

# Dynamic Elite Artificial Bee Colony Algorithm Applied to Solve the Time-Cost Trade-Off Problem for Software Project

洪燕竹，邱國華

國立嘉義大學 資訊工程學系

Yen-Chu Hung, Guo-Hua Qiu

Computer Science and Information Engineering, National Chiayi University

andrew@mail.ncyu.edu.tw s1030475@mail.ncyu.edu.tw

## 摘 要

現今科技的不斷進步以及變化快速，各家公司為了加快上線的速度，在不增加成本的考量下只能選擇縮短軟體開發及測試的時間，這同時導致了軟體生命週期(Software Development Life Cycle, SDLC)逐漸縮短，對於專案經理(Project Manager, PM)來說，如何利用有限的成本以及有限的時間內提出一套或多套可行的方案，是一個重要的議題。

本論文將軟體專案排程的時間成本問題視為一個時間成本權衡(Time-Cost Trade-Off)問題，以人工蜂群演算法(Artificial Bee Colony, ABC)來解決此時間成本權衡問題。並以找出柏拉圖最佳解前緣(Pareto-optimal front)為目標，提供給專案經理一個較佳的決策空間來進行決策。

本論文分別與傳統的多目標基因演算法(Multi-Objective Genetic Algorithms, MOGA)以及向量評估基因演算法結合禁忌搜尋法(TSVEGA)進行多目標求解能力的評估與比較。

**關鍵字：**多目標最佳化、人工蜂群演算法、時間成本權衡問題、柏拉圖最佳解前緣

## 一、簡介

近年來隨著資訊科技的高速發展，科技公司為了能盡快的在市場當中佔有一席之地，不停推出各式各樣的產品，為了搭配不斷發展的硬體，資訊軟體也不斷更新與創新，如手機軟體、遊戲軟體、作業系統等等。而軟體的複雜度與時效性是專案經理(Project Manager, PM)對於專案開發的重要考量，如何在公司有限的時間與金錢內將此軟體完成，成為專案經理的一大困難。

任何的軟體最初都是從一個模糊想法開始的，從這想法提出的那一刻開始，該軟體就進入了軟體生命週期(Software Development Life Cycle, SDLC)。軟體在歷經了需求、分析、設計、實現、測試、維護，直至軟體缺少維護而逐漸消亡。專案經理的工作就是將這些模糊的想法，轉換成一個明確且可行的軟體專案。並在公司指定的時間與資金內，將資金、時間及人力等資源有目的地分配與協調。同時有效的與專案利害關係人溝通，以期在預設的條件下，完成公司預期的目標。而類似的軟體專案(如公司設計某一款軟體的第二代或第三代等)通常具有重複或類似的任務，當專案的進行，專案經理若能依照作業需要選取較佳的資訊技術，不僅能縮短

任務所耗費的時程，或是減少任務花費的金錢。由於不同的資訊技術有著不同的工作成本，因此在軟體專案的管理與規劃中，時間成本權衡(Time-Cost Trade-Off)是個相當重要的議題[7][8]。透過時間成本分析，專案經理能有效的利用作業時間、花費金錢、不同的設備或測試，使得軟體的專案成本降低或在時程縮短之間取得一個平衡。

早期在解決軟體專案時間成本最佳化問題上大多數都是利用傳統數學理論來計算，在運算過程中產生大量計算的同時，雖可求得最佳解，但遇到複雜的情境時，計算量會大幅增加，導致計算時間大幅增加甚至可能導致系統崩潰無法計算。因此近年來許多學者使用基因演算法[5][6]、禁忌搜尋法[4][7]等演算法來改善傳統計算模式的缺點。

