

應用粒子群演算法於產生分類規則之研究

徐培倫

清雲科技大學電子研究所

Email: hsupl@cyu.edu.tw

詹豐澤

清雲科技大學電子研究所

Email: M9611022@cyu.edu.tw

摘要—本論文提出粒子群演算法來解決資料探勘之分類問題。由於粒子群演算法是透過適應值函數來評估分類規則，但是面對較高複雜度的問題時，除了造成搜尋時間增加之外，同時產生陷入區域最佳解之問題。為了解決上述問題，本論文將整合限制推理於粒子群演算法之架構中，透過粒子過濾機制，能夠幫助粒子群演算法快速的找出滿足限制條件之最佳解，並且整合基因演算法於粒子更新機制中，其目的是在改善粒子群演算法於區域搜尋能力不足的問題。根據實驗結果比較，本論文提出之演算法可以獲得較佳分類結果。

關鍵詞—限制推理、粒子群演算法、基因演算法

一、前言

資料經過長時間的收集與累積後，若使用者要在大量的資料中分析出特定的資訊時，則需要一套有效率的資料探勘(Data Mining)技術來提供有價值的資訊於使用者或企業作決策之用。而資料探勘則是指從大量的資料中挖掘出具有規律性、隱含性及關聯性的資訊，因此分類規則(Classification Rule)則成為資訊的表達方式之一。而在資料探勘的相關技術中，較常使用的技術為規則歸納法(Rule Induction)，此技術能夠幫助使用者或企業在大量的資料中分析出具有價值的資訊與分類規則。因此規則歸納法是將資訊以 IF-THEN 的規則型式表達，並將資料轉換成知識庫中的規則，幫助使用者在資料庫中搜尋出特定知識或樣本的能力。換言之，規則歸納法的目地是要使資料集在分類後的結果能夠符合使用者預先定義的結果。

近年來用於搜尋分類規則之相關技術，如類神經網路[3]、Frequent Patterns [13]、決策樹[6]

等都已受到學者的重視與研究。更有學者提出利用隨機方法來搜尋出分類規則問題之最佳解，如基因演算法[8]及粒子群演算法[7] [9] [11] [12] [15]。在基因演算法演化過程中是透過適應值函數來評估分類規則，利用此適應值函數來搜尋出一個最佳近似解來解決複雜資料集的分類問題。但基因演算法主要的問題是在演化的過程中，會產生較大的搜尋空間而增加計算時間上的成本。而粒子群演算法(Particles Swarm Optimization, PSO)則是以群體作為基礎[7]的演化技術，利用群體的概念來搜尋群體中最適當的解，因此具有收斂快速之優點。在近年來研究中，許多學者已針對粒子群演算法提出改善的方法，或與結合其它技術並應用於各種問題上，其實驗結果與傳統的粒子群演算法比較後，有較顯著之提升。但當 PSO 應用在較複雜問題時，會造成過早收斂而陷入區域最佳解之問題，造成搜尋時間的增加且喪失全域搜尋之概念。由於 PSO 是透過適應函數(Fitness Function) 評估可行解的好壞，並且利用更新公式將適應值較低之可行解，往最適當解之方向移動。但隨著使用對參數的設定逐漸複雜，會造成演算法針對所有使用者提出的限制做搜尋，使整個計算成本提高[7]。

基於上述問題，本篇論文提出了混合式之方法，整合限制推理機制與粒子群及基因演算法於產生分類規則之問題，幫助使用者挖掘出更準確的分類規則。而限制推理應用在描述限制與資料之間的關係與分類問題中，因此在利用粒子群演算法做分類規則挖掘前，先透過限制推理利用向前檢查法(Forward Checking)的方法

來保證區域一致性(Local Consistency)，能夠將不合法且不合理之粒子預先排除，而達到減少粒子群演算法之搜尋空間，因此粒子能夠滿足使用者預先定義的分類規則之限制，透過粒子群過濾演算法(Particle Filtering Algorithm)，可排除無意義或無關聯的分類規則，進而減少可行解之搜尋空間，並且透過粒子群演算法收斂快速的特性，達到快速解題與提高分類正確率的效果。

