

以調變式策略改善機率分佈估計演算法-以 TSP 問題為例

尹邦嚴

國立暨南國際大學資訊管理研究所

pyyin@ncnu.edu.tw

吳昕澧

國立暨南國際大學資訊管理研究所

austin77725@gmail.com

摘要

傳統基因演算法在處理變數間連結(Variable Linkage)時，會遇到好的基因砌塊(Building Blocks)被單點交配的交配方式破壞而無法保留，而新興的演化式演算法中的機率分佈估計演算法則利用統計的方式對母體空間內個體分佈建立機率模型，再由此機率模型抽樣產生下一代母體，取代傳統交配、突變的演化機制，以避免好的基因砌塊被破壞。本研究提出多個改善機率分佈估計演算法的調變式策略，並以旅行銷售員問題(Traveling Salesman Problem)來實證分析，實驗結果顯示我們所提出的演算法比傳統的機率分佈估計演算法更有效率及效能。

關鍵字：基因砌塊、機率分佈估計演算法、機率模型、遺傳演算法、調變式策略

壹、 簡介

機率分佈估計演算法(estimation of distribution algorithm, EDA)是演化式演算法(Evolution Algorithm, EA)領域中新興的演算法，過去曾被稱作基於母體建立機率模型的基因演算法(Probabilistic Model Building Genetic Algorithm, PMBGA)，由於過去傳統基因演算法(Genetic Algorithm)在處理某些連結問題時會出現基因砌塊(Building Blocks)結構緊密的基因被演化機制打斷，而造成陷入區域最佳解的困境，為了克服這樣的困境，針對母體空間個體(solution)分佈建立機率模型的演算法成為熱門研究主題。

演化式演算法(Evolutionary Algorithm, EA)是人工智慧研究領域之一，結合演化生物學(Evolutionary Biology)和電腦科學(Computer Science)的領域，方法多從自然界中觀察生物演化的機制所得，並且在各個最佳化領域中相對較具有彈性、效率、強健的搜尋方法。演化式演算法主要有基因演算法(Genetic Algorithm, GAs)、演化式規劃(Evolutionary Programming, EP)、演化策略(Evolution Strategy, ES)、遺傳規劃(Genetic Programming, GP)。其中基因演算法是運用最廣泛也是最具有代表性之方法。

基因演算法由密西根大學 Holland 教授(1975)所提出，根據達爾文的「演化論」為基礎，以「物競天擇，適者生存」的生物進化方式，適者生存不適者淘汰，利用電腦模擬出自然界的演化步驟，透過編碼解碼、適合度評估、天擇(Selection)、交配(Crossover)、突變(Mutation)的機制反覆實作(Deb & Goldberg 1991)，直到滿足停止條件或求得最佳解。但是對於處理連結問題(Linkage Problem)(Harik & Goldberg et.al. 1997; Chen 2005)，只透過簡單的選擇和交配這兩項機制又要同時保留基因中重要的連結資訊，可能無法發揮足夠的效用，而導致無法求得最佳解，因為連結問題是指在基因之間存在著緊密關係，又稱為基因砌塊(Building Blocks, BBs)，當基因砌塊的連結關係較鬆散時，就會發生欺騙問題(Deceptive Problem)，使得傳統基因演算法會陷入區域最佳解中。根據 Holland 的 Schema theorem(Holland 1992)，短的(shorter defining length)、低階的(lower order)和高於平均適應值(above average fitness)的 Schema，其子代個數會呈現指數型增加，這現象也符合基因砌塊假設(Building Block hypothesis)(Goldberg 1989)。

EDA 最早的概念源自於 1994 年提出的 PBIL(Baluja 1994)，而 EDA 名詞的產生則是在 1996 年首次出現(Mühlenbein & Paaß 1996)，其流程圖如圖 1 所示。首先以隨機的方式產生 N 組個體(solution)做為初始母體，並評估每個個體的適合度。根據選取的原則(通常是根據個體的適合度)從母體中選出 M ($M \leq N$) 個個體(solution)，再用這 M 個個體(solution)建立機率分佈模型。接著從機率分佈模型中抽樣產生子代的母體成員，依照更新原則取代原本親代母體中的全部成員，成為下一代新的母體，以此反覆演化直到滿足停止條件為止。

過去所發表過的 EDA 文獻，根據編碼方式主要可以分成二進位編碼以及實數編

碼。二進位編碼著重在解離散問題，根據機率模型的不同可以再分為 Population based incremental learning(PBIL) (Baluja 1994)、Univariate marginal distribution algorithm(UMDA) (Mühlenbein & Paaß 1996)、Compact genetic algorithm(cGA)(Harik & Lobo et.al. 1999)、Bayesian optimization algorithm(BOA)(Pelikan & Goldberg 1999)。

