

利用基因演算法於選題策略之研究

A Study of Applying Genetic Algorithms to Select Test Items

孫光天 戴伯昌 賴膺守
臺南師範學院資訊教育所 臺南師範學院資訊教育所 臺南師範學院資訊教育所
ktsun@ipx.ntntc.edu.tw m87507@stu.ntntc.edu.tw m87522@stu.ntntc.edu.tw

摘要

在教育評量技術中，題目反應理論(Item Response Theory, 簡稱 IRT)為目前廣泛研究與使用之評量方法，其中，如何應用訊息函數以選取最適當的題目，建構一份理想測驗，來符合預定施測目標，為一重要研究重點；由於此問題為一組合最佳化問題，屬於 NP-Complete 中一種問題，因此，沒有任一演算法可於多項式時間內求得最佳解。本文中，我們將提出一人工智慧技術“基因演算法”做為選題策略，經模擬結果，本方法十分有效率的求得最佳解(或接近最佳解)，肯定了本研究之人工智慧技術可成功的運用於教育測驗領域。

關鍵字：基因演算法，選題策略，題目反應理論，人工智慧，適性測驗

1. 簡介

題目反應理論(Item Response Theory, 簡稱 IRT)[7, 8]為目前發展較成熟之評量理論，其特點在於(1)題目參數估計與受試者無關，(2)受試者能力估計與施測題目無關，及(3)能力估計的精確度是可估計(已知)的；因此，廣泛應用於教育評量之相關研究中，其中一重要研究領域為：「如何建構一個理想的測驗，以符合測驗設計者施測目的」，亦為多人所探討；例如：要建構一份學術獎勵之成就測驗，則須由題庫中選取較難的題目來施測，才符合目標(測驗訊息函數)需求，此時，一測驗試題選取，變為一重要研究工作；而目前常被探討的選題方式有：(1)任意選取法(random)；(2)標準法(standard)；(3)中間難度法(middle difficulty)；(4)上下法(up and down)；(5)最大訊息法(maximum information)等五種[3]，其運算複雜度[5]為 $O(m)$ (任意選取法)與 $O(mn)$ (其他四種方法)(其中， m 為測驗之題數， n 為題庫中題目數)，執行效率均相當高，但由於選題方式過於簡單而缺乏彈性，以致於選出之試題與目標訊息函數間之誤差均偏高，故本研究擬以基因演算法技術[2, 9]為基礎，採用孫光天於 1999 年所提之能量函數[11, 12]，設計出一有效基因演算法的選題策略，使測驗題目反應出之訊息更能符合測驗設計者的需求。

基因演算法的想法是來自於生物學的進化理論和遺傳法則，是 John Holland 於 1975 年所提出來的構想[4]。基因演算法的特點是運算簡單、有效，它主要是使用三個運算子：複製(reproduction)、交配(crossover)、突變(mutation)。演化的根據是適合度函數(fitness function) -- 亦即所要求最適化的成本、時間、資源。應用基因演算法在最佳化之設計上，參數的設定 往是最困難的地方，因其將影響最後搜尋結果，且參數之間互相關連，彼此相依。我們則以類神經網路中的能量函數，將其用於基因演算法中的適合度函數，使問題之呈現簡單化。本文第二章，將介紹傳統適性測驗之選題方法，並以一實例做說明；第三章則介紹基因演算法模式及其選題運

作原理；第四章則進行實例分析及效能評估；最後，則做一結論與建議。

2. 傳統選題策略

在適性測驗中，每一題目對不同能力(θ)均反應出不同訊息量($I(\theta)$)，而此訊息量將可用來做選題時之參考，當訊息量越大，表示此題目對受試者能力估計越有效，因此，適性測驗可在最短施測題數內估計出受試者之潛在能力。表一為我們早期研究所建構之國民小學自然科電腦化適性測驗系統中題庫之部份題目(35 題)，每一題目在不同能力則呈現不同之訊息量($I(\theta)$)。而一測驗之訊息量，為測驗中每一題目累加求得，因此，對不同施測需求，則選取不同題目，以達成測驗設計者的目標。今以傳統之五種方法來選 30 題，建構一單峰型態測驗，最大峰值在能力為 0，各方法之步驟分述如下：