在本論文中除了利用限制推理技術先行排除不合理且不合法之粒子以達到縮減搜尋空間之目的，使粒子群演算法能夠在縮減後的解空間下符合使用者定義下進行搜尋，而粒子更新技術方面，則是與基因演算法更新機制相互結合作為粒子之更新方式，使粒子群演算法能夠快速挖掘出符合使用者定義之分類規則。

二、文獻探討

(一) 粒子群演算法(Particle Swarm Optimization Algorithm; PSO)

粒子群演算法是由 Kennedy 與 Eberhart[6] 於 1995 年所提出的演化計算技術，其靈感來自於動物界飛行中的鳥群及水中魚群的群體社會行為，並模仿生物界中的演化現象及物競天擇的機制所發展出來的演算法。而 Shi 與 Eberhart[14]則引進了慣性權重值 w ，使粒子保持運動的慣性，使其能有擴展搜索空間及增加其探索新區域的能力。基本的粒子群演算法是幾個單元所組成，包括下列幾點：

1. 設計問題的編碼方式。
2. 決定群體大小(Population Size)。
3. 訂出評估問題解答之適應函數(Fitness Function)。
4. 依據粒子所記錄的路線，比對粒子群中個體及群體之位置與方向，修正粒子的飛行路線與最佳解
5. 訂定結束條件(Stopping Condition)。

圖 1 為 PSO 演化流程圖，首先，PSO 會以隨機方式產生初始速度及粒子(X_{id})並散佈於解空間(Solution Space)中，針對所設計的適應值函數來計算所有粒子的適應值，之後記錄個體最佳位置(P_{id})與群體最佳位置(P_{gd})，而在每次更新時，粒子會根據個體所瀏覽過的最佳位置及群體所瀏覽過的最佳位置來計算粒子的位移量，之後分別透過粒子群演算法公式(1)及(2)，得到更新後的速度與位置。最後的結束條件則是持續到搜尋出最終指定目標或是滿足使用者所設定之最大演算世代數為止。

$$v_{id} = w \cdot v_{id} + c_1 \cdot rand() \cdot (P_{id} - x_{id}) + c_2 \cdot rand() \cdot (P_{gd} - x_{id}) \quad (1)$$

$$x_{id} = x_{id} + v_{id} \quad (2)$$

公式(1)中 $rand()$ 之值為 0 至 1 之間的隨機亂數值，學習因子 C_1 和 C_2 表示每個粒子推向 P_{id} 與 P_{gd} 位置的統計加速項之權重，通常設定為 2。而速度更新公式(1)可分為三部分，第一部分為慣性權重值(w)，其值範圍從 0.4 遞減至 0.9；第二部分為認知(Cognition)行為部分，表示粒子本身的思考；第三部分為社會(Social)行為部分，表示粒子間的訊息共享與相互合作。

分類問題之相關文獻方面，Wang[16]於 2005 年以 PSO 作為研究的基礎提出一項新的分類規則演算法，此方法能夠得到較高的正確率及較小規則集(Rule Set)。研究中所使用的更新方法是透過收縮係數法 (Constriction Factor)，根據實驗結果顯示，將 PSO 應用於分類規則中可得到較小的規則集並且改善計算時間。2006 年 Wang[17]也提出相同理論，其效果也顯示 PSO 在搜尋最佳解的能力。