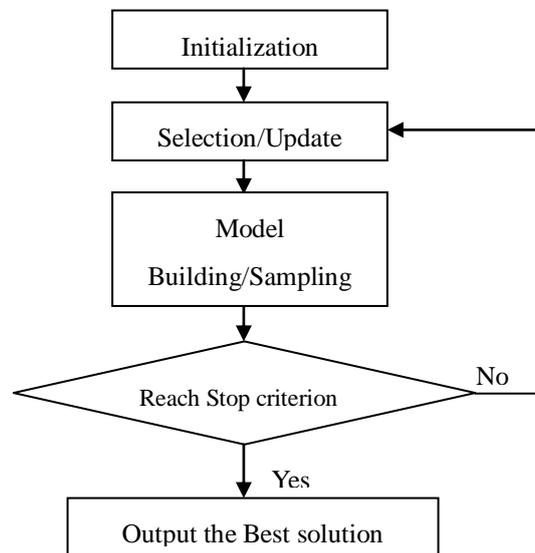


圖 1 基本型 EDA

PBIL、UMDA 的初始母體是以給每一維度一個固定的機率向量代表個體(solution)的每一個維度值為 1 的機率是多少，舉例說明在 PBIL 裡初始個體中每個維度值為 1 的機率都設成 0.5；相對的為 0 的機率也是 0.5 以此機率隨機產生初始母體，PBIL、UMDA 在天擇(selection)的機制中皆使用截斷天擇(truncate selection)，也就是直接從母體中選取適應值排名在前面的最好的個體(solution)來建立機率分佈模型，PBIL 以機率向量來進行演化，再根據學習率更新演化下一代的機率分佈，學習率的概念在於以不同權重去結合過去經驗和現在的估計結果來更新機率模型。UMDA 跟 PBIL 不同的地方在於機率的更新方法，UMDA 是估計母體中個體(solution)每個位置的聯合機率分佈來做更新的動作。

cGA、BOA 則是以隨機的方式產生母體。BOA 在天擇(selection)的機制使用截斷天擇(truncate selection)和上一小節描述一樣，cGA 則是使用(tournament)透過隨機競爭取代的方式選出較好的個體來建立機率分佈模型。cGA 在初始化以隨機產生兩個個體，更新方式則是適應值較大的個體則在機率向量加上一個固定值，反之適應值較小的個體則在機率向量剪掉相等固定值。BOA 則是先以隨機的方式產生母體，利用適應值較大的樣本建構貝式網絡，貝式網絡可表示個體(solution)的機率分佈，抽樣產生新的母體。以上的二進位編碼透過適當的量化也可以應用在連續型問題(Xiao & Yan et.al. 2009)。

實數編碼主要著重在連續型問題，大部分方法都是由離散型 EDA 方法所延伸得

來，但是連續型空間的機率分佈非常複雜，因此並非每一種離散型 EDA 都能發展出連續型。UMDAc(Larranaga & Etxeberria 2000)由 UMDA 延伸得到，PBILc(Sebag & Ducoulombier 1998)則是 PBIL 的延伸，PBILc、UMDAc 都採用高斯分配來估計連續空間機率分佈的機率模型，不同點在於更新的方式不同。解連續型問題除了高斯分配以外還有以直方圖來做為機率模型的方法。

然而在機率分佈估計演算法領域中必須用機率模型來描述變數之間的相互關係，並且隨著問題的複雜度不同而產生多種機率模型來描述變數之間的關係，過去文獻大致上分為單變數相關、雙變數相關和多變數相關，也分別都各有其代表性的文獻可以參考，以下簡單做描述。單變數相關屬於最簡單的情況可以不用考慮變數之間會有互相干預的情形發生，通常可以透過一個簡單的機率向量表示個體(solution)的分佈，代表性的演算法有：Population based incremental learning(PBIL)、Univariate marginal distribution algorithm(UMDA)、Compact genetic algorithm(cGA)。但在大部分實際問題中變數之間並不完全獨立，因此考慮到變數之間彼此之間會有相互關係，就提出了雙變數相關，代表性的演算法有：Mutual information maximization for input clustering(MIMIC)(Bonet & Isbell 1997)、Combining optimizers with mutual information trees(COMIT)(Baluja & Davies 1997)、BMDA(Pelikan & Mühlenbein 1999)，這幾種演算法所提出的機率模型可以描述雙變數間的相互關係，在本研究中所提出的方法也是屬於此類。但是雙變數相關還不足以描述複雜度非常高的最佳化問題，因此有了多變數相關專門來處理非常複雜的問題，代表性的演算法有:Extended compact genetic algorithm(ECGA)(Harik 1999)、Factorized distribution algorithms(FDA)(Mühlenbein & Mahnig 1999)、Bayesian optimization algorithm(BOA) (Pelikan & Goldberg 1999)等。我們可以從上述文獻探討以及其他參考文獻中大致上歸納出如表 1 所示，在每個步驟列出過去文獻中具有代表性的方法。

表 1 EDA 文獻之策略

Initialization	Selection/Update	Model Building/Sampling
1. Set probability vector (Baluja 1994) (H. Mühlenbein and G. Paaß 1996)	Selection	Building
2. Random (Harik G.R. & Lobo F.G. et.al. 1999) (Pelikan & Goldberg et.al. 1999)	1. Truncate selection (Baluja 1994) (H. Mühlenbein and G. Paaß 1996)	1. Joint distribution (H. Mühlenbein and G. Paaß 1996)
	2. Tournament (Harik G.R. & Lobo F.G. et.al. 1999)	2. Bayesian Network (Pelikan & Goldberg et.al. 1999)
	3. Boltzmann selection (Mühlenbein H & Mahnig T 1999)	3. Histogram (Jing Xiao & YuPing Yan et.al. 2009)
	Update	4. Gaussian distribution (Larranaga P & Etxeberria R et.al. 2000) (Sebag & Ducoulombier 1998)
	1. Learning rate (Baluja 1994)	
	2. Convergence rate (H. Mühlenbein and G. Paaß 1996)	Sampling
	3. Replace parent all or partially (all)	1. Offspring with better fitness (all)