- (1) 任意選取法(random)
使用亂數表，由題庫中任意選取 30 題。
- (2) 標準法(standard)
使用傳統題目統計法(題目的難度(p)和鑑別度(r))來選題，首先，選出難度為 .30 至 .70 之題目，接著再由其中選出鑑別度最大之 30 題。
- (3) 中間難度法(middle difficulty)
對所有題目在能力(θ)=0 中，挑出訊息量最大之 30 題。
- (4) 上下法(up and down)
此方法包括三個步驟，第一步驟為：在所有題目中，選出題目之能力值(θ)=-1 中，訊息量最大之一題；第二步驟：類似第一步驟，但在題目之能力值(θ)=0 中，選出訊息量最大之一題；第三步驟亦類似，但由題目之能力值(θ)=1 中，選出訊息量最大之一題；此三種選取之題目均不重覆。當此三步驟重覆 10 次，則可選出測驗所需之 30 題題目。
- (5) 最大訊息法(maximum information)
由題庫中選出能力值(θ)=-1.0 和 1 三個訊息量和，其和最大的 30 題題目。

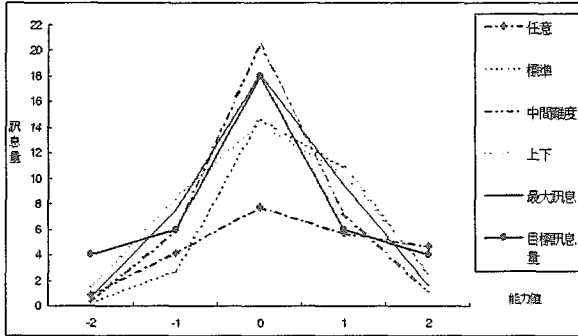
以上列五種方法，自題庫中選取 30 題編制成一份測驗，當測驗目標訊息函數如表二所示時，則最後所求得之測驗訊息量將如圖一所示，由圖中可明顯感覺出傳統五種方法彈性並不大，與早期相關研究者所得之結果類似 [1]，表示此五種方法仍存在不小誤差，因此，若能提出一更有效之選題策略，將使測驗之設計及測驗品質，達到一更高水準。

表一、適性測驗題目之題目參數與訊息量及傳統統計法之難度(p)與鑑別度(r)(部分題目)

