity-attending schedule based on the user's past record on activity attendance. With this illustrative example, we empirically prove the applicability of this cloud-computing framework in school communities. The experimental results show that the schedule obtained by the particle swarm optimization is better than that obtained by a genetic algorithm.

Keywords: cloud computing, smart mobile phones, apps, genetic algorithm, particle swarm optimization, optimization