傳統機率分佈估計演算法是從基因演算法的概念發展得來，所以大部分研究都還停留在用二進位編碼求解離散型問題，實數編碼求解連續型問題也多是從二進位編碼的基本型延伸得來，但是二進位編碼用在描述實際生活中最佳化問題確實有些困難，因此本研究中提出了新的概念，以整數型編碼求解離散型問題，並且提出多個以機率分佈估計演算法為基礎的變形，首先加入學習率(Learning Rate)和取代率(Replacement Rate)，結合擴散式搜尋法(Scatter Search)中參考解集合(Reference Set)的概念提出 EDA^*_R (EDA with Reference Set)和結合禁忌搜尋法中重新啟動(Restart)的概念提出 EDA^*_S (EDA with restart)、門檻值法 EDA^*_T (Threshold)、權重混合門檻值法 EDA^*_{WT} (Weight+threshold)，最後用旅行推銷員問題(Traveling Salesman Problem, TSP)來測試本研究所提出的演算法。

貳、 問題定義

旅行推銷員問題最早是由 Hassler Whitney(Flood 1956)於 1934 年在普林斯頓大學的研討會中提出。後續被廣泛討論和研究，並成為組合最佳化問題(Combinatorial Optimization Problem)中最具有代表性的問題之一。TSP 已被證明為困難度極高的 NP-Complete (Non-deterministic Polynomial)問題，以往用傳統整數規劃求解，計算時間複雜度會隨著節點數越多而呈現指數型成長($O(n!)$)。因此為了提升效率，啟發式演算法被廣泛運用在 TSP 問題上。

TSP 問題之定義可描述如下：有 n 個節點數(n 個城市)，銷售員要從某一個起始節點出發並拜訪過(行經)每一個節點，再回到原來的起始節點，並且每一個節點不可以被拜訪過(行經)兩次以上，目的是要找出總旅行距離最短之拜訪路徑。

TSP 數學模式可建構如下(Lawler & Lenstra et.al. 1985)：

$$\text{Minimize } \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (1)$$

$$\text{subject to } \sum_{i=1}^n x_{ij} = 1 \quad (j = 1, 2, \dots, n) \quad (2)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad (i = 1, 2, \dots, n) \quad (3)$$

$$x_{ij} = 0, 1 \forall i, j = 1, 2, \dots, n \quad (4)$$

其中 n 表示節點個數， c_{ij} 代表由 i 點至 j 點之成本。 x_{ij} 為二元變數，當 $x_{ij} = 1$ 時，表示拜訪路徑將從 i 點移動至 j 點；當 $x_{ij} = 0$ 時，表示拜訪路徑中不包含 (i, j) 之路徑。(1) 式目標函數為求總路徑成本最小化，(2)(3) 式表示限制每個節點只能被拜訪(行經)一次，(4) 式為二元決策變數限制式。

參、 研究方法

本研究所提出的方法為以機率分佈估計演算法(EDA)為基底加入學習率(Learning)和取代率(Replacement Rate)，再結合本研究中所提出的多個調變式策略(Adaptive Strategy)，調變式策略包含有參考解集合(Reference Set)、重新起始搜尋程序(Restart)、門檻值法(Threshold)和權重混和門檻值法(Weight+Threshold)，完整方法流程圖如圖 2 所示：

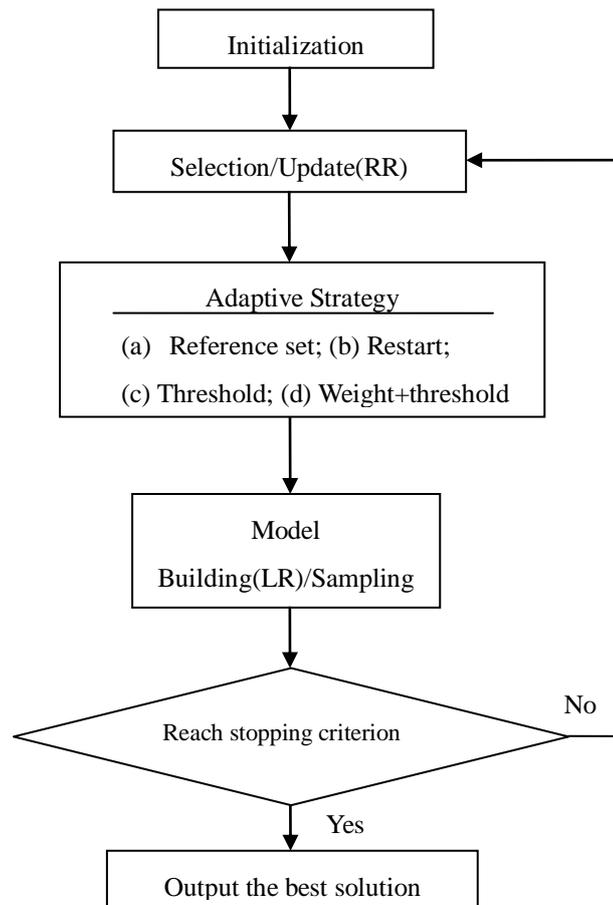


圖 2 本研究方法的流程

首先以隨機的方式初始化母體產生多組解，根據 TSP 問題計算適應值的方式來評估每一組解的適應值，接著選出適應值排名前百分之五十的解來建立機率分佈模型，從機率分佈上抽樣出子代母體成員，根據取代率多寡取代掉親代母體中的成員，在下一代建立機率分佈的同時用學習率去更新機率分佈模型，直到滿足停止條件。在更新策略中子代母體成員要對親代母體成員根據適應值好壞做競爭取代的動作，當子代中的最佳解持續無法比歷代最佳解還要更好的時候，則啟動調變式策略(Adaptive Strategy)，分別有以下幾種：Reference Set、Restart、Threshold 和 Weight+Threshold 的機制，實驗數據在第四節詳加說明，以利觀察本研究中所提出之調變式策略對 EDA 的影響。