編號	題目參數			能力等級							傳統統計數	
	a	b	c	-3	-2	-1	0	1	2	3	p	r
1	0.763	-3.000	0.270	0.242	0.193	0.078	0.024	0.007	0.002	0.001	0.969	0.584
2	0.542	-2.218	0.280	0.085	0.123	0.104	0.060	0.028	0.012	0.005	0.897	0.405
3	1.252	0.665	0.330	0.000	0.000	0.007	0.202	0.577	0.155	0.021	0.547	0.442
4	0.648	-2.978	0.250	0.181	0.159	0.081	0.031	0.011	0.004	0.001	0.953	0.455
5	0.756	-1.816	0.260	0.082	0.227	0.215	0.096	0.031	0.009	0.002	0.894	0.607
6	0.445	-1.451	0.290	0.038	0.068	0.082	0.069	0.045	0.025	0.013	0.809	0.299
7	0.446	-1.774	0.270	0.051	0.079	0.084	0.064	0.039	0.021	0.011	0.834	0.322
8	0.732	-3.000	0.320	0.199	0.167	0.073	0.024	0.007	0.002	0.001	0.984	0.245
9	0.647	-0.611	0.360	0.008	0.043	0.119	0.146	0.091	0.038	0.014	0.778	0.386
10	1.594	0.791	0.440	0.000	0.000	0.001	0.081	0.760	0.142	0.010	0.606	0.351
11	0.537	-1.598	0.290	0.049	0.101	0.118	0.085	0.045	0.020	0.009	0.847	0.407
12	0.564	-1.124	0.220	0.038	0.100	0.149	0.126	0.071	0.032	0.013	0.778	0.449
13	0.745	0.989	0.350	0.000	0.001	0.013	0.081	0.194	0.162	0.068	0.534	0.293
14	0.467	0.546	0.360	0.003	0.011	0.032	0.062	0.078	0.066	0.042	0.628	0.228
15	0.493	-1.497	0.290	0.042	0.083	0.100	0.080	0.047	0.024	0.011	0.828	0.333
16	0.601	-2.323	0.270	0.110	0.155	0.116	0.058	0.024	0.009	0.003	0.913	0.471
17	0.728	0.005	0.320	0.002	0.015	0.086	0.197	0.167	0.073	0.024	0.672	0.415
18	0.612	-1.362	0.250	0.045	0.123	0.167	0.120	0.058	0.023	0.008	0.822	0.473
19	0.836	-0.262	0.250	0.002	0.028	0.170	0.312	0.178	0.056	0.014	0.688	0.579
20	0.715	-1.953	0.260	0.097	0.214	0.184	0.083	0.028	0.009	0.003	0.900	0.547
21	0.636	-1.572	0.270	0.055	0.142	0.170	0.106	0.046	0.017	0.006	0.856	0.495
22	0.559	-0.850	0.260	0.023	0.071	0.128	0.128	0.080	0.038	0.016	0.756	0.436
23	1.492	-2.281	0.240	0.235	0.966	0.174	0.015	0.001	0.000	0.000	0.962	0.905
24	0.805	-1.344	0.200	0.044	0.207	0.315	0.172	0.056	0.015	0.004	0.837	0.641
25	0.564	-1.936	0.280	0.073	0.127	0.122	0.074	0.035	0.014	0.006	0.878	0.424
26	1.714	0.924	0.340	0.000	0.000	0.000	0.055	1.088	0.221	0.013	0.488	0.354
27	0.493	-0.998	0.230	0.032	0.074	0.110	0.104	0.069	0.037	0.018	0.753	0.384
28	0.827	-1.629	0.200	0.074	0.276	0.302	0.129	0.037	0.009	0.002	0.875	0.655
29	0.717	-0.985	0.140	0.035	0.150	0.280	0.218	0.095	0.032	0.010	0.759	0.605
30	0.974	-1.803	0.240	0.081	0.381	0.324	0.094	0.020	0.004	0.001	0.913	0.707
31	0.435	-0.546	0.200	0.024	0.052	0.083	0.093	0.075	0.049	0.027	0.656	0.331
32	0.803	0.537	0.310	0.000	0.003	0.035	0.171	0.250	0.130	0.041	0.575	0.381
33	1.187	0.080	0.350	0.000	0.002	0.059	0.466	0.293	0.052	0.007	0.684	0.534
34	0.844	-1.176	0.180	0.031	0.191	0.364	0.215	0.068	0.017	0.004	0.816	0.681
35	0.730	0.169	0.290	0.001	0.012	0.075	0.199	0.192	0.090	0.030	0.628	0.444

表二、單峰型態測驗目標訊息函數

		能力等級				
能力值	-2.0	-1.0	0.0	1.0	2.0	
訊息量	4	6	18	6	4	



圖一、單峰型態測驗目標訊息曲線與五種不同選題策略之測驗訊息曲線圖

如果是雙峰型態測驗，其測驗目標訊息函數如表三所示，則一些傳統方法必須修改如下：

- (1) 標準法(standard)
使用傳統題目統計法(题目的難度(p)和鑑別度(r))來選題，首先，選出難度為.10 至.40(較難題目)與難度為.60 至.90(較簡單題目)之題目，接著再由其中分別選出鑑別度最大之 15 題。
- (2) 中間難度法(middle difficulty)
此方法改成高低難度法(low-high difficulty)。對所有題目在能力(θ)=-1 和 1 中，分別挑出訊息量最大之 15 題。
- (3) 最大訊息法(maximum information)
由題庫的較低能力範圍和較高能力範圍中，分別選出訊息量之和最大的 15 題題目，其訊息量和的計算方式如下：
較低能力範圍之訊息量和 = $0.5 * I(-2.0) + I(-1.0) + 0.25 * I(0.0)$
較高能力範圍之訊息量和 = $0.5 * I(2.0) + I(1.0) + 0.25 * I(0.0)$