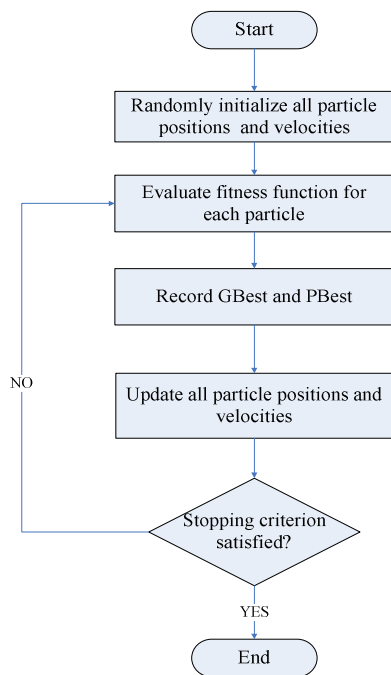


圖 1.PSO 演化之流程圖

(二) 限制滿足問題(Constraint Satisfaction Problem ;CSP)

限制滿足問題 (CSP) 是由一群變數 (Variables)與限制(Constraints)所組成，每一個變數有其值域，值域中的值都是變數可能出現的值，限制式是描述變數之間的相關性，會影響變數值，而限制滿足的問題就是要找出滿足其所有相關限制式之解答。

限制滿足問題是屬於 NP-Complete 問題，一般都缺乏適合的方法解決此問題，目前已經有學者提出不同的方法來解決 CSP 問題，有一些學者採用限制推進(Constraints Propagation)的方式縮減搜尋空間之大小或者直接使用回溯法(Backtracking)搜尋可能的解，也有學者結合樹搜尋(Tree-Search)與一致性演算法(Consistent Algorithm)有效決定一或多個適合的解答。

Nadel[9]比較幾種演算法的效能，包括產生檢查法(Generate and Test)、簡單回溯法(Simple Backtracking)、前向檢查法(Forward Checking)、部份前視法(Partial Lookahead)、完整前視法(Full

Lookahead)、及真實完整前視法(Really Full Lookahead)等，在這些演算法裡，主要差異是在解決樹狀結構的過程中，節點之間的一致性程度，除了產生檢查法外，其他方法都是採用混合式技術。主要的特性描述如後；每當一個新的值(Value)被指定到一個變數裡，其他未被分配到的變數則透過相關的限制式檢查，可過濾刪除不能滿足限制式之所有的值，而使得剩餘範圍之可能的解答值均能滿足相關的限制式，進而達成一致性的關係。如果，這些變數的可能範圍值變成空的，則限制式之間發生衝突(Contradiction)的現象，可利用回溯演算法來解決變數無解的問題。

在相關文獻方面，Barnier 與 Brisset[1]提出一項整合基因演算法與限制式滿足技術的混合式系統。此方法已被應用於解決車輛工作路線排程(Vehicle Routing)與無線電廣播頻率分配(Radio Link Frequency Assignment)最佳化問題上。基本上此方法藉由基因演算法解決限制滿足問題以減少搜尋空間，並以限制式滿足為基礎的推論來改善基因演算法的計算效率。Kowalczyk[6]提出將限制式滿足的觀念運用在基因演算法中。但目前只有少數的研究是應用限制推理的技術來改善粒子群演算法的計算效率，對於如何將問題限制與使用者需求整合在限制式網路中則缺乏有系統之描述。

三、系統建置

近年來有學者將限制推理機制與基因演算法作結合，對於改善基因演算法在求解複雜問題時所面臨的缺點，如搜尋最適解時間增長與落於區域最佳解之問題，其實驗結果有相當大的效益。因此本論文提出 CBPSOGA 演算法係利用粒子群演算法及基因演算法之結合，透過粒子過濾機制(Particle Filtering)的加入，幫助粒子在演化過程中，更快速搜尋至最適解，改良因複雜問題或維度變多所造成的演化停滯，且

利用粒子過濾演算法針對滿足限制條件之條件式但結果屬性不滿足之規則做替換，並且配合 PSO 運算技術及搜尋最適解之能力，不僅可縮小搜尋空間，也可加速搜尋速度。

本論文所提出的研究架構，由粒子群演算法(PSO)與粒子過濾演算法(Particle Filtering Algorithm)為基礎，建構成限制式粒子群演算法(Constraint-Based Particle Swarm Optimization Algorithm；CBPSO)模組，為了改善粒子群的效能，本論文提出了二個更新機制來改善此問題，更新機制分別為：粒子群演算法更新機制(PSO Approach)與基因演算法更新機制(GA Approach)。