本研究提出的四種調變式策略，是基於許多可能組合策略的前測實驗之後，留下四種最好的調變式策略，以下逐一說明。

1. 學習率(Learning Rate ; LR)及取代率(Replacement Rate ; RR)

學習率(Learning Rate)是取自 Shumeet Baluja 在 1994 年發表的 PBIL 中所提出的方法，從類神經網絡的非監督式學習方法建立學習率用在更新機率模型的公式如下：

$$probability_i = (probability_i \times (1 - LR)) + (LR \times vector_i) \quad (6)$$

在 Shumeet Baluja 發表的 PBIL 中編碼方式是以二元編碼為主，因此每一個位置上都有出現 1 的機率， $probability_i$ 所表示的意思是根據過去的經驗在 i 的位置出現 1 的機率， $vector_i$ 所表示的是目前觀察到適應值較好的解在 i 這個位置出現 1 的機率， LR 是學習率，以學習率當作權重去結合過去的學習經驗和現在的估計結果來更新機率模型。

取代率(Replacement Rate)則是本研究提出的新概念，代表基本型 EDA 中抽樣產生的子代母體成員取代掉親代母體成員的百分比，由於取代百分比的多寡會影響到上一代母體對機率分佈的影響力，譬如取代率如果很高即表示親代母體成員留下的比率較少，因此在估計機率模型的時候對機率模型的影響力相對較弱；反之則是親代母體成員留下的多，對估計機率模型的影響力較強，我們的最終目的是期望適應值佳的母體成員可以保留下來，讓下一代朝最佳解前進；適應值較差的母體成員則被取代掉，避免下一代受到干擾。

2. 參考解集合(Reference Set)

參考解集合(Reference Set)的概念來自於擴散式搜尋法(Scatter Search, SS)，SS 會記錄過去搜尋歷史中曾經出現過適應值較佳的參考解(Reference Solution)，儲存成為參考解集合，再利用隨機線性組合的方式，從數個參考解中產生新的解。本研究在 EDA 中加入此概念，目的希望在最佳解無法突破時，從參考解集合中對好的參考解做突變產生新的解，期望這些新的解可以出現比目前最佳解還要好的解，並且引導機率分佈朝向全域最佳解演化。

在本研究中加入參考解集合的概念，流程圖參考圖 2，在評估適應值的時候把目前最佳解放入參考解集合，如果參考解集合滿了那就透過競爭取代的方式，當我們當代最

佳解經過幾代演化一直無法突破歷代最佳解時，從參考解集合中挑出一定比例的參考解做 two swap 產生新的解，然後和選擇母體中前百分之五十最好的解競爭取代，留下適應值最好的解拿來更新機率模型。

3. 重新起始搜尋程序(Restart)

禁忌搜尋法(Tabu Search, TS) 為 Glover 在 1986 年所提出，主要概念是將走過的路徑用禁制串列(tabu list)記錄起來，避免搜尋到重複的區域，並且可以藉由禁制串列的長度不同，做出搜尋範圍大小的變化。

本研究在 EDA 中加入禁制搜尋法關於重新起始的概念，流程圖如圖 2，期望最佳解在未滿足停止條件以前，可以持續的突破找到更好的解，但是隨著代數演化到後期，母體中個體分佈的狀況趨近收斂，個體之間差異度很小，此時我們的演化會遇到一個瓶頸，並且很有可能陷入區域最佳解。因此，為了在個體間趨近收斂時還能維持我們的搜尋範圍，利用重新起始的概念，可以放棄演化到目前為止的所有母體成員，重新隨機產生新的母體，接著在估計新母體機率分佈時和舊有的機率分佈做結合，因此新的母體可以維持差異化，舊的機率分佈則可以保留好的經驗，讓下一代重新從機率模型抽樣的結果不會太差，並且有機會跳脫區域最佳解。

4. 門檻值法(Threshold)

在機率分佈估計演算法中，機率分佈除了要學習往最佳解的方向前進之外還必須要同時維持個體差異化(diversity)是很困難的議題，因此過去文獻中有人提出給定門檻值的方法，為了不讓在機率模型中只聚焦在機率很高的個體而忽略了其他機率低但有機會還能跳脫區域最佳解的個體，本研究將此概念結合到實驗中，在建立機率模型時給定門檻值，如果超越門檻值的話則必須將此個體的機率以比例的方式降低，並且將降低的量平均分配給其他個體的機率，來達到避免陷在機率非常高的個體而忽略了其他能夠再進步的機會。

5. 權重混合門檻值法(Weight+Threshold)

在個體(solution)演化的過程中我們希望好的個體能夠對於機率分佈具有較大的影響能力，以藉此能夠加速搜尋範圍朝最佳解方向前進，因此對於分數(fitness)高的個體(solution)我們以呈比例的方式給予權重，分數越高將會得到越大的權重值因而對機率分佈造成越大的影響力；反之亦然，同時我們混合門檻值法以避免權重將搜尋範圍導向區域最佳解(local optima)而忽略了其他可能往全域最佳解(global optima)的機會。

肆、 實驗結果

本節將分成三個部分來探討，第一部分介紹本研究方法的參數最佳化過程，第二部分則是測試本研究提出的各個調變式策略的效能，在此學習率(LR)都設為 10%，而取代率(RR)都設為 30%，並測試資料為 TSP 的二十個標竿問題，分別是 bays29、att48、berlin52、eil51、eil76、pr76、st70、kroA100、kroB100、kroC100、lin105、pr124、pr144、pr152、rat195、kroA200、kroB200、pr226、tsp225、lin318，每個標竿問題都獨立實驗 30 次做統計分析。第三部分則是用關鍵事件分析(epoch analysis)來檢視我們所加入的參