表三、雙峰型態測驗目標訊息函數

		能力等級				
能力值	-2.0	-1.0	0.0	1.0	2.0	
訊息量	4	12	6	12	4	

3. 基因演算法選題策略

對一測驗訊息量之需求，可如圖一曲線中之目標訊息量所示，在每一能力點(θ)我們有一期望之訊息量值(以 d_i 表示 (i 為能力等級指標))，與我們選出題目所形成測驗之訊息量(以 O_i 表示)，將有一誤差，此時我們將此誤差定義為一能量函數 (energy function)：

$$E_i = \sum_{i=1}^s (d_i - O_i)^2 \quad (1)$$

其中， d_i 為測驗目標訊息量在能力等級指標 i 時之訊息

量值， s 為能力等級數(例如：我們只探討能力值為 -2.0,-1.0,0.0,1.0,2.0 時，能力等級數 s 為 5)， O_i 為以選題策略產生出之測驗其訊息量值。愈小的 E_i 值(Error, 訊息量誤差值)，則愈符合設計者所需求之測驗目標。另外，以選題策略產生之訊息量 O_i 可以下式(2)表示：

$$O_i = \sum_{j=1}^n w_{ij} x_j, \quad x_j \in \{0,1\} \quad (2)$$

其中， w_{ij} 表示題目 j 在能力等級指標 i 時之訊息量； x_j 表示題庫第 j 題題目是否被選擇放入測驗試題中，若放入測驗中則為 1，否則為 0； n 為題庫之總題數。

此外，尚須考量下列因素：

- (1) 初始值之設定：
對於選題策略而言，初始之測驗題目數應為 0 (即初始時，未選任何題目於測驗中)，等決定那一題目較適當時，才選入測驗中，所以，題庫中所有題目初始時均未選入測驗中，即初始的 $x_j = 0, j = 1 \sim n$ 。
- (2) 題數限制：
當一測驗之題數為固定值，除了考量訊息量之誤差外，亦要加入題目數之限制：

$$E_m = T \left(\sum_{j=1}^n x_j - m \right)^2 \quad (3)$$

其中， m 為測驗中所須題數， T 為一加重值：

$$T = \sum_{i=1}^s d_i^2 \quad (4)$$

因此，一個 m 題的測驗，總能量函數可定義為：

$$E_{Total} = E_i + E_m \quad (5)$$

當測驗總題數 ($\sum_{j=1}^n x_j$) 不等於 m 時，能量函數 E_m 值將遠大於 E_i ，因此，總能量函數最小值之必要條件為 $E_m = 0$ ，我們將式(5)能量函數作為基因演算法之適合度函數 (fitness function)，當式(5)值愈小時，表適合度愈高，即測驗之選取題目數必須為 m 題，此時，才能達到較高適合度。

- (3) 基因演算法

基因演算法選題方式之詳細步驟說明如下：

步驟一、設定所有變數初始值：

(a)適合度函數之變數值： d_i 為能力值指標 i 之測驗目標訊息量值，題庫之總題數 $n = 320$ ，測驗總題目數 $m = 35$ 。

(b)設定基因型態：題庫中所有題目 j 之選取狀態 x_j 為染色體中基因型態，0 表示題目 j 未選入測驗中，1 表示題目被選入測驗中； w_{ij} 為每一題目 j 在能力指標 i 之訊息量值。染色體(基因組合)長度等於總題數 n 。

(c)初始時間變數 $t = 0$ ，族群數目=30，世代數目=1500。

步驟二、初始階段(initialization)：

任意選取 30 個染色體(即 30 份測驗)，設定為初始親代母群，每一個體即代表一個選題結果。

步驟三、評估階段(evaluation)：

評估親代母群中的個體，將每一個體代入前式(5)總能量函數，如果評估條件符合就結束演化(到達最大世代數目(=1500)或連續 10 代最佳解未有進一步改善)，此時，對所有 $x_j = 1$ 的題