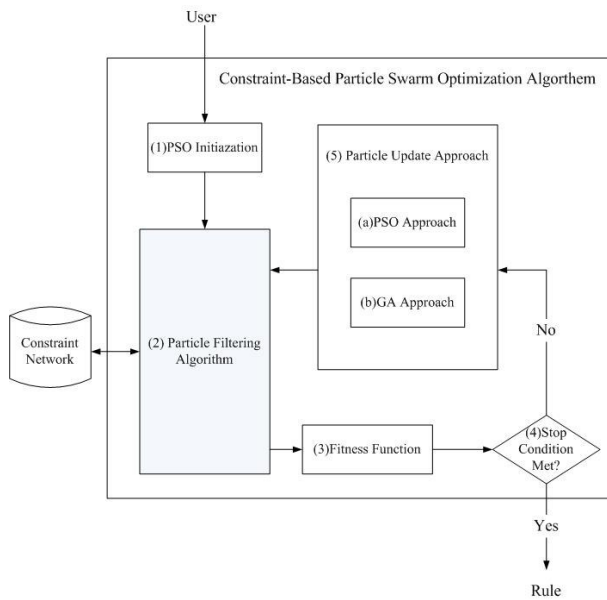


圖 2. CBPSO 之分類問題研究架構圖

研究架構圖如圖 2 所示，先由 PSO 初始模組按照初始參數隨機產生粒子，接著透過粒子過濾演算法將每個粒子所表示之分類規則過濾成合理且合法之分類規則，再經由適應函數之計算，判斷是否達成終止條件，若未達到終止條件則進入粒子更新機制中更新速度(Velocity)及位置(Position)，且進入粒子過濾演算法檢查所產生之粒子，在粒子初始與演化期間，都必須

經過粒子過濾演算法來將每個粒子過濾成合理且合法之粒子，當粒子經過更新機制後，即計算該粒子之適應值並記錄 P_{id} 及 P_{gd} ，直到滿足限制條件後才終止。

在粒子裡，我們定義局部變數被分配到數值的集合稱為 I_p ， I_p 由兩個集合所組成：一個是已分配到數值的集合 $\{x_1 = v_1, \dots, x_p = v_p\}$ ，以及未被分配到數值的集合 $\{x_{p+1}, \dots, x_m\}$ ，而 I_0 是一個無任何變數分配到數值的集合， I_m 為所有 m 個變數分配到數值的集合。

在 CBPSO 的架構裡，為了要限制粒子建構中的每一個位置值，在 PSO 的初始族群以及飛行過程中產生的族群需經由限制推理及位置值挑選機制來處理。圖 3 為粒子過濾過程的示意圖，基本上 CBPSO 是利用過濾粒子演算法來執行限制推理，主要的技術為向前檢查法(Forward Checking)[22]，其基本觀念包含使用限制條件本身的資訊找出合法的位置值，同時也參考到限制網路來限制位置值的有效範圍，這些被限制之變數，則透過相關的限制式來限制其它剩餘下來的位置合法範圍值。

傳統的 PSO 主要工作在完全已分配好的位置值環境裡，而 CBPSO 是工作在只有局部分配的位置值上，合併過濾技術在粒子的建構步驟裡，而分配位置值的過程是對於每一個位置 X_p 分配一個 V_p 值，此過程可以用增加一元限制式(Unary Constraint) $X_p = V_p$ 到限制網路中。

在增加一元限制式 $X_p = V_p$ 到向前檢查法的過程之後，有三種情況可能發生：

1. 無解(No-Solution)：位置變數的值域變成空集合。

- 可行解(Solution)：所有的值域被減少到只剩餘一個，則有一合適的可行解。
- 未知(Unknown)：至少有一個值域存在兩個或以上的值，表示分配位置變數值的過程還未完成。詳細的粒子過濾演算法的虛擬碼可由下列說明：

