*Journal of Science and Engineering Technology, Vol. 5, No. 2, pp. 25-34 (2009)*

# 基因演算法中不同選擇策略的替代性與互補性

郭定
德明財經科技大學國際貿易系
114 台北市內湖區環山路一段 56 號

## 摘　要

基因演算法是一種基於演化概念的群體式搜尋法。根據新達爾文主義的觀念，演化過程可分爲三種：方向性選擇、分裂性選擇與平衡性選擇。傳統的基因演算法依循「適者生存，不適者淘汰」原則，採用的就是方向性選擇；它讓群體中表現較佳的個體有較多的機會繁衍，但是如此往往也容易使搜尋過程陷入局部最佳解的困境。本研究顯示了，不同的選擇策略的確會影響基因演算法找到最佳解的能力；相較於平衡性選擇而言，分裂性選擇對於方向性選擇是較具有替代性；而分裂性選擇與平衡性選擇兩者對於方向性選擇而言具有互補性與替代性；同時，也顯示了分裂性選擇與平衡性選擇兩者之間具有互補性。總而言之，方向性選擇並非基因演算法的必然策略。

**關鍵詞：**基因演算法，方向性選擇，分裂性選擇，平衡性選擇

# Substitution and Complementarities of Different Selection Strategies in Genetic Algorithms

TING KUO
*Department of International Trade , Takming University of Science and Technology*
*56, Huan-Shan Rd., Sec.1, Taipei, Taiwan 114, R.O.C.*

## ABSTRACT

A genetic algorithm is a population-based search technique employing natural selection to imitate evolution. According to the Neo-Darwinists, natural selection can be classified into three categories: directional, disruptive, and stabilizing. By following the “survival-of-the-fittest” principle, traditional genetic algorithms can be viewed as evolutionary processes for adopting directional selection to obtain more reproductive opportunities for superior individuals. However, this strategy is sometimes myopic, being apt to trap the search into a local optimum. This study shows that (1) different selection strategies actually have an effect on the searching power of genetic algorithms, (2) disruptive selection can substitute for directional selection more than stabilizing selection does, (3) both disruptive and stabilizing selection can complement and substitute for directional selection, and (4) disruptive and stabilizing selection are complementary. Therefore, directional selection is not the default strategy in genetic algorithms.

**Key words:** genetic algorithms, directional selection, disruptive selection, stabilizing selection

## 一、前言

基因演算法（或稱遺傳演算法）是一種基於遺傳演化概念的群體式（population-based）搜尋法。群體中的每一「個體」（individual）分別表示待解問題中的一個可行解；使用基因演算法解決問題時，首先要選擇一個適當的表示法來表示待解問題中的可行解，一般採用二元字串（binary string）表示法，當然還有許多其他的表示法 [20, 25, 30, 35]。這些以特定表示法表達的可行解稱爲該可行解的「基因型」（genotype），而每一「個體」所對應的「表現型」（phenotype），係用以衡量該「個體」對於解決該待解問題上的表現優劣。而遺傳演化概念則是表現在每一「世代」（generation）中反覆套用的三種運算子：選擇（selection）、交叉組合（crossover）與突變（mutation）[24]。選擇運算子，或稱爲複製（reproduction）運算子，是指在一個群體中讓當代較優秀的「個體」能有多些繁殖下一代的機會。交叉組合運算子則是期待透過兩個「個體」之交配能產出更好的下一代；最常見的是單點交叉，這是指將兩個「個體」的「基因型」在隨機選定的某個位置後面的一段基因互換。而突變則是演化過程或許是爲了適應環境也或許是無可避免的變異；最常見的突變運算子是將「個體」的「基因型」在隨機選定的某個位置上的值替換成另一個值。

基因演算法已經成功地被應用在許多領域上，例如設計、規劃、控制、最佳化問題與機器學習等等 [24]。這些年來，有越來越多的研究採用基因演算法來處理各式各樣的問題。國內學術界也有許多研究成果，例如辨識系統 [14, 17]、影像處理 [1, 6]、預測 [15, 16]、決策支援與專家系統 [7]、投資組合 [3, 4]、排程 [10]、設計 [2]、電子商務 [8]、資訊管理 [12] 與資訊安全方面 [5, 11, 13]。儘管，基因演算法已經成功地被應用在解決許多問題上，但仍存在一些問題對基因演算法而言並不容易解。探討究竟是那些問題特性導致基因演算法表現不好，一直是個很重要的研究方向 [19, 21-23]。由於依循著「適者生存，不適者淘汰」之原則，傳統的基因演算法會賦予較多的機會給那些比起整個群體平均表現要好的區域，但是如此往往也容易陷入局部最佳解的困境。關鍵在於甚麼樣的「個體」才應被視爲「適者」？由於，本質上基因演算法就是一種調適的搜尋過程，而這調適能力主要是來自於選擇運算子。另外，已知選擇運算子是所有演化計算（evolutionary computation）技術，如演化策略（evolution strategies）及演化規劃（evolutionary programming），的共同核心部份[18]。 因此，本研究要探討基因演算法中最關鍵的機制：選擇。

根據新達爾文主義（Neo-Darwinism）的觀念，演化過程可分爲三類：方向性選擇（directional selection）、分裂性選擇（disruptive selection）與平衡性選擇（stabilizing selection）[33]。[1]方向性選擇是讓整個群體朝向「表現型」的某一端繁衍；分裂性選擇是傾向淘汰平庸「個體」而讓整個群體朝向「表現型」的兩個極端繁衍；而平衡性選擇則是淘汰極端「個體」而讓整個群體朝向「表現型」較正常化去繁衍。一般我們提到天擇（nature selection）時，多半就是指方向性選擇，傳統的基因演算法採用的也就是方向性選擇。本研究採用 120 個測試例子證實了：不同的選擇策略的確會影響基因演算法找到最佳解的能力。本研究更顯示了：相較於平衡性選擇而言，分裂性選擇對於方向性選擇是較具有替代性；而分裂性選擇與平衡性選擇兩者對於方向性選擇而言具有互補性與替代性；同時，也發現分裂性選擇與平衡性選擇兩者之間具有互補性。總而言之，方向性選擇並非基因演算法的必然策略。