## 二、文獻探討

本論文主要針對多目標解以及人工蜂群演算法來進行探討，以下分為兩部分的小節逐一介紹。

### 2.1 多目標最佳化

多目標最佳化與單目標最佳化的差別在單目標最佳化是根據單一目標函數來尋找一個全域最佳解，而多目標最佳化則需要同時考慮多個目標函數來尋找，而如果這些目標是互相衝突(conflict)的，各目標所對應的全域最佳解就會有所不同，因此，多目標最佳化會找到的是一組最佳解而非單一最佳解。在多目標的可選擇的集合中，存在著一組最佳解，在這一組最佳解中的任意解在全部解中就無法找到另一個解，可使多方效能都增加，或是一方效能不變另一方效能增加，而這一組最佳解內的任意點稱之為柏拉圖最佳解(Pareto-optimal solutions)，而這一組最佳解稱之為柏拉圖最佳解前緣(Pareto-optimal front) [4][9]。

#### 2.1.1 柏拉圖最佳解前緣

假設決策者需要同時考慮「時間」與「成本」這兩個目標，且希望能夠同時滿足最小時間與最小成本的目標，則此兩目標所形成的柏拉圖最佳解前緣如圖 1 所示：

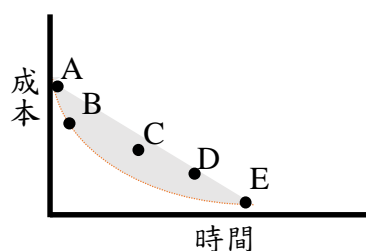


圖 1 柏拉圖最佳解前緣示意圖

圖 1 中灰色區域代表可行解區域，其中，A 點的成本最高，但花費的時間最短；相反的，E 點的成本最低，但花費的時間最長。雖然 A、E 各有一個目標值勝過對方，但因為使用者不清楚兩目標之間的相對重要性為何，故無法比較 A 與 E 的好壞。若在此區域中已經找不到任何點，其兩個值都比 A 或 E 小，則 A 和 C 即為柏拉圖最佳解，還有其他類似的點，例如 B，這些柏拉圖最佳解連在一起所形成的虛線，就是所謂的柏拉圖最佳解前緣(Pareto-optimal front)。

## 2.2 人工群演算法

群體智慧可設計為解決分散式的問題，其靈感來自於鳥類、昆蟲及其他動物的集體行為。群體智慧不同於以往的人工智慧，具有自我組織、分散式控制和適應的能力的特性。群體智慧研究著重設計出生物個體間互動的規則及整體性的行為。演化式演算法為，將群體間的社會行為轉為算術模型，利用算術模型解決複雜工程問題[11]。

在許多的最佳化演算法中，人工蜂群演算法為一種群體智慧概念的仿生演算法，在問題解域中透過迭代運算來搜尋全域最佳解。特色為需要的參數設定較少以及搜索的範圍較廣。藉由搜尋以及探索的步驟，配合蜜蜂特有的搖擺舞，進行大範圍搜索並追尋前次較佳解的雙重行為，可以避免陷入局部最佳解的情況。由文獻可以發現，人工蜂群演算法已經被運用在各種組合最佳化問題上，並且獲得不錯的結果[2][3]。

Karaboga 於 2005 年提出人工蜂群演算法[1]，根據觀察蜜蜂採食花蜜的行為，透過在蜂巢傳遞訊息的方式，使蜂群朝著某一花蜜源或某一目標前進。人工蜂群演算法是利用當前的最佳解，朝著此方向大範圍搜索，藉此達到最佳化目標值，是種具有群體智慧的仿生演算法。

人工蜂群演算法是由蜜源與工蜂(Employed Bees)、觀察蜂(Onlookers)和偵察蜂(Scouts)組成的。蜜源在人工蜂群演算法代表可行的一組解，蜜源透過適應函數計算出花蜜量價值，花蜜量價值為適應值。演算法首先隨機初始化 N 個解(蜜源)，在初始化後，進行工蜂搜尋蜜源、觀察蜂依據蜜源適應值來選擇前往哪一個蜜源來探索，偵察蜂依據蜜源被探索次數來決定放棄該蜜源並搜尋新蜜源，直到滿足最大迭代數，人工蜂群演算法流程圖如圖 2 所