目，即為被選取而編入之測考試題，否則進入步驟四，進行進化。

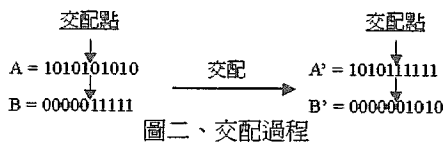
步驟四、選擇階段(selection)：

讓評估中適合度較高的親代個體較易生存，淘汰部份表現不佳的親代個體。方法是根據親代個體的適合度大小，做輪盤式選擇(roulette wheel selection)[9, 10]，即適合度越高，被選的機率越高，反之則越低。

步驟五、基因操作階段(genetic operation)：

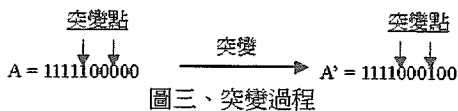
以交配(crossover)、突變(mutation)、複製(reproduction)的基因操作，產生新的變異個體，而得到下一世代的母群。方法如下：

(a) 交配(crossover)：我們把交配率定為 100%，即子代個體皆是由親代個體互相交配產生。首先依照步驟四的選擇方法挑出兩個親代個體，當要做交配的動作時，先要在字串中隨機選擇一個交配點(crossover point)，然後根據此交配點，將交配點左邊或右邊的片段與另一個字串(依照相同互換點劃分)交換即可，所產生新的字串即為子代個體。交配過程如圖二所示：



圖二、交配過程

(b) 突變(mutation)：當測驗為固定題數時，我們檢查選取題目數是否為 35 題，若少於 35 題，則從未選取題目中隨機選取至 35 題；若多於 35 題，則從選取題目中隨機剔除至 35 題。當測驗為不固定題數時，我們只做若多於 35 題，則從選取題目中隨機剔除至 35 題。最後，我們選取一個子代隨機改變其中的一個基因，若該基因型態為 0，則使之變為 1，並隨機再找一個基因型態為 1 者，使之變為 0。突變過程如圖三所示：



圖三、突變過程

(c) 複製(reproduction)：因為交配率、突變率愈大時，新的結構將愈快出現於群體中。當交配率、突變率太高時，會使得群體中表現優良者在發揮效果前就被改變了。因此，我們使用菁英選擇(elitist selection)[6]，即無條件直接複製保留 2 個最佳親代個體。所以，實際上子代只產生 28 個新的個體。

步驟六、將所產生的子代新母群，代入並回到步驟三評估階段，進行下一世代的演化過程。

4. 效能評估

我們以先期完成之題庫為依據，比較傳統五種選題法、孫光天的類神經網路選題法[11]與本研究法之效能，我們以 100 次選題結果的平均值來做效能之比較，其中測驗目標訊息函數在不同能力(θ)的訊息量 d_1, d_2, d_3, d_4, d_5 ，我們以隨機函數來產生，其範圍如下所示：

(1) 在單峰型態時， d_1, d_2, d_3, d_4, d_5 的範圍：

$$3 \leq d_1 \leq 4,$$

$$5 \leq d_2 \leq 8,$$

$$19 \leq d_3 \leq 22,$$

$$5 \leq d_4 \leq 8,$$

$$3 \leq d_5 \leq 4$$

(2) 在雙峰型態時， d_1, d_2, d_3, d_4, d_5 的範圍：

$$4 \leq d_1 \leq 5,$$

$$10 \leq d_2 \leq 12,$$

$$6 \leq d_3 \leq 8,$$

$$10 \leq d_4 \leq 12,$$

$$4 \leq d_5 \leq 5,$$

結果如圖四(單峰不固定題數，但上限為 35 題)、圖五(單峰固定題數 35 題)、圖六(雙峰不固定題數，但上限為 35 題)與圖七(雙峰固定題數 35 題)所示，由圖之結果顯示，我們所提的方法其效能遠優於其它方法(如表四所示)，證明了本研究之價值。圖八至圖十一為基因演算法進化過程之收斂狀態圖。