## 二、研究動機

本研究的第一個研究動機是希望提供更充份的實驗證據顯示使用基因演算法並不一定非得採用方向性選擇。最常見的選擇運算子是採用比例性選擇（proportionate selection），這是指依照群體中每一「個體」的適應值（fitness）在整個群體適應值的比例原則賦予其繁殖（或是被選擇）的機會。另外，還有競賽式選擇（tournament selection）及排序選擇（rank selection）等方法 [26]。值得提醒讀者的是，本文中所稱的選擇策略並不是對比於這些方法，而是對比於採用甚麼原則將群體中每一「個體」的目標函數（objective function）值，本文以下均簡稱爲函數值，轉換爲適應值？不同的轉換原則對應著不同的選擇策略。傳統基因演算法採用的方向性選擇是直接以「個體」的函數值當作其適應值或是兩者之間的對映關係爲單調式（monotonic）[27]。亦即，若 $f(x)$ 是「個體」$x$ 的函數值，$g(x)$ 是「個體」$x$ 的適應值，則函數值與適應值之間的對映關係爲：$g(x_1) \le g(x_2)$ 若且唯若 $f(x_1) \le f(x_2)$。

Kuo 與 Hwang 曾開創地以分裂性選擇作爲衡量「個體」

[1] 針對某個特徵（trait）而非基因體（genome）。

適應值的原則，並證實該法確實能解決傳統基因演算法無法解決或表現不好的一些問題，並且能適度地保持整個群體的亂度而不會太早收斂 [31, 32]；其函數值與適應值之間的對映關係爲：g(x)=*abs*(f(x) - f(*t*))；其中，f(*t*) 是整個群體在「世代」*t* 時的平均函數值，而 *abs* 是取絕對值的運算。很顯然地，這是一種非單調式（non-monotonic）的對映關係。最近，Kuo 更以平衡性選擇作爲衡量「個體」適應值的原則，並驗證這種選擇策略也確實能掌握成敗關鍵 [9]。其函數值與適應值之間的對映關係爲：g(x)=1/(*abs*(f(x) - f(*t*)))。同樣地，這也是一種非單調式的對映關係。不過，這些研究都只是以少數的測試例子作爲佐證，而且並未探討不同的選擇策略彼此之間有何關聯。因此，本研究採用 120 個測試例子作實驗，以提供更充份的證據顯示使用基因演算法並不一定非得採用方向性選擇。至於本研究的第二個研究動機則是希望進一步分析，不同的選擇策略彼此之間有何關聯。

## 三、研究焦點

基因演算法中主要有三種運算子：選擇、交叉組合與突變；而本研究的焦點是集中在選擇運算子。因爲，基因演算法本質上是一種調適過程（adaptive process），它主要是藉著逐步調整搜尋空間中各個區域被「取樣」（sampled）到的機率而達成最佳化的目標，而這調適能力主要是來自於選擇運算子，以下將詳細說明之。基因演算法中的選擇運算子乃是基於各個可行解，或稱爲「個體」，的「表現型」來運作，而交叉組合運算子與突變運算子則是操作於各個可行解的「基因型」上。表面上，基因演算法是直接地在待解問題的「基因型」空間（genotype space）上操作，實際上它背後是間接地在待解問題的「模式」空間（schema space）中進行搜尋。這就是眾所周知的基因演算法之隱含式平行化（implicit parallelism）特性。

所謂「模式」（schema）是指一組的「基因型」組合。在以長度 $L$ 的二元字串爲表示方法的情況下，「個體」是定義在 $\{0,1\}^L$ 中的一個字串，而「模式」則是定義在 $\{0,1,*\}^L$ 中的一個字串。換句話說，共有 $2^L$ 個不同「個體」，而有 $3^L$ 個不同「模式」。一個含有 $k$ 個 $*$ 的「模式」，所代表的是「基因型」空間中 $2^k$ 個不同「個體」的集合。值得提醒讀者的是，任一個「個體」本身就是一個不含任何 $*$ 的「模式」。而一個含有 $k$ 個 $*$，也就是說有 $L$-$k$ 個位置已是確定的 0 或 1，的「模式」稱爲一個 $L$-$k$「階」（order）的「模式」；而這 $L$-$k$ 個的 0 或 1，稱爲「已定義位元」（defined bits），中最左邊與最右邊所距離的位置數稱爲此「模式」的「定義長度」（defining length）。例如，**0*11**1 是一個 4「階」的「模式」，其「定義長度」爲 6。當一個「個體」與一個「模式」的所有「已定義位元」均吻合時，稱此「個體」爲該「模式」的一個「代表個體」（representative individual）。例如，00010，00011，00110，與 00111 都是「模式」00*1* 的「代表個體」，而 10011 或 01110 則不是。很顯然的，一個含有 $k$ 個 $*$ 的「模式」具有 $2^k$ 個不同的「代表個體」。當一個「模式」至少有一個「代表個體」出現在群體中時，我們稱此「模式」被「取樣」到了。根據 Holland [29] 的設計，基因演算法本質上是一種調適過程，以逐步調整各「模式」被「取樣」的機率，而基因演算法中最重要的理論基礎：「模式」定理（schema theory），就是用來說明此調適過程。「模式」定理是指在一個使用比例性選擇、單點交叉組合與突變運算子的基因演算法中，對於每一個被「取樣」到的「模式」$H$ 而言，其「代表個體」出現在群體中的數量佔整個群體的期望比例由「世代」$t$ 到「世代」$t$+1 時的變化如下：

$$P(H,t+1) >= P(H,t)\frac{f(H,t)}{f(t)}[1-P_c\frac{d(H)}{L-1}(1-P(H,t)\frac{f(H,t)}{f(t)})](1-P_m)^{o(H)} \quad (1)$$

其中，$P(H, t+1)$ 表示「模式」$H$ 在「世代」$t$+1 時其「代表個體」出現在群體中的數目佔整個群體的期望比例； $f(H, t)$ 表示「模式」$H$ 在「世代」$t$ 時其出現在群體中的「代表個體」之平均適應值；$f(t)$ 表示「世代」$t$ 時群體的平均適應值；$P_c$ 表示交叉組合機率；$P_m$ 爲突變機率；$L$ 表示二元字串長度；$d(H)$ 表示「模式」$H$ 的「定義長度」；而 $o(H)$ 表示「模式」$H$ 的「階」。

換句話說，由上述的「模式」定理可知在一個使用比例性選擇、單點交叉組合與突變運算子的基因演算法中，對於那些低「階」、「定義長度」短且具有比群體平均適應值要高的「模式」而言，其「代表個體」出現在群體中的數量佔整個群體的期望比例會隨著演化「世代」的遞進而呈指數式增加。[2]

很明顯的，「模式」定理是建構在適應值而非函數值的

[2] 有關「模式」定理公式的詳細說明，建議讀者參考 Holland 之經典論著 [29]。

基礎上；只不過傳統上大家都是直接以「個體」的函數值當作其適應值。更重要的是，從「模式」定理中可以瞭解到基因演算法的調適能力主要是來自於選擇運算子，而交叉組合運算子與突變運算子扮演的只是輔助角色。因此，本文的研究重點放在選擇運算子。