考解集合和重新起始的機制是否確實產生效用。

1. 參數最佳化

(1) 學習率(LR)和取代率(RR)

在本研究中加入兩個可調變參數，分別為學習率(Learning Rate)和取代率(Replacement Rate)，學習率的作用在於我們以權重的概念去結合過去經驗和現在的估計結果來更新我們的機率模型，針對新的機率模型做抽樣產生子代的母體成員，子代母體中成員要對親代母體中成員做取代時，則要根據取代率做競爭取代的動作。實驗結果顯示當學習率和取代率為極端值時，最佳解的品質表現不好，譬如當學習率與取代率皆是100%的時候代表完全放棄過去經驗學習的結果；反之若學習率與取代率皆是0%則會造成演化停滯，本研究針對兩個參數值結合效果做測試，LR從0到100%，取代率從10%到100%，最終所得到的最佳參數值則是當學習率是10%和取代率是30%的結果品質最好。

(2) 重新啟動時間

本研究中的關鍵事件是採用到目前為止搜尋歷史最佳解有若干代沒有被當代最佳解更新改善時，我們就要啟動參考解集合(Reference Set)或者重新起始搜尋程序(Restart)的機制，經由實驗測試結果我們可以找出這兩項機制啟始時機相對於問題規模大小的對應關係，如圖3所示。

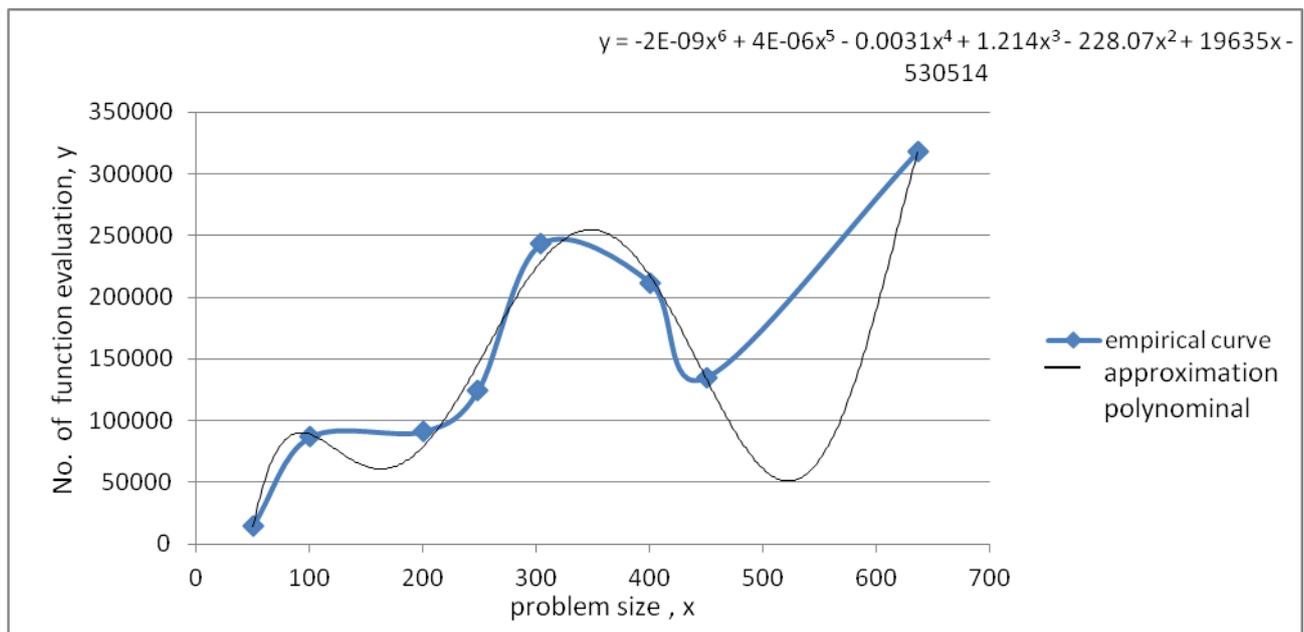


圖3 Refset 和 Restart 重啟時機

圖3所示X軸表示母體大小、Y軸表示歷代最佳解停滯多少次function evaluations未被當代最佳解做更新時，我們要啟動Reference set/Restart機制的最佳時機，經過我們實驗測試各個問題維度的實證結果，最後以六次多項式進行迴歸所得的趨勢線公式如下：

$$y = -2E - 09x^6 + 4E - 06x^5 - 0.0031x^4 + 1.214x^3 - 228.07x^2 + 19653x - 530514 \quad (7)$$

此公式可以趨近在問題陷入區域最佳解時的最佳啟動(Reference set/Restart)時機，只需把問題的維度大小帶入公式中的 x 值，即可計算出此維度的 y 值，代表若經過 y 次 function evaluation 以後，歷代最佳解還未被當代最佳解更新時，此時啟動 Reference set/Restart 機制出來的效果會是最好。

(3) 門檻值設定

在本研究方法第三節所提出的門檻值法(threshold)中該如何去設定門檻值以及縮減的量也是非常重要的問題，最初給定門檻值以及縮減量為固定值，但隨著問題維度增加固定的門檻值就會隨著統計量增大而失去效用，因此必須將門檻值以及縮減量訂為隨著問題維度大小而變動。將統計量超過門檻值的機率以問題維度大小的一定比例做縮減，並且將縮減的量平均分給其他維度的機率值，藉以在演化的過程中同時維持個體間的多樣性(diversity)。實驗結果顯示，門檻值須為維度大小的 1.5 倍，縮減量則以維度大小除以 8 的比例設定，如此對於超越門檻值的機率做縮減效果最好。