示。

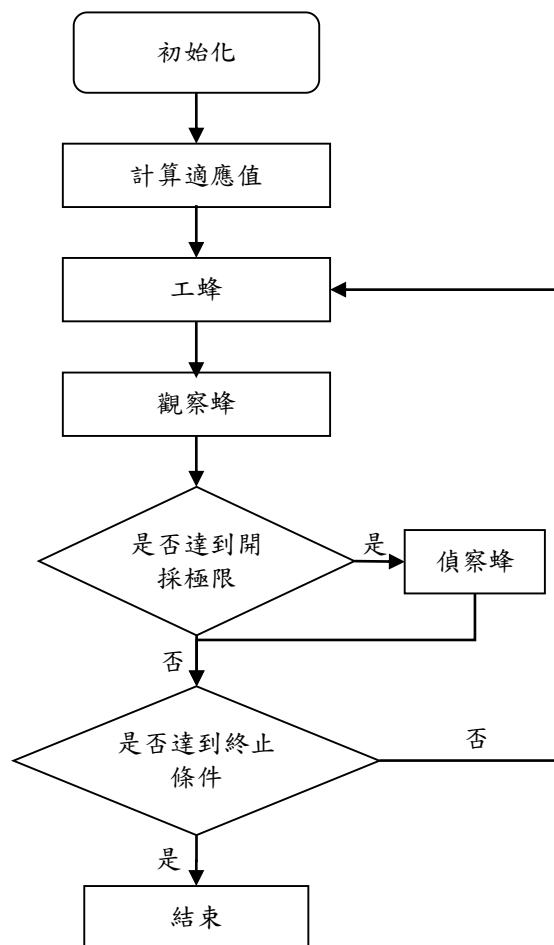


圖 2 人工蜂群演算法流程圖

## 三、動態菁英人工蜂群演算法

本論文利用動態菁英人工蜂群演算法(DEABC)來解決軟體專案時間成本最佳化問題，動態菁英人工蜂群演算法與菁英人工蜂群演算法[10]最大的差異是在於，雖說兩者在演算開始時都不知道適合的菁英族群的數量，但由於動態菁英蜂群演算法中的菁英族群是在演化過程中逐步產生的，故適合用於不知道有多少最佳解的多目標最佳化問題。而傳統的菁英人工蜂群演算法是在演算開始時設定好菁英族群的數量，此與同時，若其中的菁英族群的數量不足，會導致演化過程中的優良菁英族群遺失，造成最終找到的最佳解數量較少，故適用於單目標最佳化問題。

在動態菁英人工蜂群演算法中必先將問題編碼成可運算的形式，故將單一專案排程編碼成蜜源。每個蜜源包含數個區段，代表該專案排程中所包含的任務。而區段的內容代表該任務的方案選擇，如圖 3 所表示的就是此專案排程中包含三個任務。其中最左邊的格子代表任務 1 採取方案 2；中間的格子代表任務 2 採取方案 4；而最右邊的格子是任務 3 採取方案 1。



圖 3 蜜源示意圖

利用這種編碼方式，可將專案排程編碼成可運算的形式，並利用工蜂(Employed Bees)、觀察蜂(Onlookers)和偵察蜂(Scouts) 來搜尋蜜源，同時將搜尋到的最佳解儲存起來，直到滿足最大迭代數為止，整體流程圖如圖 4 所示，並分別以以下小節簡述之。