The Particle Filtering Algorithm

Function Particle-filtering ($\langle X, D, C \rangle, I_p$)

- $E \leftarrow$ the initial set of unary constraints derived from I_p
- $p \leftarrow p+1$ (i.e., the current variable; x_p)
- repeat**
- result \leftarrow Forward Checking ($\langle X, D, C \cup E \rangle, x_p$)
- switch** (result)
- case** No-Solution:
- $x_j \leftarrow$ randomly selected from the original domain S_{x_j} in the remaining unassigned position $x_j, j=p, \dots, m$.
- return** the total instantiation (i.e., the infeasible particle)
- case** Solution:
- $I_m \leftarrow$ a singleton solution
- return** the total instantiation (i.e., the feasible particle)
- case** Unknown:
- $x_p \leftarrow$ any value v_p selected from S_{x_p}
- $E \leftarrow E \cup \{x_p=v_p\}$
- endswitch**
- until** false

在粒子過濾演算法裡，限制網路 $\langle X, D, C \rangle$ 中

的限制式集合 C 應該要被滿足。 I_p 是局部位置變數被分配到數值的集合，表示只有局部的位置被分配到位置值。

粒子過濾演算法開始是由指定單一限制式 E 的集合，這個集合是由 I_p 決定，主要是從無任何位置變數分配到數值的集合 I_0 到完整分配位置值的 I_m ，過程實質上是應用向前檢查法用於 $C \cup E$ 而導致下列三種可能的結果：

- 無解：表示 $C \cup E$ 是不一致的且 E 是一個衝突的分配變數值， E 不可被延伸且隱含著沒有位置值可以建構一適合的粒子，而回傳一個不適合的粒子。
- 有解：表示一個單一解已經找到，演算法終止且回傳全部分配到的位置變數值(也就是回傳一合法的粒子)。
- 未知：表示粒子過濾演算法試著增加單一限制式來產生下一個分配的位置值來擴大集合 E 。

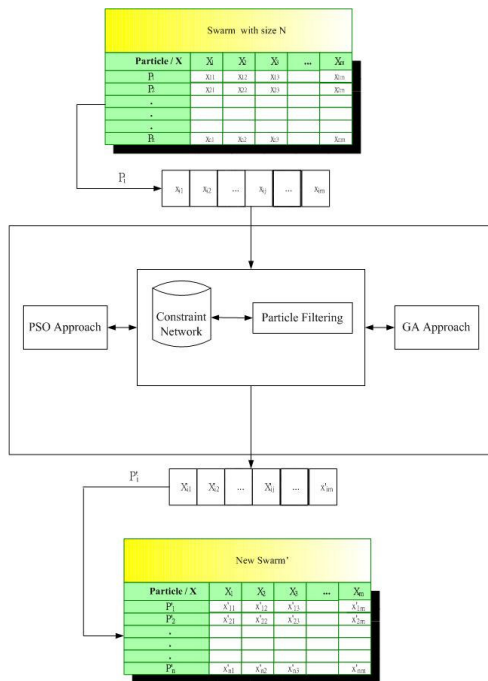


圖 3. 粒子過濾示意圖

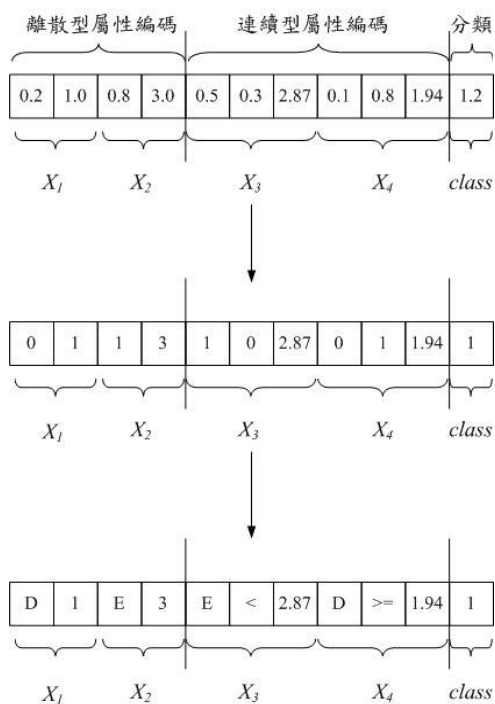
四、分類問題之規則設計

此章節說明本論文中所使用之相關技術細節與使用範例說明方式，限制推理型之粒子群演算法於產生分類規則之流程如下：

(一) 粒子編碼(Particle Encoding)