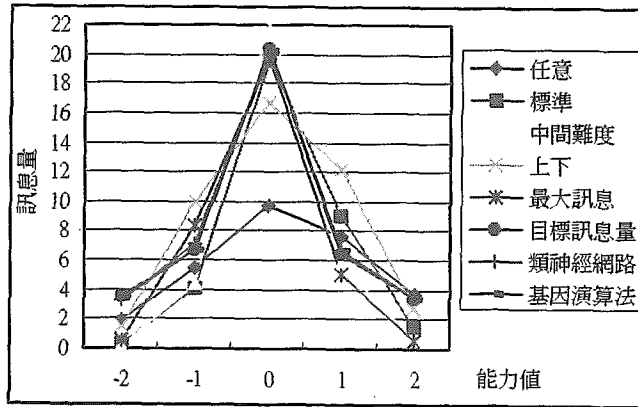
表四、基因演算法、類神經網路法與傳統五種選題法之誤差量(MSE)

	基因演算法	類神經網路法	任意法	標準法	中間難度法	上下法	最大訊息法
圖四	0.068	0.607	120.110	31.304	31.269	65.704	26.688
圖五	0.164	1.122	120.110	48.440	84.948	77.985	58.507
圖六	0.011	1.655	65.602	34.265	10.677	NA	18.521
圖七	0.013	1.941	65.602	40.111	21.835	NA	21.119

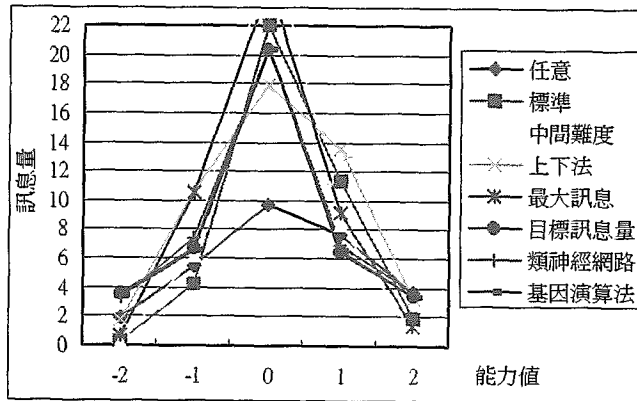
在分析時間複雜度上，我們分成兩種狀況：(1)題數不限定(但上限為 m 題)，(2)測驗題數限定為正好 m 題；因此，傳統的標準法、中間難度法、上下法、最大訊息法等，不論在狀況(1)或(2)條件下，每次均需由題庫中找最大值(時間複雜度為 $O(n)$)，最多找 m 題，所以，總複雜度為 $O(mn)$ ；而類神經網路，每次題目選取時，計算修改每一 x_i 狀態，而題庫中共有 n 個 x_i 須計算，每選一題所須之計算亦為 $O(n)$ ，最多找 m 題，所以，總複雜度也為 $O(mn)$ [11]。然而，我們所提出的基因演算法，因為在每一世代每一族群中的每一親代，皆須計算總訊息量，所以總共要做 n 次；而在產生子代時，當測驗為固定題數時，必須檢查選取題目數是否為 m 題，若少於 m 題，則從未選取題目中隨機選取至 m 題，所以，在最壞情況下：

$$\text{時間複雜度} \propto \text{世代數目} \times \text{族群數目} \times (n + m)$$

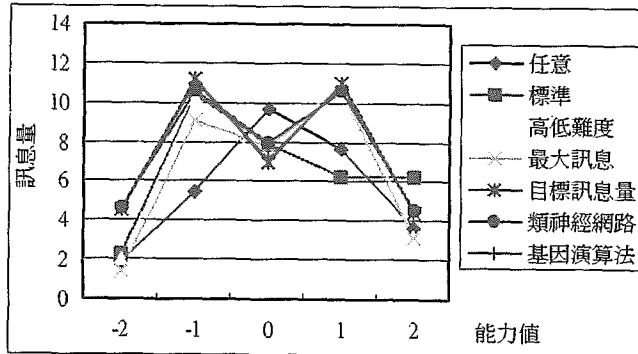
因為世代數目與族群數目為固定常數，且 $m \ll n$ ，所以，總複雜度為 $O(n)$ 。各選題法之時間複雜度如表五所示。