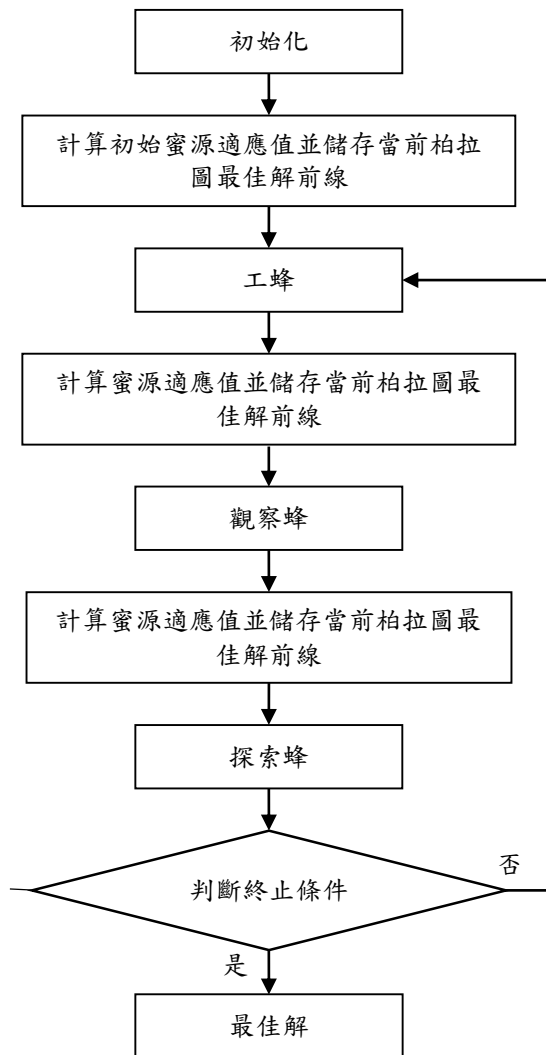


圖 4 動態菁英人工蜂群演算法流程圖

### 3.1 初始化

在初始化階段會將蜜源、開採極限與菁英族群數等參數初始化。蜜源會採取隨機的方式初始化該任務會採取的方案，以供演算過程計算；開

採極限會初始化設定為 0，在演化過程中逐漸增加，當蜜源更新後，開採極限重新設定為 0；菁英族群數初始化設定為 0，當演化過程中找到可能為柏拉圖最佳解前緣的解時加 1，並將該解加入菁英族群，當判斷菁英族群中有解非柏拉圖最佳解前緣的解時，將該解從菁英族群中刪除並將菁英族群數減 1。

### 3.2 適應值

論文所設計的適應函數由兩部分構成：時間適應值函數與成本適應值函數。因在演化的過程中，使用者並不清楚時間與成本兩者對於專案排程的重要性，所以在演化當中應該同時著重考慮時間與成本兩者的適應值，並用於以後的計算。故本實驗之時間與成本之適應函數方程式設計如下：

$$\text{時間適應值}(\text{fitt}) = \text{專案最大時間} / \text{蜜源總時間} \quad (1)$$

$$\text{成本適應值}(\text{fitn}) = \text{專案最大成本} / \text{蜜源總成本} \quad (2)$$

在取得專案中最大時間與成本之後，取該蜜源之時間成本與專案最大時間成本之比例做為本實驗之適應值，好處是不會因為時間與成本的單位不同造成計算時的差異，例如專案最大時間成本為(30 天, 20000 元)，而某一蜜源之總時間成本為(15 天, 10000 元)，則該蜜源之時間適應值為  $30 / 15 = 2$ ，成本適應值為  $20000 / 10000 = 2$ ，皆能分辨出該蜜源在該專案中所需之時間成本在最大時間成本所占之比例相同。

### 3.3 蜜蜂搜索策略

蜜蜂(工蜂、觀察蜂)利用方程式(3)來搜尋周圍的蜜源，其中 $x_{t+1}$ 表示蜜蜂新找到的蜜源也就是新的專案排程， $x_{t,i}$ 表示當前蜜源第 $i$ 的任務所採取的方案， $x_{k,i}$ 表示隨機蜜源 $k$ 中第 $i$ 的任務所採取的方案， $\emptyset$ 是隨機 1 或-1，用隨機的方式來決定蜜蜂的前進方向， $length$ 代表該專案排程包含的任務數。當 $x_{t,i} + \emptyset \times x_{k,i}$ 計算結果是大於任務 $i$ 的可選取方案數或小於 0 時，對計算結果加減任務 $i$ 的可選取方案數來做調整。找到新蜜源後計算新蜜源的適應值，若新蜜源的適應值優於原蜜源，則選擇新蜜源，否則保留原蜜源。