粒子的編碼設計方面，於本論文中將一個粒子設為一個規則集，在資料集方面，將資料分為連續型(Continuous)資料、離散型(Discrete)

資料與分類結果(Class)：一個離散屬性具備兩個子粒子(SubParticle)，分別為(1) 屬性是否出現(Enabled/Disabled) (2) 屬性值 (value)；而一個連續型屬性具備三個子粒子，分別為(a) 屬性是否出現 (Enabled/Disabled) (b) 運算子 (Operator) (c) 屬性值 (value)；運算子的設計是針對連續型屬性為主，其符號表示為" $>=$ "或" $<$ "。



Rule = {If $X_2 = 3$ AND $X_3 < 2.87$ THEN Class = 1}

圖 4. 粒子群編碼範例

在本論文的編碼方式中，Enable/Disable 代表屬性是否出現於規則條件中，範圍值在 0~1 之間並且採取四捨五入方式，以隨機方式產生第一代，若值介於小於 0.5 之間即此屬性不出現 (Disable) 於規則條件中，反之介於 0.5 ~ 1.0 之間，則出現 (Enable) 於規則條件中。運算元 (Operator) 則是屬於連續型屬性的編碼格式，其範圍值介於 0 ~ 1 之間，如範圍值為 0 ~ 0.4 時代表此屬性的運算子為" $<$ "，反之則為" $>=$ "，而其中所產生的屬性值皆不會超出各個屬性之範圍值內。

(二) 適應函數 (Fitness Evaluation)

在本論文中，一個粒子代表一組分類規則集，因此 PSO 的適應函數則是以一組分類規則之正確率作為評估標準。若以 m 條規則， n 個屬性為例，則完整規則集如下：

```

IF cond1,1 AND ... AND cond1,n THEN Class = C1
Else IF cond2,1 AND ... AND cond2,n THEN Class = C2
.
.
.
Else IF condi,1 AND ... AND condi,n THEN Class = Ci
.
.
.
Else IF condm,1 AND ... AND condm,n THEN Class = Cm
Else Unknown Classification Results

```

其中 $cond_{i,j}$ 表示第 i 條規則中的第 j 個屬性， C_i 表示第 i 條規則之分類結果。

(三) 終止條件 (Stop Condition)

當飛行到達設定次數或在一定的飛行次數中改善效率低於設定值 1% 時停止，而本論文中終止條件則是設定當飛行次數達 500 次時停止演化。

(四) 粒子更新機制 (Particle Update Approach)

在傳統的 PSO 更新機制中，所使用的粒子更新方式為參考個體及群體最佳解並輔以權重值先更新其速度，之後再利用更新的速度來更新位置，由於粒子群與基因演算法主要是透過適應值函數來評估分類規則，但是當問題的複雜度提高時，除了搜尋時間會相對提高外，且有可能會陷入區域最佳解之困境。基於上述之缺點，本論文提出二種更新機制來改善此問題，更新機制分別為：粒子群演算法更新機制 (PSO Approach) 與混合式更新機制 (Hybrid approach)，二種更新機制分別於後續說明。

1. 粒子群演算法更新機制 (PSO Approach)

粒子群演算法更新機制如圖 5 所示，在進入 PSO 公式運算時，每個規則屬性都有其初始飛行速度與位置，利用公式(1)求得更新後的速

度後，再利用公式(2)來達到更新位置的目的，利用更新後的位置達到更新規則集，之後再將粒子與最佳規則集作相互比較，保留相同位置的規則屬性值後再重新更新粒子位置，若優於個體最佳解，則其值則更新為 P_{id} ；若優於群體最佳解，則其值則更新為 P_{gd} 。