2. 效能分析

本節要把本研究中含有學習率和取代率的三個方法和標準 EDA 及 GA 做比較，表 2 至 3 則是效能比較結果：

表 2 總體效能正規化比例分析

Problem	GA	EDA	EDA*	EDA*_R	EDA*_T	EDA*_WT	EDA*_S
bays29	1.042247191	1	1.102921	1.107865169	1.093483146	1.071460674	0.922022472
att48	0.897507669	1	0.680253	0.683224693	0.671108129	0.676016104	0.674879271
berlin52	1.020222244	1	0.828011	0.817599359	0.811592752	0.818900791	0.814612663
eil51	0.837481698	1	0.667643	0.661786237	0.655929722	0.655929722	0.663412065
eil76	0.682512733	1	0.521222	0.516977929	0.513582343	0.510186757	0.514273521
pr76	0.705289773	1	0.498987	0.491473511	0.487860322	0.483533268	0.491519412
st70	0.69829222	1	0.487034	0.474383302	0.483870968	0.484503479	0.481165944
kroA100	0.599568338	1	0.508607	0.505044726	0.509153318	0.50750208	0.421786712
kroB100	0.608102625	1	0.512638	0.518711958	0.521604817	0.519706787	0.453754027
kroC100	0.591003469	1	0.513483	0.521499159	0.518371531	0.517688183	0.433180525
lin105	0.599717396	1	0.462781	0.461730766	0.4620025	0.412057753	0.391245709
pr124	0.609998338	1	0.516538	0.524639041	0.518343646	0.513309286	0.498310369
pr144	0.590929491	1	0.499953	0.50156559	0.493628047	0.482632384	0.472259044
pr152	0.631113841	1	0.490559	0.495754177	0.482009538	0.48777542	0.472067199
rat195	0.614404576	1	0.599151	0.597850581	0.57774311	0.580169873	0.567488246
kroA200	0.610583613	1	0.576164	0.583092646	0.553321484	0.559805423	0.544955585
kroB200	0.615278066	1	0.584207	0.579927191	0.558747447	0.562305029	0.547667325
pr226	0.603237807	1	0.521969	0.521562682	0.48900302	0.494498523	0.478438055
tsp225	0.626777678	1	0.604365	0.604365437	0.579207921	0.578577858	0.561599175

lin318	0.664068526	1	0.686771	0.685077613	0.63014004	0.627853235	0.580146756
average	0.692416865	1	0.593163	0.592706588	0.58053519	0.577220631	0.549239204

每個方法在不同問題中贏輸的次數都不同，贏多少跟輸多少的影響程度也是個評估難題，可能在某個問題中只有贏一些，但是在別的問題卻輸很多，很難去分辨本研究中所提出的多個方法到底哪一個方法效果最好。因此，表 2 將所有方法正規化以利我們做相互比較，正規化的方式以標準 EDA 的適應值平均數為分母，其他方法的適應值平均數為分子去計算，就可以很直接由數值大小來判斷贏多少跟輸多少的比例，最後算出平均我們可以觀察到效果最好的是 EDA^{*}_S，也就是重新起始搜尋(Restart)的效能是所有調變式策略中最好的，並且數值呈現由左往右遞減的趨勢，表示在本研究中所提的各種調變式策略都有達到預期改善效果。

表 3 名次排序表

Problem	GA	EDA	EDA [*]	EDA [*] _R	EDA [*] _T	EDA [*] _{WT}	EDA [*] _S
bays29	3	2	6	7	5	4	1
att48	6	7	4	5	1	3	2
berlin52	7	6	5	3	1	4	2
eil51	6	7	5	3	2	1	4
eil76	6	7	5	4	3	1	2
pr76	6	7	5	3	2	1	4
st70	6	7	5	1	3	4	2
kroA100	6	7	4	2	5	3	1
kroB100	6	7	2	3	5	4	1
kroC100	6	7	2	5	4	3	1
lin105	6	7	5	3	4	2	1
pr124	6	7	3	5	4	2	1
pr144	6	7	4	5	3	2	1
pr152	6	7	4	5	2	3	1
rat195	6	7	5	4	2	3	1
kroA200	6	7	4	5	2	3	1
kroB200	6	7	5	4	2	3	1
pr226	6	7	5	4	2	3	1
tsp225	6	7	5	4	3	2	1
lin318	6	7	5	4	3	2	1
average	5.9	6.7	4.4	3.95	2.9	2.65	1.5

表 3 中我們將所有的適應值平均值的名次列出，在每個問題裡由最好到最壞依序排名，最好的給予第一名，依此類推，因此所有排名的平均越小越好，從表 3 可以觀察到也是 EDA^{*}_S 贏過其他調變策略，並且在問題維度大於 100 以後全部都第一，由此可見

EDA_s 在處理大維度問題時是非常具有貢獻的。

3. 性質分析

本節要分析本研究中所提出的參考解集合和重新起始搜尋程序這兩項機制是否產生改善作用，因此利用關鍵事件分析(epoch analysis)的方式對 TSP 二十個問題做測試，將每一代的當代最佳解記錄下來，以 pr76 這個問題的 epoch analysis 為例，圖 4、圖 5 顯示參考解集合和重新起始搜尋程序這兩項機制的當代最佳解變化，其中橫軸代表的是 function evaluation 次數，藍色點代表當代最佳解的適應值(fitness)在 t function evaluations 次數時的大小，紅色點代表該策略機制被啟動的時機。