$$x_{t+1} = \sum_{i=0}^{length} (x_{t,i} + \emptyset \times x_{k,i}) \quad (3)$$

### 3.4 觀察蜂選取策略

觀察蜂選擇蜜源是透過輪盤法方式計算機率， $\text{fitt}_i$ 為全部蜜源數量中第 $i$ 個蜜源的時間適應值， $\text{fitc}_i$ 為全部蜜源數量中第 $i$ 個蜜源的成本適應值， $P_i$ 為全部蜜源數量中第 $i$ 個蜜源被選擇到的機率，當適應值越高，該蜜源被選擇到的機率越高。

$$P_i = \frac{f_{\text{fit}i} + f_{\text{itc}i}}{\sum (f_{\text{fit}i} + f_{\text{itc}i})}$$

### 3.5 探索蜂探索策略

當蜜源一直不能改善的情形達到開採極限時，工蜂會放棄該蜜源，轉變為觀察蜂重新尋找蜜源，並計算新的蜜源適應值。若新的蜜源適應值比原本的佳，則選擇新的蜜源，否則保留原有的蜜源。

### 3.6 動態菁英策略

由於本論文是為了解決多目標最佳化的問題，多目標最佳化找到的解將會是一組最佳解而非單個最佳解。基於我們運算前不知道會有多少個最佳解，因此本論文提出的方法中含有一個儲存機制，以建表的方式將運算中找到目前的最佳解先儲存起來，並刪除在儲存機制中判斷非柏拉圖最佳解前緣的解。

## 四、實驗與評估

本論文參考瀑布式模型(Waterfall Model)、正規化系統開發(Formal System Development)、遞增式開發(incremental development)等的軟體開發流程，將軟體開發流程每一階段之活動(如瀑布式模型中的需求定義、系統及軟體設計等)設定為該階段的任務，因此每個任務的先後順序已經固定，無需考慮任務間的前後順序的關係。而每個任務有許多的解決方案，例如測試的樣本數不同、開發人員的數量不同等因素，而這些解決方案有不同的時間成本和金錢成本來完成該任務。

本論文設計出兩種模擬實驗，一種是簡單專案一種是複雜專案，比較提出的方法動態菁英人工蜂群演算法(DEABC)、向量評估基因演算法結合禁忌搜尋法(TSVEGA)[4]和多目標基因演算法(MOGA)對於簡單專案與複雜專案的搜尋柏拉圖最佳解前緣的效能評估。

#### 4.1 簡單專案

在簡單專案中，全部有 5 個任務，每個任務有 4 個可選擇的方案，如表 1 所述。

表 1 簡單專案時間成本列表

	任務 1	任務 2	任務 3	任務 4	任務 5
方案 1 時間(天)	5	3	2	5	5
方案 1 成本(元)	2800	1500	5000	3000	4000
方案 2 時間(天)	8	2	10	7	6
方案 2 成本(元)	1700	1700	1400	2500	3000
方案 3 時間(天)	10	4	7	10	8
方案 3 成本(元)	900	1000	3500	2000	2000

本(元)					
方案 4 時間(天)	6	8	11	4	10
方案 4 成本(元)	2500	500	1000	4000	1000

在簡單專案中總計共有 1024 種不同的專案時間成本組合，而柏拉圖最佳解前緣(藍線)上的點共有 30 個，而這些 1024 種組合顯示如圖 5。在實驗中會使用 DEABC、TSVEGA 與傳統 MOGA 分別與此組最佳解進行對比，並將族群數目設定為 20，演化代數設定為 50，觀察每次演化結果與理論上柏拉圖最佳解的有多少差距。