## 四、研究方法

本研究採用的測試例子是文獻中常用的 Function of *Unitation*，亦即可行解 x 的函數值是只與其二元字串中「含 1 的個數」，即 *Unitation* 數 $u(x)$，有關，而與這些 1's 所在的位置無關 [22]。換句話說，當二元字串的長度爲 $L$ 時，整個搜尋空間中的 $2^L$ 個可能解的函數值最多可設爲 $a_0$、$a_1$、…、及 $a_L$ 共 $L$+1 個不同值，它們分別對映到 $u(x)$ 爲 0、1、…、及 $L$ 的二元字串。以 $L$ 等於 5 爲例，$u(x)$=0 的二元字串只有一個 (00000)，其函數值可設爲 $a_0$；$u(x)$=1 的二元字串共有 5 個（分別是 10000、01000、00100、00010、及 00001），其函數值可設爲 $a_1$；$u(x)$=2 的二元字串共有 10 個（分別是 11000、10100、10010、10001、01100、01010、01001、00110、00101、及 00011），其函數值可設爲 $a_2$；依此類推，$u(x)$=5 的二元字串只有一個（11111），其函數值可設爲 $a_5$。但爲了公平地驗證不同搜尋空間下採用不同選擇策略的基因演算法的效能，我們將測試所有可能對映的情況。換句話說，$u(x)$=0 的二元字串其函數值可不設爲 $a_0$ 而改設爲 $a_1$、$a_2$、…、或 $a_5$ 中任何一個；同理，$u(x)$=1 的二元字串其函數值可不設爲 $a_1$ 而改設爲 $a_0$、$a_2$、…、或 $a_5$ 中任何一個。這些所有可能對映的數目正好是所有可能排列的數目 $(L+1)!$ 換句話說，我們要測試所有可能地貌的搜尋空間。[3] 由於所有可能排列的數目 $(L+1)!$ 會隨著字串長度 $L$ 的增加而迅速增加；因此，本研究僅採用長度爲 5 的二元字串作爲「個體」之表示法以建構測試例子，但這並不減損本研究的有效性。理由有三，第一個理由是：本文的研究重點放在選擇運算子，而選擇運算子乃是基於「個體」的「表現型」上來運作；但字串長度 $L$ 的增加影響到的是「基因型」而非「表現型」。第二個理由是：因爲當字串長度 $L$ 增加時，表示搜尋空間的地貌可以更多樣更複雜，因此也更容易出現會使傳統基因演算法(採用方向性選擇)陷入局部最佳解困境的搜尋空間，自然給了其它採用不同選擇策略的基因演算法發揮的地方。第三個理由是：因爲長度 $L$ 的增加只是會讓我們所需測試的「問題例子」(problem instance, PI) 數目迅速如指數成長般的增加，但是對於一次只針對一個「問題例子」求解的基因演算法而言，雖然程式處理的時間或許會增加，卻是完全公平的；因爲實驗中的三種基因演算法唯一的差別只在於採用何種選擇策略，也就是說差別只在於如何將群體中每一「個體」的函數值轉換成適應值。讀者可回顧第二節研究動機中所述。

至於我們所建構的「問題例子」其函數值 $a_0$、$a_1$、…、及 $a_5$ 分別可對映成 2、4、8、16、32 或 64。換句話說，我們應該要測試所有 6!=720 個搜尋空間，也就是要測試 720 個「問題例子」。不過爲簡化起見，但不失其一般性，我們假設每一「問題例子」的最佳解均發生在 $u(x)$=5 ，即 x=11111 時。因此，共測試所有 5!(=120) 個「問題例子」。針對這 120 個「問題例子」，代號爲 $PI_1$~$PI_{120}$，我們以三種基因演算法(分別採用平衡性選擇、分裂性選擇與方向性選擇)加以測試，然後統計它們各自的「效能」；此處所謂「效能」是以能否找到最佳解爲依據。

## 五、實驗設計與結果分析

實驗中的參數設定如下：群體大小固定爲 6，單點交叉組合機率（$P_c$）分別爲 0.9、0.7、0.5、0.3、0.1 與 0，而突變機率（$P_m$）分別爲 0.001、0.01、0.05、0.1 與 0.3。換句話說，共有 30 種參數組合，而針對每種組合均分別以三種基因演算法各測試 10 次並記錄成功找到最佳解的次數。每一次均採用重新隨機初始化的一個新群體，並執行 12 世代演化。[4] 值得提醒讀者的是，本實驗係將單點交叉組合機率與突變機率兩個參數當作控制變數，只將選擇策略此參數當作解釋變數。更重要的是，我們還將基因演算法程式中的隨機亂數啓始值（random number seed）設定成相同，以確保整個實驗中的三種基因演算法只有在所採用的選擇策略上有所不同，而在其他條件上均相同。換句話說，儘管單點交叉組合與突變兩個運算子對基因演算法的搜尋結果會有影響，但在我們的實驗設計下，可以確認結果的差異是起因於

[3] 地貌（landscape）與搜尋空間（search space）其實是一體兩面的說法。

[4] 本實驗仍是採用比例性選擇（proportionate selection）並以權重式輪盤（weighted roulette wheel）挑選機制實行之 [24]。另外，本實驗採用的是標準的無重疊式（non-overlapping）的全世代交替（generational replacement）策略而非重疊式（overlapping）的部份世代交替（steady-state replacement）策略 [34]。

採用了不同的選擇策略。其中，SGA 代表的是採用平衡性選擇的基因演算法，DGA 代表的是採用分裂性選擇的基因演算法，而 TGA 代表的是採用方向性選擇的傳統基因演算法。[5]

以下我們將針對參數組合與「問題例子」兩個分析單位分別從微觀與巨觀的層面來說明。換句話說，我們有四個角度來分析：參數微觀而問題巨觀、參數微觀且問題微觀、參數巨觀而問題微觀、與參數巨觀且問題巨觀。所謂參數微觀是指以每一種參數組合下的實驗結果爲資料單位，而參數巨觀是指以 30 種參數組合下的實驗結果之平均值爲資料單位；同理，問題微觀是指以每一個「問題例子」的實驗結果爲資料單位，而問題巨觀是指以 120 個「問題例子」下的實驗結果之平均值爲資料單位。我們是利用 MATLAB 軟體工具中 Friedman's ANOVA（analysis of variance）來檢定虛無假設（null hypothesis）$H_0$：這三組樣本係來自於三個平均數相同的母群體，與對立假設（alternative hypothesis） $H_a$：至少有兩個母群體其平均數是不同的。[6]在虛無假設成立下，已知 Friedman's ANOVA 檢定統計量爲自由度( degree of freedom）2 的卡方（chi-squared）分配，而我們所設定的顯著水準（significance level）爲 5%。