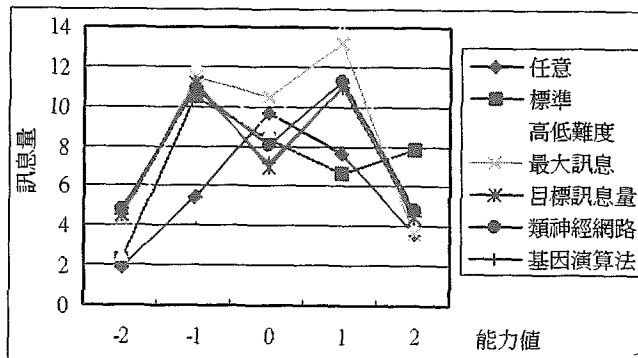
圖四、基因演算法選題法與傳統選題法在單峰不固定題數下效能比較圖



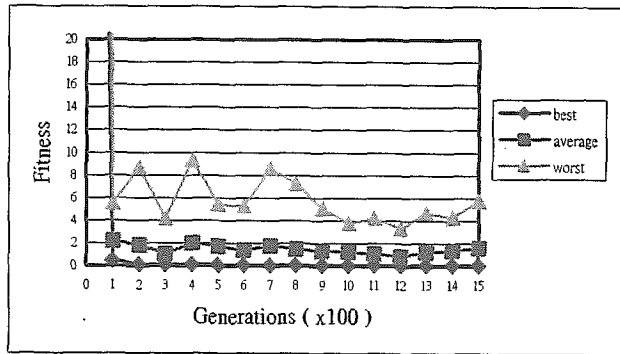
圖五、基因演算法選題法與傳統選題法在單峰固定題數(35 題)下效能比較圖



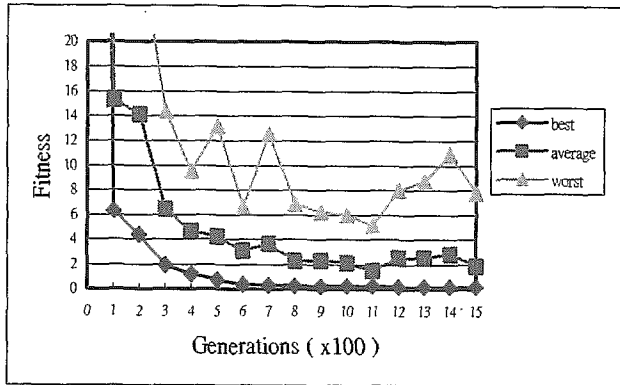
圖六、基因演算法選題法與傳統選題法在雙峰不固定題數下效能比較圖



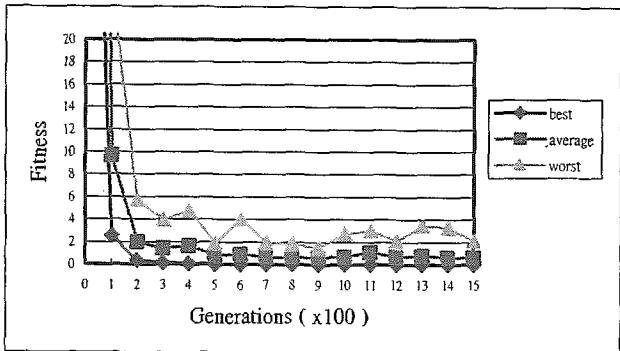
圖七、基因演算法選題法與傳統選題法在雙峰固定題數(35 題)下效能比較圖



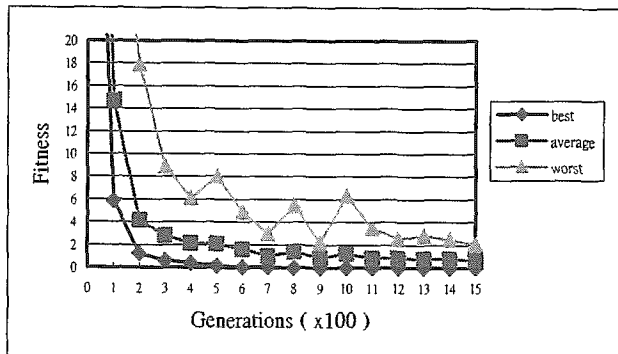
圖八、基因演算法選題法單峰不固定題數下之收斂狀態圖



圖九、基因演算法選題法單峰固定題數下之收斂狀態圖



圖十、基因演算法選題法雙峰不固定題數下之收斂狀態圖



圖十一、基因演算法選題法雙峰固定題數下之收斂狀態圖

表五、基因演算法、類神經網路法與傳統五種選題法之時間複雜度

(m: 表測驗之題數, n: 表題庫之題數($m << n$))

	基因演 算法	類神經 網路法	任意法	標準法	中間難 度法 高低難 度法	上下法	最大訊 息法
時 間 複 雜 度	$O(n)$	$O(mn)$	$O(m)$	$O(mn)$	$O(mn)$	$O(mn)$	$O(mn)$

5. 結論與建議

本文中，我們以自行設計的基因演算法來進行選題策略，以此技術建構出之測驗，其訊息函數與測驗目標之訊息函數，兩者間的誤差遠低於其他的六種選題策略(類神經網路選題法、任意選取法、標準法、中間難度法、上下法及最大訊息法)，對於測驗之選題策略，提供更精確且有效率的技術，也擴展了基因演算法之運用範圍，對日後人工智慧技術運用於測驗領域，提供更廣泛用途與更有效益的技術，使測驗品質更能符合設計者之需求，對教育評量研究者提供更有效的工具。

誌謝

本研究計畫經費蒙國科會補助(編號 NSC 88-2511-S-024-005-), 謹此致謝。

參考文獻

[1] L. L. Cool and R. K. Hambleton. "A comparative study of item selection methods utilizing latent trait theoretical models and concepts," Laboratory of Psychometric

and Evaluative Research Report, No. 88, School of Education, University of Massachusetts, Amherst, 1978.

- [2] D. E. Goldberg. Genetic algorithms in search, optimization, and machine learning, Addison-Wesley, Reading, Massachusetts, 1989.