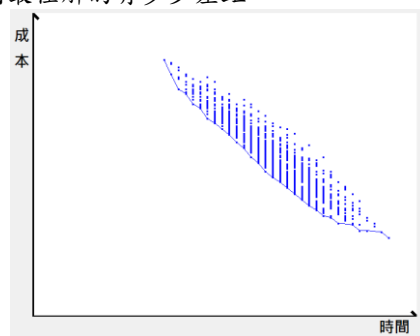


圖 5 簡單專案中柏拉圖最佳解前緣

利用 DEABC 模擬結果如圖 6 所示，圖 6 中綠色點代表 DEABC 搜索到的最佳解，綠色連線代表 DEABC 搜索到的柏拉圖最佳解前緣，與實際柏拉圖最佳解前緣對照，在模擬 100 次後取平均數，DEABC 找到了 24.74 個在柏拉圖最佳解前緣的解，而 TSVEGA 找到了 0.5 個在柏拉圖最佳解前緣的解，MOGA 找到了 0.3 個在柏拉圖最佳解前緣的解。

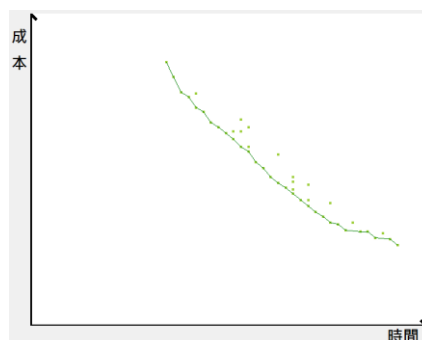


圖 6 簡單專案中 DEABC 模擬結果

#### 4.2 複雜專案

在複雜專案中，全部有 11 個任務，每個任務有 2 到 4 個可選擇的方案，如表 2 所述。

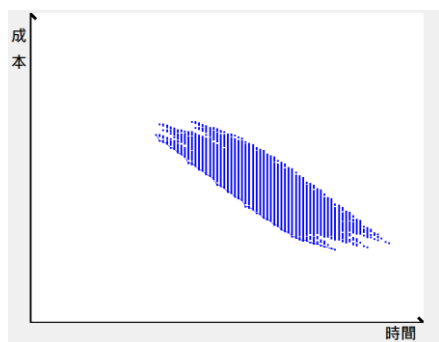


圖 7 複雜專案中柏拉圖最佳解前緣

表 2 複雜專案時間成本列表

	作業 項 1	作業 項 2	作業 項 3	作業 項 4
任務 1 時間(天)	5	8	10	
任務 1 成本(元)	2800	1700	900	
任務 2 時間(天)	1	2		
任務 2 成本(元)	500	300		
任務 3 時間(天)	2	10		
任務 3 成本(元)	5000	1400		
任務 4 時間(天)	5	7	10	4
任務 4 成本(元)	3000	2500	2000	4000
任務 5 時間(天)	5	7	10	
任務 5 成本(元)	2000	1500	100	
任務 6 時間(天)	2	10		
任務 6 成本(元)	5000	1500		
任務 7 時間(天)	5	7	10	4
任務 7 成本(元)	2000	1500	1000	300
任務 8 時間(天)	5	8		
任務 8 成本(元)	2800	1700		
任務 9 時間(天)	1	10		
任務 9 成本(元)	500	900		
任務 10 時間 (天)	2	10		
任務 10 成本 (元)	5000	1400		
任務 11 時間 (天)	5	7	10	4
任務 11 成本 (元)	3000	2500	2000	3000

在簡單專案中總計共有 294912 種不同的專

案時間成本組合，而柏拉圖最佳解前緣(藍線)上的點共有 50 個，而這些 294912 種組合顯示如圖 7。在實驗中會使用 DEABC、TSVEGA 與傳統 MOGA 分別與此組最佳解進行對比，並將族群數目設定為 200，演化代數設定為 50，觀察每次演化結果與理論上柏拉圖最佳解的有多少差距。