**（一）參數微觀而問題巨觀**

表 1 中列出三種基因演算法分別在每一種參數組合下，針對所有 120 個「問題例子」測試的實驗平均結果。如果我們令 $hit_k$ 代表的是，對第 $k$ 個「問題例子」而言，在 10 次的測試中可以找到最佳解的次數，則 *avg.*代表的是這 120 個 $hit_k$ 的平均數，而 *std.* 代表的是這 120 個 $hit_k$ 的標準差；*LT3* 代表的是有多少個「問題例子」其 $hit_k$ 是小於等於 3 次。很明顯的，我們希望 *avg.*越大越好，而 *std.* 與 *LT3* 越小越好。

根據表 1 中的資料，Friedman's ANOVA 檢定結果所算得之 *P-value* 爲 0.0065，因此我們可以拒絕虛無假設；從而推論：不同的選擇策略的確會影響基因演算法找到最佳解的能力。事實上，由表 1 中的資料可看出 SGA 與 DGA 表現普遍稍優於 TGA；僅在少數情況下（如粗體字部份），TGA 表現爲較佳。

**（二）參數微觀且問題微觀**

除此之外，我們也根據實驗結果的三組樣本資料，也就是三種基因演算法針對每一個「問題例子」分別在 30 種參數組合下各測試 10 次中所可以找到最佳解的次數，作了統計分析。整個分析的結果顯示，在 120 個「問題例子」中有高達 112 個「問題例子」所算得之 *P-value* 是小於 0.05，其平均值爲 8.13E-4。[7]換句話說，針對這 112 個「問題例子」，我們可以拒絕虛無假設；從而推論：不同的選擇策略的確會影響基因演算法找到最佳解的能力。限於篇幅，此處僅列出第一個「問題例子」之三個樣本資料，如表 2 所示。根據表 2 的樣本資料，Friedman's ANOVA 檢定結果所算得之 *P-value* 爲 1.41426E-10，如表 3 所示。至於另外那 8 個「問題例子」，其函數值如表 4 所示，我們則無法拒絕虛無假設。爲了讓讀者更容易瞭解這些「問題例子」所對應之搜尋空間的地貌，特別將其中之一 $PI_6$ 繪於圖 1。

**（三）參數巨觀而問題微觀**

爲了讓讀者清楚地看到這三種基因演算法各有所長，我們以圖 2 至圖 4 來顯示這三種基因演算法在每一個「問題例子」上的表現（也就是在 30 種參數組合下每 10 次測試中成功找到最佳解的次數，即 $hit_k$，之平均數）。顯而易見的，不同的選擇策略適用於不同地貌的搜尋空間。以第一個「問題例子」而言，SGA 平均每 10 次測試中成功找到最佳解的次數爲 3.57 次，DGA 爲 5.77 次，而 TGA 爲 7.6 次。但若以第五個「問題例子」來看，SGA 平均每 10 次測試中成功找到最佳解的次數爲 6.57 次，DGA 爲 2.53 次，而 TGA 爲 3.2 次。這兩個「問題例子」的函數值，如表 5 所示。而根據圖 2 至圖 4 的資料，Friedman's ANOVA 檢定結果所算得之 *P-value* 爲 0.0326，如表 6 所示，因此我們可以拒絕虛無假設。從而推論：不同的選擇策略的確會影響基因演算法找到最佳解的能力。

**（四）參數巨觀且問題巨觀**

如果我們再將圖 2 至圖 4 的資料分別加總以求出平均數，則可得到針對所有 120 個「問題例子」，在所有 30 種參數組合下實驗結果之總平均值。也就是說，三種基因演算法在每 10 次測試中成功找到最佳解的次數分別爲 SGA：4 次、DGA：4.17 次與 TGA：3.81 次。儘管，這個結果令人雀躍，

[5] 三者差異僅在函數值轉換爲適應值的原則不同，亦即是差異僅在於視甚麼樣的可行解爲「適者」。

[6] 一般使用 ANOVA 檢定時需假設樣本之母體爲常態分配，而我們並無充分證據顯示本實驗所得之樣本滿足此假設，因此，採用無需此假設的 Friedman's ANOVA 檢定，這是一種無母數（nonparametric）檢定[28]。

[7] 即使所設定的顯著水準爲 1%，仍有高達 110 個「問題例子」所算得之 *P-value* 是小於 0.01。

**表 1. 三種基因演算法分別測試 120 個測試例子的實驗結果**

| | $P_m$ | 0.3 | | | 0.1 | | | 0.05 | | | 0.01 | | | 0.001 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $P_c$ | | *avg.* | *std.* | *LT3* | *avg.* | *std.* | *LT3* | *avg.* | *std.* | *LT3* | *avg.* | *std.* | *LT3* | *avg.* | *std.* | *LT3* |
| **0.9** | SGA | 8.44 | 1.26 | 0 | 6.14 | 2.2 | 20 | 4.28 | 2.65 | 46 | 2.60 | 1.84 | 83 | 2.18 | 1.62 | 95 |
| | DGA | 8.29 | 1.38 | 0 | 6.34 | 2.12 | 13 | 4.41 | 2.56 | 55 | 2.50 | 1.91 | 84 | 2.03 | 1.24 | 101 |
| | TGA | 8.20 | 1.70 | 2 | 5.67 | 2.79 | 34 | 3.90 | 3.16 | 65 | 2.42 | 2.47 | **82** | 2.15 | 2.28 | **86** |
| **0.7** | SGA | 8.12 | 1.35 | 0 | 5.37 | 2.51 | 32 | 3.87 | 2.39 | 60 | 1.66 | 1.43 | 101 | 1.72 | 1.14 | 112 |
| | DGA | 8.24 | 1.46 | 0 | 5.87 | 2.31 | 20 | 3.85 | 2.01 | 58 | 1.20 | 0.86 | 117 | 1.25 | 1.01 | 117 |
| | TGA | 7.71 | 1.82 | 2 | 4.98 | 3.01 | 42 | 3.16 | 2.85 | 73 | 1.36 | 1.40 | 109 | 1.40 | 1.31 | 113 |
| **0.5** | SGA | 8.65 | 1.19 | 0 | 5.25 | 2.64 | 35 | 3.40 | 2.27 | 69 | 1.97 | 1.86 | 89 | 1.29 | 1.21 | 115 |
| | DGA | 8.84 | 1.04 | 0 | 5.52 | 2.34 | 28 | 4.39 | 2.26 | 41 | 1.98 | 1.95 | 91 | 1.52 | 1.34 | 108 |
| | TGA | 8.42 | 1.14 | 0 | 4.84 | 2.85 | 46 | 3.03 | 2.82 | 82 | **2.04** | 2.43 | 87 | 1.40 | 1.62 | **101** |
| **0.3** | SGA | 8.49 | 1.34 | 0 | 4.90 | 2.26 | 39 | 3.19 | 2.53 | 82 | 1.67 | 1.40 | 104 | 0.83 | 1.05 | 120 |
| | DGA | 8.85 | 1.00 | 0 | 5.88 | 2.45 | 22 | 3.85 | 2.52 | 66 | 1.60 | 1.24 | 113 | 0.95 | 0.95 | 120 |
| | TGA | 8.40 | 1.35 | 0 | 4.65 | 2.76 | 51 | 3.07 | 3.12 | 81 | 1.50 | 2.18 | **94** | **1.04** | 1.65 | **103** |
| **0.1** | SGA | 8.15 | 1.58 | 1 | 4.31 | 2.48 | 48 | 2.90 | 1.93 | 82 | 1.25 | 1.37 | 110 | 0.66 | 0.84 | 120 |
| | DGA | 8.24 | 1.34 | 0 | 5.39 | 1.86 | 19 | 3.40 | 2.00 | 69 | 0.8 | 0.87 | 120 | 0.17 | 0.38 | 120 |
| | TGA | 7.75 | 1.70 | 2 | 4.48 | 3.22 | 64 | 3.19 | 2.44 | 78 | 1.00 | 1.35 | 111 | 0.33 | 0.62 | 120 |
| **0** | SGA | 8.66 | 1.19 | 0 | 4.65 | 2.66 | 42 | 3.65 | 2.17 | 61 | 1.26 | 1.50 | 103 | 0.28 | 0.45 | 120 |
| | DGA | 8.56 | 1.22 | 0 | 5.45 | 2.34 | 26 | 4.35 | 2.13 | 47 | 1.07 | 1.09 | 117 | 0.25 | 0.43 | 120 |
| | TGA | 8.37 | 1.55 | 1 | 4.31 | 2.99 | 54 | 3.61 | 2.98 | 66 | **1.58** | 2.19 | **92** | **0.40** | 0.49 | 120 |