而本論文與傳統 PSO 不同之處在於會將最佳之規則集與每個粒子相互比較，保留相同的規則集，其目的在於要保留最佳規則集之特徵，使每個粒子能夠保留最佳規則集的優點，有更大的機會搜尋出最佳規則集。其後使用粒子過濾演算法來確保其規則集之合法性，將規則集放入粒子過濾演算法中求出合法之分類規則，其後重新計算其適應值並將適應值放入粒子中更新。

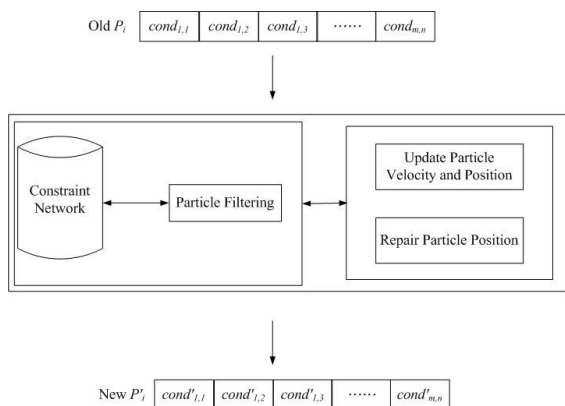


圖 5. 粒子群演算法更新機制示意圖

2. 混合更新機制(Hybrid Approach)

混合更新機制如圖 6 所示，先將舊的 Swarm 中以隨機產生的比例函數 Ψ 選擇出執行 GA Approach 之粒子，將隨機選擇出的粒子透過基因演算法之更新機制更新，更新方式是分別與粒子本身的 PBest 與 Swarm 中的 GBest 作交配(Crossover)，並且分別比較交配過後粒子的適應值，突變(Mutation)元則是將 Enable 轉換成 Disable，運算元(Operator)則是將 " $=$ " 轉換成 " $>$ "，反之。若突變元為屬性值時，則隨機產生，取代的方式是以適應值較好者取代舊粒子

成為新的粒子，透過此更新方式以得到適應值較高的 Swarm，重複此動作直到滿足中止條件為止。

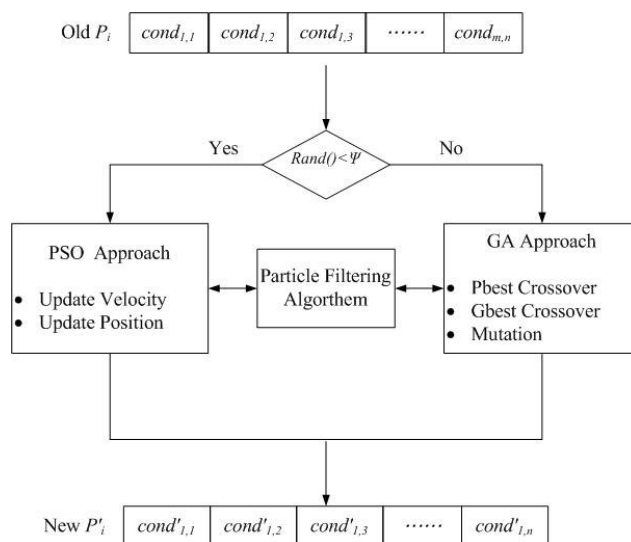


圖 6. 混合更新機制示意圖

五、實驗結果

本論文中所使用之資料集心臟病資料集(Heart Disease)，此資料集之出處為 Machine Learning Repository，其資料集之缺值資料(Missing value)，本論文皆予以移除。Heart Disease 此資料集共 303 筆資料，其中輸入屬性包含 7 個離散型(Discrete)屬性 6 個連續型(Continuous)屬性，結果屬性為離散型資料型態，分為健康(Healthy)及不健康(Sick)兩類別；其中有 160 筆為健康類別、135 筆為不健康類別及 8 筆缺值資料(Missing Value)，表 4.1 為心臟病資料集中離散型與連續型屬性之說明。