利用 DEABC 模擬結果如圖 8 所示，圖 8 中綠色點代表 DEABC 搜索到的最佳解，綠色連線代表 DEABC 搜索到的柏拉圖最佳解前緣，與實際柏拉圖最佳解前緣對照，在模擬 100 次後取平均數，DEABC 找到了 34.02 個在柏拉圖最佳解前緣的解，而 TSVEGA 找到了 7.3 個在柏拉圖最佳解前緣的解，MOGA 找到了 4.96 個在柏拉圖最佳解前緣的解。

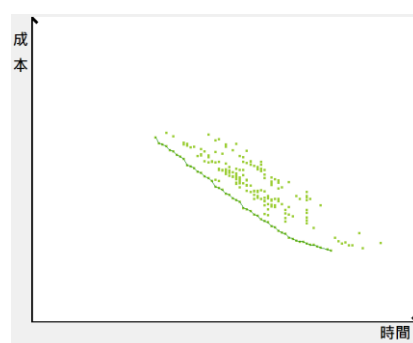


圖 8 複雜專案中 DEABC 模擬結果

本論文利用簡單專案與複雜專案來比較 DEABC、TSVEGA 與 MOGA 在搜尋柏拉圖最佳解前緣的效果，在模擬實驗中 DEABC 找到的柏拉圖最佳解前緣數量優於其他兩種演算法找到的數量。

## 五、總結

本論文利用動態菁英人工蜂群演算法(DEABC)來解決軟體專案時間成本最佳化問題，並與向量評估基因演算法結合禁忌搜尋法(TSVEGA)和多目標基因演算法(MOGA)來比較。由實驗結果證實本論文提出的方法和傳統窮舉法相比，在搜尋到柏拉圖最佳解前緣的解比窮舉法少，但 DEABC 在計算方面可以減省許多，並與近年來提出解決多目標最佳化的演算法 TSVEGA 和 MOGA 相比搜尋到柏拉圖最佳解前緣的能力，DEABC 相對優異。

## 參考文獻

- [1] Karaboga, Dervis. "An idea based on honey bee swarm for numerical optimization." Technical report-tr06, Erciyes university, engineering faculty, computer engineering department, Vol. 200, 2005.
- [2] 周于文，「應用蜂群演算法於結構最佳化設計之研究」，第九屆海峽兩岸航空太空學術研討會，台北淡水，2014 年 11 月。
- [3] 李廷威、洪燕竹，「人工蜂群演算法應用於

改善即時路況自動導航之研究」。碩士論文，國立嘉義大學資訊工程研究所，2015 年。

- [4] 許祐銘、洪燕竹，「結合向量評估基因演算法與禁忌搜尋法於軟體專案時間成本最佳化問題」。碩士論文，國立嘉義大學資訊工程研究所，2015 年。
- [5] 吳秉威、曾文宏。「以遺傳演算法求解 PCB 廠生產排程問題」。碩士論文，南台科技大學工業管理研究所，2005 年。
- [6] 林孝柔、蔣宗哲。「混合式基因演算法於多目標彈性零工式工廠排程問題之研究」。碩士論文，國立臺灣師範大學資訊工程研究所，2011 年。
- [7] 李建漳、潘乃欣、楊益東。「以禁忌搜尋法應用於專案工期/成本權衡問題最佳化之研究」。碩士論文，朝陽科技大學營建工程研究所，2003 年。
- [8] 李增源、黃怡碩、楊亦東。「重複性工程工期成本權衡之不確定性評估模式」。碩士論文，朝陽科技大學營建工程研究所，2004 年。
- [9] 賴紅松、董品杰、祝國瑞。「求解多目標規劃問題的 Pareto 多目標遺傳算法」。系統工程期刊，第 21 卷，第 5 期，第 24-28 頁，2003 年。
- [10] 馬衛孫正興。「基於精英蜂群搜索策略的人工蜂群算法」。計算機應用，2014 年。
- [11] 鄭東輝、林維垣、潘文超。「應用演化式計算預測美國道瓊股價指數。碩士論文。東吳大學經濟學系，2011 年。