- [3] R. K. Hambleton and H. Swaminathan. ITEM RESPONSE THEORY -- Principles and Applications, Kluwer Academic Publishers Group, Netherlands, 1985.
- [4] J. H. Holland. Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, Michigan, 1975. (Second edition: MIT Press, Cambridge, Massachusetts, 1992.)
- [5] E. Horowitz and S. Sahni. Fundamentals of Computer Algorithms, Computer Science Press, Potomac, Maryland, 1978.
- [6] S. R. Ladd. Genetic algorithms in C++, M&T Books, New York, 1996.
- [7] F. M. Lord. "A theory of test scores," Psychometric Monograph, No. 7, 1952.
- [8] F. M. Lord and M. R. Novick. Statistical theories of mental test scores, Addison-Wesley, Reading, Massachusetts, 1968.
- [9] M. Mitchell. An introduction to genetic algorithms, MIT Press, Cambridge, Massachusetts, 1996.
- [10] G. Winter, J. Periaux, M. Galan, and P. Cuesta, (eds.). Genetic Algorithms in Engineering and Computer Science, John Wiley & Sons, inc., New York, 1995.
- [11] K. T. Sun. "An Effective Item Selection Method b Using AI Approaches," Proc. of Advances in Intelligent Computing & Multimedia Systems, pp. 137-142, Baden-Baden, Germany, August 2-7, 1999.
- [12] K. T. Sun, "Applying AI Techniques to Construct a Desired Test," 7th International Conference on Computers in Education, pp. 117-124, Chiba, Japan, November 4-7, 1999.