**表 2. 以三種基因演算法分別求解第一個「問題例子」$PI_1$ 時之表現（即 $hit_1$）**

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **SGA** | 0 | 0 | 4 | 4 | 10 | 0 | 1 | 2 | 2 | 8 | 0 | 1 | 2 | 5 | 9 |
| **DGA** | 1 | 3 | 7 | 8 | 10 | 0 | 0 | 4 | 9 | 9 | 1 | 2 | 9 | 10 | 10 |
| **TGA** | 1 | 5 | 8 | 10 | 10 | 2 | 4 | 8 | 10 | 10 | 3 | 4 | 9 | 10 | 10 |
| **SGA** | 1 | 0 | 2 | 2 | 9 | 2 | 2 | 4 | 5 | 8 | 3 | 3 | 4 | 5 | 9 |
| **DGA** | 2 | 3 | 8 | 9 | 10 | 2 | 1 | 7 | 9 | 7 | 3 | 4 | 8 | 8 | 9 |
| **TGA** | 6 | 8 | 10 | 9 | 10 | 6 | 7 | 9 | 8 | 9 | 6 | 7 | 10 | 9 | 10 |

**表 3. Friedman's ANOVA 檢定之結果（以第一個「問題例子」$PI_1$ 爲例）**

| Friedman's ANOVA Table | | | | | |
|---|---|---|---|---|---|
| **Source** | **SS** | **df** | **MS** | **Chi-sq** | **Prob>Chi-sq** |
| **Columns** | 40.0667 | 2 | 20.0333 | 45.36 | 1.41426e-010 |
| **Error** | 12.9333 | 58 | 0.2230 | | |
| **Total** | 53.0000 | 89 | | | |

註：SS: sum of square, df: degree of freedom, MS: mean square (=SS/df).

**表 4. 在 5% 的顯著水準下無法拒絕虛無假設 $H_0$ 的 8 個「問題例子」的函數值，依 $u(x)$ 而定**

| | $u(x)=0$ | $u(x)=1$ | $u(x)=2$ | $u(x)=3$ | $u(x)=4$ | $u(x)=5$ |
|---|---|---|---|---|---|---|
| $PI_6$ | 2 | 4 | 32 | 16 | 8 | 64 |
| $PI_{21}$ | 2 | 32 | 8 | 4 | 16 | 64 |
| $PI_{22}$ | 2 | 32 | 8 | 16 | 4 | 64 |
| $PI_{30}$ | 4 | 2 | 32 | 16 | 8 | 64 |
| $PI_{45}$ | 4 | 32 | 8 | 2 | 16 | 64 |
| $PI_{67}$ | 8 | 32 | 2 | 4 | 16 | 64 |
| $PI_{69}$ | 8 | 32 | 4 | 2 | 16 | 64 |
| $PI_{86}$ | 16 | 8 | 2 | 32 | 4 | 64 |

此時我們仍不宜驟下斷語。畢竟，還需要更多的實驗證明；而這正是我們未來要繼續努力的方向。

不過，從以上四個角度的分析結果，我們可以肯定的得到一個結論：不同的選擇策略的確會影響基因演算法找到最佳解的能力。換句話說，我們的實驗提供了更充份的證據顯示：使用基因演算法並不一定非得採用方向性選擇。至於，不同的選擇策略彼此之間有何關聯，則需進一步分析。

因此，以圖 2 至圖 4 的資料，我們首先進行了相關性分析。結果發現，DGA 與 TGA 兩者的相關係數爲 0.636，SGA 與 TGA 兩者的相關係數爲 -0.245，而 DGA 與 SGA 相關係