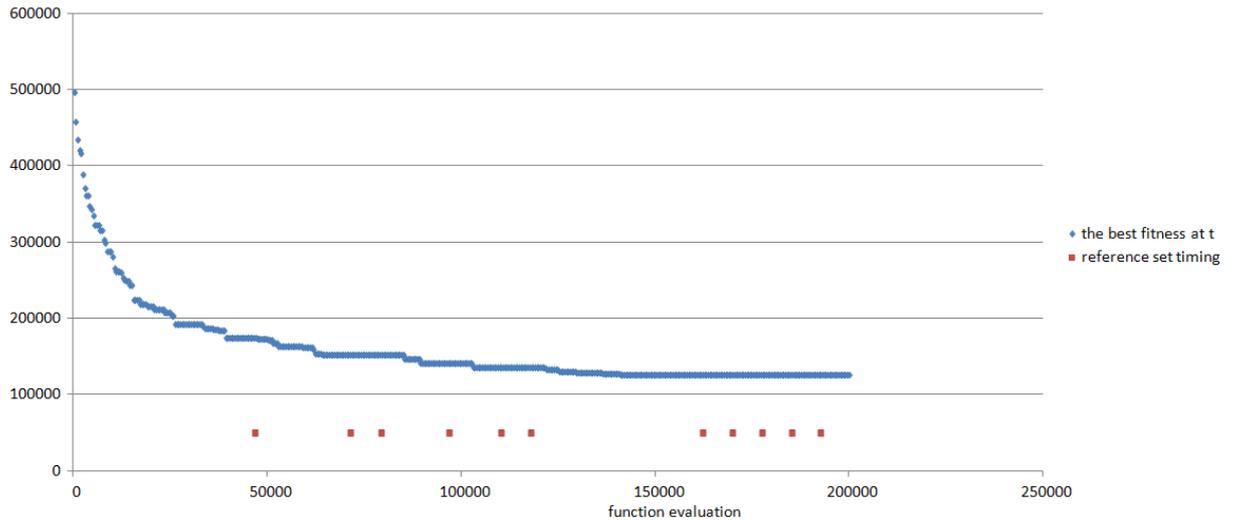


圖 4 pr76 問題在 EDA_{reference set} 的關鍵事件分析示意圖

根據圖 4 所示很明顯可以觀察出當目前的最佳解持續突破(下探)時，並不會啟動參考解集合突變的機制，在約略第 47000 次 function evaluation 時最佳解停滯在約 170000 多，且無法繼續突破小於歷史最佳解而陷入區域最佳解，因此在約略第 48500 次 function evaluation 時，系統便啟動參考解集合的機制，明顯看出經過此機制的的作用，最佳解始得跳脫區域最佳解開始繼續朝向全域最佳解前進。

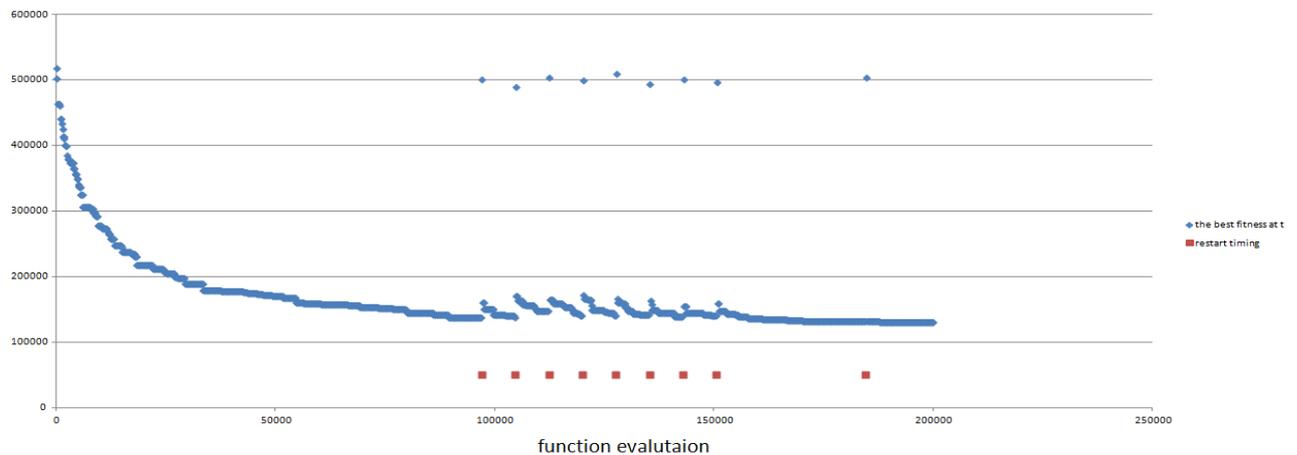


圖 5 pr76 問題在 EDA_{restart} 的關鍵事件分析示意圖

另外，根據圖 5 所示，很明顯可以觀察出在約略 90000 次 function evaluation 以前，當代最佳解持續突破，重新起始搜尋程序的機制並沒有被啟動。而著當代最佳解停滯在約 95000 多次的 function evaluation 時持續無法繼續突破，大約到第 98000 次 function evaluation 時，系統啟動 Restart 重新起始搜尋程序，由於 Restart 是以隨機的方式重新產生母體成員，因此在下一代評估母體中個體(solution)的適應值時所挑選出的當代最佳解會遠比歷史最佳解的適應值大(差)非常多，因此在圖 5 中第 99000 次 function evaluation 時所挑選出的當代最佳解適應值約 500000，而造成明顯的跳昇，但是這個機制會增加母體中個體的 diversity(差異化)，有機會讓演化跳脫之前遇到的局部最佳解，繼續改善解的品質，所以在後續的演化中才能持續改善當代最佳解。