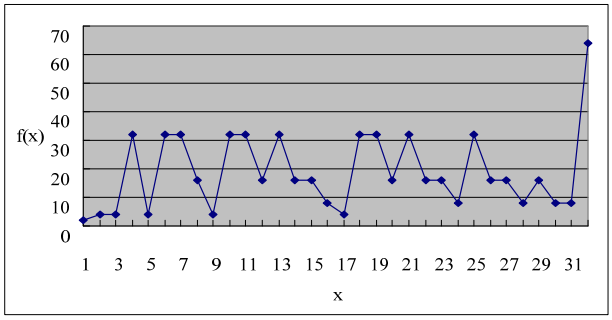


**圖 1. 「問題例子」$PI_6$ 的函數值**

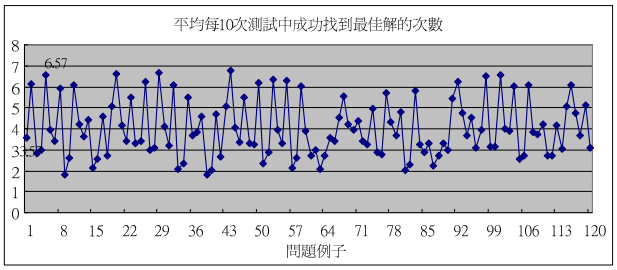


**圖 2. SGA 在 120 個「問題例子」的表現（以 30 個參數組合之平均）**

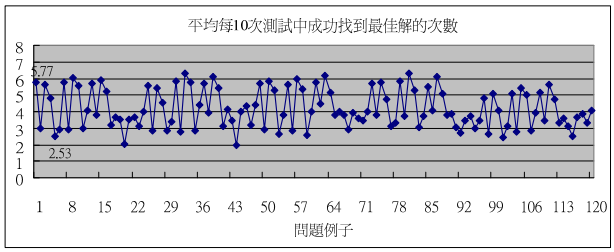


**圖 3. DGA 在 120 個「問題例子」的表現（以 30 個參數組合之平均）**

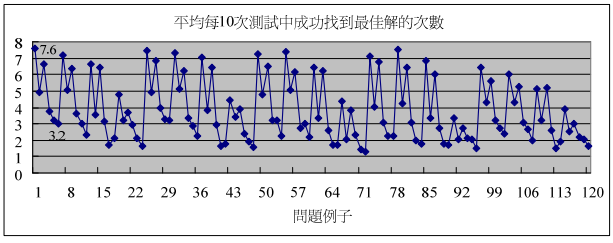


**圖 4. TGA 在 120 個「問題例子」的表現（以 30 個參數組合之平均）**

數爲 -0.85。因此，我們推測：分裂性選擇對於方向性選擇而言具有替代性，而分裂性選擇與平衡性選擇兩者具之間有

**表 5. 第一個「問題例子」與第五個「問題例子」的函數值，依 *u*(x) 而定**

| | *u*(x)=0 | *u*(x)=1 | *u*(x)=2 | *u*(x)=3 | *u*(x)=4 | *u*(x)=5 |
|---|---|---|---|---|---|---|
| $PI_1$ | 2 | 4 | 32 | 16 | 8 | 64 |
| $PI_5$ | 2 | 32 | 8 | 4 | 16 | 64 |

**表 6. 參數巨觀而問題微觀下 Friedman's ANOVA 檢定之結果**

| Friedman's ANOVA Table | | | | | |
|---|---|---|---|---|---|
| **Source** | **SS** | **df** | **MS** | **Chi-sq** | **Prob>Chi-sq** |
| **Columns** | 6.817 | 2 | 3.40833 | 6.85 | 0.0326 |
| **Error** | 232.183 | 238 | 0.97556 | | |
| **Total** | 239.000 | 359 | | | |

註：SS: sum of square, df: degree of freedom, MS: mean square (=SS/df)

很強的互補性。此處，所謂替代性是指若採用選擇策略 X 的基因演算法能找到最佳解，而採用選擇策略 Y 的基因演算法也能找到最佳解，則稱選擇策略 Y 對於選擇策略 X 具有替代性。至於，所謂互補性是指若採用選擇策略 X 的基因演算法未能找到最佳解，而採用選擇策略 Y 的基因演算法卻能找到最佳解，則稱選擇策略 Y 對於選擇策略具有互補性。此種替代性與互補性也間接驗證了第一節中有關新達爾文主義對此三類演化過程在本質上差異的描述。上述的替代性可擴充至三種選擇策略間，這是指若採用選擇策略 Z 的基因演算法能找到最佳解，而採用選擇策略 X 或 Y 的基因演算法也能找到最佳解，則稱選擇策略 X 與選擇策略 Y 對於選擇策略 Z 具有替代性。同樣地，互補性也可擴充至三種選擇策略間，這是指若採用選擇策略 Z 的基因演算法未能找到最佳解，而採用選擇策略 X 或 Y 的基因演算法卻能找到最佳解，則稱選擇策略 X 與選擇策略 Y 對於選擇策略 Z 具有互補性。

接著，爲了進一步驗證不同選擇策略之間的替代性與互補性，我們整理了一些統計數字，列在表 7 中。表中各個欄位的涵意分別如下：A 欄代表在 TGA 能解之的「問題例子」中 SGA 也能解之的比率；B 欄代表在 TGA 能解之的「問題例子」中 DGA 也能解之的比率；C 欄代表 TGA 未能解之而 SGA 或 DGA 卻能解之的「問題例子」個數；D 欄代表 TGA 未能解之的「問題例子」個數；而 E 欄代表 C/D 的比例；F 欄代表 TGA 能解之而 SGA 未能解之的「問題例子」個數；也就是說，F 欄應該等於（120-D 欄）乘上（1-A 欄）；而 G 欄代表 TGA 能解之而 DGA 未能解之的「問題例子」

表 7. 不同選擇策略間之替代性與互補性

| $P_c$ | $P_m$ | A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.9 | **0.300** | 1.00 | 1.00 | 0 | 0 | N.A. | 0 | 0 | 0 | N.A. | N.A. | N.A. | N.A. |
| | **0.100** | 1.00 | 1.00 | 0 | 0 | N.A. | 0 | 0 | 0 | N.A. | N.A. | N.A. | N.A. |
| | **0.050** | 0.93 | 0.96 | 15 | 15 | 1.00 | 7 | 4 | 0 | 1.00 | 1.00 | 1.00 | 1.00 |
| | **0.010** | 0.85 | 0.92 | 34 | 35 | 0.97 | 13 | 7 | 0 | 0.68 | 0.95 | 0.50 | 0.93 |
| | **0.001** | 0.86 | 1.00 | 40 | 41 | 0.98 | 11 | 0 | 0 | 0.15 | 0.95 | 0.00 | 0.89 |
| 0.7 | **0.300** | 1.00 | 1.00 | 0 | 0 | N.A. | 0 | 0 | 0 | N.A. | N.A. | N.A. | N.A. |
| | **0.100** | 0.98 | 1.00 | 7 | 7 | 1.00 | 2 | 0 | 0 | 1.00 | 1.00 | N.A. | N.A. |
| | **0.050** | 0.96 | 0.98 | 24 | 24 | 1.00 | 4 | 2 | 0 | 1.00 | 1.00 | 1.00 | 1.00 |
| | **0.010** | 0.70 | 0.88 | 39 | 39 | 1.00 | 24 | 10 | 1 | 0.80 | 0.97 | 0.56 | 0.94 |
| | **0.001** | 0.86 | 0.72 | 37 | 37 | 1.00 | 12 | 23 | 1 | 0.75 | 0.94 | 0.74 | 0.97 |
| 0.5 | **0.300** | 1.00 | 1.00 | 0 | 0 | N.A. | 0 | 0 | 0 | N.A. | N.A. | N.A. | N.A. |
| | **0.100** | 0.98 | 1.00 | 3 | 3 | 1.00 | 2 | 0 | 0 | 1.00 | 1.00 | N.A. | N.A. |
| | **0.050** | 0.93 | 0.98 | 19 | 19 | 1.00 | 7 | 2 | 0 | 1.00 | 1.00 | 1.00 | 1.00 |
| | **0.010** | 0.69 | 0.73 | 46 | 49 | 0.94 | 22 | 19 | 0 | 0.73 | 0.90 | 0.49 | 0.92 |
| | **0.001** | 0.71 | 0.74 | 49 | 52 | 0.94 | 20 | 18 | 0 | 0.51 | 0.92 | 0.47 | 0.92 |
| 0.3 | **0.300** | 1.00 | 1.00 | 0 | 0 | N.A. | 0 | 0 | 0 | N.A. | N.A. | N.A. | N.A. |
| | **0.100** | 0.98 | 1.00 | 1 | 1 | 1.00 | 2 | 0 | 0 | 1.00 | 1.00 | N.A. | N.A. |
| | **0.050** | 0.90 | 0.93 | 29 | 29 | 1.00 | 9 | 6 | 0 | 0.90 | 1.00 | 0.67 | 1.00 |
| | **0.010** | 0.73 | 0.98 | 71 | 71 | 1.00 | 13 | 1 | 1 | 0.48 | 0.96 | 0.03 | 0.97 |
| | **0.001** | 0.18 | 0.90 | 75 | 80 | 0.94 | 33 | 4 | 3 | 0.50 | 0.88 | 0.80 | 0.84 |
| 0.1 | **0.300** | 1.00 | 1.00 | 0 | 0 | N.A. | 0 | 0 | 0 | N.A. | N.A. | N.A. | N.A. |
| | **0.100** | 0.97 | 1.00 | 3 | 3 | 1.00 | 3 | 0 | 0 | 1.00 | 1.00 | N.A. | N.A. |
| | **0.050** | 0.95 | 0.95 | 10 | 10 | 1.00 | 6 | 6 | 0 | 1.00 | 1.00 | 1.00 | 1.00 |
| | **0.010** | 0.44 | 0.54 | 66 | 68 | 0.97 | 29 | 24 | 10* | 0.59 | 0.76 | 0.44 | 0.78 |
| | **0.001** | 0.10 | 0.03 | 68 | 90 | 0.76* | 27 | 29 | 26* | 0.40 | 0.28* | 0.29 | 0.51 |
| 0 | **0.300** | 1.00 | 1.00 | 0 | 0 | N.A. | 0 | 0 | 0 | N.A. | N.A. | N.A. | N.A. |
| | **0.100** | 0.96 | 0.99 | 10 | 10 | 1.00 | 4 | 1 | 1 | 0.80 | 1.00 | 1.00 | 1.00 |
| | **0.050** | 0.96 | 0.99 | 31 | 31 | 1.00 | 4 | 1 | 0 | 1.00 | 1.00 | 1.00 | 1.00 |
| | **0.010** | 0.53 | 0.72 | 57 | 67 | 0.85* | 25 | 15 | 0 | 0.47 | 0.79 | 0.33 | 0.76 |
| | **0.001** | 0.21 | 0.50 | 30 | 72 | 0.42* | 38 | 24 | 16* | 0.44 | 0.33* | 0.27 | 0.36 |

個數；也就是說，G 欄應該等於（120-D 欄）乘上（1-B 欄）；至於 H 欄，則代表 TGA 能解之而 SGA 與 DGA 均未能解之的「問題例子」個數。最後，I 欄代表 SGA 未能解之而 TGA 卻能解之的比例；J 欄代表 SGA 未能解之而 DGA 卻能解之的比例；K 欄代表 DGA 未能解之而 TGA 卻能解之的比例；L 欄代表 DGA 未能解之而 SGA 卻能解之的比例。上述所謂“能解之”意指能找到最佳解，也就是說 $hit_k$ 不等於零。另外，表中標示為 N.A.者代表計算該比例值時，分母恰好為零，故顯示為無法提供（not available）。

首先，從表 7 的 A 欄與 B 欄中我們可以發現 SGA 與 DGA 兩者在突變機率大於等於 0.05 時不論單點交叉組合機率為多少，其效能均十分接近 TGA。只有在突變機率為 0.01 與 0.001 時效能稍差，主要原因是如此低的突變機率使得整個群體會發生突變的字元還不到一個（6×5×0.01=0.3 與 6×5×0.001=0.03），自然較無法將平衡性選擇及分裂性選擇的預期功效發揮出來。進一步比較 A 欄與 B 欄後，可看出相較於平衡性選擇而言，分裂性選擇對於方向性選擇是較具有替代性。其次，從表 7 的 E 欄中我們可以發現絕大多數情況下，在 TGA 未能解之的「問題例子」中 SGA 或 DGA 卻能解的比例均接近 1，只有在少數情況下（E 欄中數字後有*者）效能稍差，其原因同上所述。因此，我們可說分裂性選擇與平衡性選擇兩者對於方向性選擇而言具有互補性。接著，從表 7 的 F、G 二欄中我們可以發現儘管有些「問題例子」是 TGA 能解之而 SGA 未能解之，或 TGA 能解之

而 DGA 未能解之，但只有在少數情況下（H 欄中數字後有*者）會使得 SGA 與 DGA 兩者均未能解之，其原因同上所述。換句話說，在多數情況下，TGA 能解之的「問題例子」中，SGA 或 DGA 也能解之。因此，我們可說分裂性選擇與平衡性選擇兩者對於方向性選擇而言具有替代性。最後，根據表 7 的 I、J 二欄之對比，我們發現在絕大多數情況下（除了 J 欄中數字後有*者例外），SGA 未能解之的「問題例子」中，DGA 卻能解之的比例均大於 TGA 卻能解之的比例；而根據表 7 的 K、L 二欄之對比，我們發現在所有情況下，DGA 未能解之的「問題例子」中，SGA 卻能解之的比例均大於 TGA 卻能解之的比例。因此，我們可說，相較於與方向性選擇之間而言，分裂性選擇與平衡性選擇兩者之間較具有互補性。