伍、 結論

本文提出多個調變式策略以改善機率分佈估計演算法(Estimation of distribution algorithm, EDA)，本研究在傳統型 EDA 中加入學習率和取代率，接著以 reference set、restart、threshold 和 weight+threshold 等策略，透過實驗數據證明本研究所提出的新型 EDA 確實比原標準 EDA、GA 還要強健，並且突破傳統 EDA 以二進位編碼方式來描述母體空間個體的分佈情形，而改以整數型編碼的方式來實做，增加應用的彈性。

未來我們將繼續探討本研究中所提出的方法，為何反而規模小的問題中無法求得全域最佳解，以及將本研究中整數型編碼方式應用在其他問題上，並且嘗試延伸目前的雙變數相關模式至多變數相關模式。

陸、 致謝

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。

參考文獻

1. Baluja, S., and Davies, S. "Using optimal dependency-trees for combinatorial optimization: Learning the structure of thesearch space.," in: In: Proceedings of the 14th International Conference on Machine Learning., San Francisco, CA: Morgan Kaufmann, 1997, pp. 30-38.
2. De, B., J, S., Isbell, C. L., and Viola, P. "MIMIC: Finding optima by estimating probability densities.," in: Advances in Neural Information Processing Systems, Cambridge: MIT Press., 1997, pp. 424-430.
3. Flood, M. M. The Traveling-Salesman Problem , Operations Research, 1956, Vol.4,pp.61-75

4. Goldberg, D. E. Genetic Algorithms in Search, Optimization and Machine Learning USA: Addison-Wesley Longman Publishing Co., Boston, MA, 1989.
5. Harik, G. "Linkage Learning via Probabilistic Modeling in the ECGA," in: Illigal Report No. 99010, Illinois Genetic Algorithms Laboratory, Urbana-Champaign, Illinois, 1999.
6. Harik, G. R. Learning gene linkage to efficiently solve problems of bounded difficulty using genetic algorithms University of Michigan, 1997.
7. Harik, G. R., Lobo, F. G., and Goldberg, D. E. "The compact genetic algorithm," Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on, 1998, pp. 523-528.
8. Holland, J. H. Adaptation in natural and artificial systems MIT Press, MA, USA, 1992.
9. Karp, R. M. Reducibility Among Combinatorial Problems, Complexity of Computer Computations, 1972, pp. 85-104.
10. Larranaga, P., Etxeberria, R., Lozano, J. A., and Pena, J. M. "Optimization in continuous domains by learning and simulation of Gaussian networks. In: Proceedings of the 2000 Genetic and Evolutionary Computation Conference Workshop Program.," Las Vegas, Nevada, 2000, pp. 201-204.
11. Mühlenbein, H., and Mahnig, T. "Convergence theory and applications of the factorized distribution algorithm," Journal of Computing and Information Technology) 1999, p 19~32.
12. Michalewicz, Z. Genetic algorithms + data structures=Evolution Program Springer-Verlag, New York, 1996.
13. Mühlenbein, H., and Paaß, G. From recombination of genes to the estimation of distributions I. Binary parameters, 1996, pp. 178-187
14. Pelikan, M., Goldberg, D. E., and Cantu-Paz, E. "BOA: The Bayesian optimization algorithm," In: Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99, Orlando, 1999, pp. 525-532.
15. Pelikan, M., and Mühlenbein, H. "The bivariate marginal distribution algorithm," in: Advances in Soft Computing - Engineering Design and Manufacturing, Springer-Verlag, London, 1999, pp. 521-535.
16. School Of, C., and Shumeet, B. "Population Based Incremental Learning: A method for integrating genetic search based function optimization and competitive learning," Carnegie Mellon University, Pittsburgh, 1994.

17. Sebag, M., and Ducoulombier, A. "Extending population-based incremental learning to continuous search spaces," in: Proceedings of the 5th Conference on Parallel Problem Solving from Nature - PPSN V, Springer-Verlag, 1998, pp. 418-427.
18. Xiao, J., Yan, Y., and Zhang, J. "HPBILc: A histogram-based EDA for continuous optimization," Applied Mathematics and Computation (215:3) 2009, pp 973-982.
19. Ying-Ping, C. "Extending the Scalability of Linkage Learning Genetic Algorithm," 2005.

Adaptive Strategies for Improving Estimation of Distribution Algorithms - A Case Study on TSP

Peng-Yeng Yin

Department of Information Management, NCNU
pyyin@ncnu.edu.tw

Hsi-Li Wu

Department of Information Management, NCNU
austin77725@gmail.com

Abstract

Traditional genetic algorithms are unable to preserve some variable linkage relationships because the performed crossover operation may destroy building blocks. Estimation of Distribution Algorithms (EDAs) are emerging approaches in evolutionary computation. EDA builds probability distributions based on individual instances maintained in current solution population. The next population is generated by sampling a number of instances from the built probability distributions, in contrast to the crossover and mutation mechanisms usually seen in genetic algorithms. This paper proposes several EDA improvements using adaptive strategies. We testify the performance of our algorithms on Traveling Salesman Problem. The experimental result shows that our algorithms are superior to the original EDA in terms of efficiency and effectiveness.

Keywords: Building block, estimation of distribution algorithm, probabilistic model, genetic algorithm, adaptive strategy