## 六、結論與建議

由於基因演算法的調適能力主要是來自於選擇運算子，因此，我們把研究重點放在選擇運算子上。本研究利用 MATLAB 軟體工具的 Friedman's ANOVA，在顯著水準（significance level）爲 5% 的標準下，針對參數組合與「問題例子」兩個分析單位分別從微觀與巨觀的層面進行檢定；得到一個結論：不同的選擇策略的確會影響基因演算法找到最佳解的能力。本研究亦顯示了，相較於平衡性選擇而言，分裂性選擇對於方向性選擇是較具有替代性；而分裂性選擇與平衡性選擇兩者對於方向性選擇而言具有互補性與替代性；同時也顯示了，相較於與方向性選擇之間而言，分裂性選擇與平衡性選擇兩者之間較具有互補性。總而言之，方向性選擇並非基因演算法的必然策略。不過，實際上當我們面對一個最佳化問題時，事先並不會知道它的搜尋空間爲何；因此，不應只侷限於某一種特定的選擇策略。所以，如何整合分裂性選擇與平衡性選擇共同來解決最佳化問題是下階段我們要繼續探討的方向。另一方面，進行理論上之分析與針對實際的應用問題加以測試也都是我們未來要努力的目標。

## 參考文獻

1. 田方治、游智凱（民 89），運用雙影像技術求取 3D 座標資訊--基因演算法之應用，工業工程學刊，17(3)，233-239。
2. 吳泰熙、吳奕樺、張欽智（民 95），以基因演算法求解單原片方形物件排列問題，科學與工程技術期刊，2(3)，75-83。
3. 林萍珍、陳稼興、林文修（民 89），遺傳演算法在使用者導向的投資組合選擇之應用，資訊管理學報，7(1)，155-171。
4. 林維垣（民 89），在結構性變遷時間序列下調適性遺傳演算法與投資策略組合--電腦模擬與分析，真理財經學報，4，35-73。
5. 林豐澤（民 90），設計一個基因演算法以密文攻擊方式破解 Vigenere 密碼，電腦學刊，13(4)，71-81。
6. 林志交、曾義星（民 93），以基因演算法解算 CSG 模型與影像之最佳套合，航測及遙測學刊，9(1)，27-40。
7. 邱昭彰、李安邦（民 87），遺傳演算法在發展股市投資專家知識規則之研究，資管評論，8，21-37。
8. 陳素雯（民 92），預測客戶潛在的消費行爲--使用以遺傳演算法爲基礎的組合多層分類器，資訊、科技與社會學報，3(2)，39-52。
9. 郭定（民 97），基因演算法成功的一個關鍵因素：何謂『適者』？，智慧科技與應用統計學報，6(1)，13-28。
10. 張玉鈍、諶承正（民 91），運用系統模擬與基因演算法執行限制理論於流線型生產，大葉學報，11(2)，99-106。
11. 黃昭平、張克章、侯永昌（民 89），基因演算法於密碼學之應用與研究，資訊安全通訊，6(2)，20-36。
12. 黃國禎、林宗良（民 94），基因演算法運用於分散式網路分級資訊管理之研究，資訊管理學報，12(1)，171-193。
13. 游原龍、蘇民揚（民 91），遺傳演算法在資訊安全的應用，中華民國資訊學會通訊，5(3)，65-80。
14. 蔡孟伸、黃金鷗（民 94），基於 Fisherface 以及基因演算法之人臉辨識系統，第十屆人工智慧與應用研討會，國立高雄大學。
15. 劉文卿、范饒耀（民 86），遺傳演算法在財務預測之應用，資訊管理研究，1(2)，49-66。
16. 劉書助、蕭榮興（民 92），股價預測模式中變數選取之研究，管理研究學報，2(1)，77-101。
17. 劉振隆、胡家銘（民 93），以基因演算法實現高效率之指紋辨識系統，第十屆資訊管理暨實務研討會，國立台中技術學院。
18. Bäck, T., and H. P. Schwefel (1993) An overview of

evolutionary algorithms for parameter optimization. *Evolutionary Computation*, 1(1), 1-23.

19. Borenstein, Y. and R. Poli (2004) Fitness distribution and GA hardness. Proceedings of the 8th International Conference on Parallel Problem Solving from Nature, Birmingham, UK.
20. Caruana, R. and J. D. Schaffer (1988) Representation and hidden bias: Gray vs. binary coding for genetic algorithms. Proceedings of the 6th International workshop on Machine Learning, Ithaca, New York.
21. Clergue, M. and P. Collard (2002) GA-hard functions built by combination of trap functions. Proceedings of the Congress on Evolution Computation 2002, Piscataway, New Jersey.
22. Deb, K. and D. E. Goldberg (1993) Analyzing deception in trap functions. In: *Proceedings of the 2nd Foundations of Genetic Algorithms*, L. Darrell Whitley, Ed. Morgan Kaufmann Publishers, San Mateo, CA.
23. Forrest, S. and M. Mitchell (1993) What makes a problem hard for a genetic algorithm? Some anomalous results and their explanation. *Machine Learning*, 13, 285-319.
24. Goldberg, D. E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA.
25. Goldberg, D. E., K. Deb and B. Korb (1990) Messy genetic algorithms revisited: Studies in mixed size and scale. *Complex Systems*, 6, 333-362.
26. Goldberg, D. E. and K. Deb (1991) A comparative analysis of selection schemes used in genetic algorithms. In: *Proceedings of the 1st Workshop on the Foundations of Genetic Algorithms and Classifier Systems*, G. J. E. Rawlins, Ed. Morgan Kaufmann Publishers, San Mateo, CA.
27. Grefenstette, J. J. and J. E. Baker (1989) How genetic algorithms work: A critical look at implicit parallelism. Proceedings of the 3rd International Conference on Genetic algorithms, Fairfax, Virginia.
28. Higgins, J. J. (2004) *Introduction to Modern Nonparametric Statistics*, Thomson Learning, Pacific Grove, CA.
29. Holland, J. H. (1975) *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, MI.
30. Koza, J. R. (1992) *Genetic Programming: On the Programming of Computer by Natural Selection*, MIT Press, Cambridge, MA.
31. Kuo, T. and S. Y. Hwang (1996) A genetic algorithm with dsruptive selection. *IEEE Transactions on Systems, Man, and Cybernetics*, 26(2), 299-307.
32. Kuo, T. and S. Y. Hwang (1997) Using disruptive selection to maintain diversity in genetic algorithms. *Applied Intelligence*, 7(3), 257-267.
33. Manly, B. F. J. (1984) *The Statistics of Natural Selection on Animal Populations*, Chapman and Hall, London, UK.
34. Whitley, L.D. and J. Kauth (1988) GENITOR: A different genetic algorithm. Proceedings of the Rocky Mountain Conference on Artificial Intelligence, Denver, CO.
35. Wright, A. H. (1991) Genetic algorithms for real parameter optimization. In: *Proceedings of the 1st Workshop on the Foundations of Genetic Algorithms and Classifier Systems*, G. J. E. Rawlins, Ed. Morgan Kaufmann Publishers, San Mateo, CA.

**收件：97.07.07　修正：97.10.20　接受：97.11.24**