(一) 實驗設計

本論文之參數設定方面，將學習因子 C_1 與 C_2 設定為 2，而 Data Set 之規則集為 20 條，粒子數量為 50 個粒子，飛行次數設為 500 次。在限制式方面，我們透過心臟病資料分別來設計限制條件 "**Sex=Male and Chol \leq 210**" 與 "**Sex=Male and Chol \leq 276**" 以驗證 CBPSOGA

的效率。圖中的 X 軸代表 PSO、CBPSO、CBGA 與 CBPSOGA 的飛行代數，Y 軸則為分類規則之正確率。

本論文針對資料集使用交叉驗證實驗法 (K-fold Cross Validation) 作為評估標準[27]。交叉驗證實驗法是一種用來衡量分類準確率以及分類系統之可靠度的評估方法，其方法為將資料集分成大小相等而且彼此之間互斥的 n 等份，每次以其中一個等分當作測試資料，其它等分則當作訓練資料，以此方式重複 n 次的訓練與測試，並且將這 n 次的結果平均即可求得最後分類的準確率。舉例來說，在本論文中將資料集分成 5 個等份(n=5)，如同上述此方法將需要執行 5 次，在第一次的訓練與測試中，以 fold 1 當測試資料，其它 4 個 fold 則當作訓練資料，以這樣的方式來挖掘規則集以建立模型並進而計算正確率，以此類推在執行 5 次之後將會得到 5 組準確率，將這 5 組準確率取平均值即為此資料集的分類準確率。由於這樣的方式能確保每筆資料都有成為測試資料機會，所以所有的測試資料均具備了獨立的特性，資料彼此之間相依性的影響也會因此降低。

(二)實驗結果分析

表 1 為限制式“Sex=Malse and Chol<=210”下之 PSO、CBPSO、CBGA 與 CBPSOGA 實驗結果，分別為每個限制式於不同更新機制之演算法飛行 500 次所得到之結果，並且取出飛行 100、200、300、400 及 500 次後的結果置於表中，而此限制式下之飛行 500 次後的訓練正確率分別為 77.20%、79.13%、80.14%、84.07%，而測試正確率 PSO 最高為 74.21%，CBPSO 最高為 77.54%，CBGA 最高為 77.78%，CBPSOGA 最高為 79.15%，可看出所提出之方法效能高於其它三者。在圖 7 限制式“Sex=Male and Chol<=210”中可得知，PSO 在第 500 次飛行次數後所得到的正確率大約是 CBPSOGA 中飛行 10 次至 20 次之間的正確率。

表 2 為限制式“Sex=Male and Chol<=276”下之 PSO、CBPSO、CBGA 與 CBPSOGA 實驗結果，分別為每個限制式於不同更新機制之演算法飛行 500 次所得到之結果，並且取出飛行 100、200、300、400 及 500 次後的結果置於表中，而此限制式下之飛行 500 次後的訓練正確率分別為 77.20%、79.13%、85.16、87.03%，而測試正確率 PSO 最高為 74.21%，CBPSO 最高為 77.54%，CBGA 最高為 80.25%，CBPSOGA 最高為 81.25%，可看出所提出之方法效能高於其它三者。在圖 8 限制式“Sex=Male and Chol<=276”中可得知，PSO 在第 500 次飛行次數後所得到的正確率大約是 CBPSOGA 中飛行 40 次至 50 次之間的正確率。

表 1 限制式“Sex=Male and Chol<=210”之結果

Sex=Malse and Chol<=210										
generations	100		200		300		400		500	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
PSO	76.52	68.72	76.52	74.19	77.20	71.50	77.20	73.65	77.20	74.21
CBPSO	78.30	74.24	78.99	76.97	78.99	76.43	78.99	77.54	79.13	77.52
CBGA	79.45	76.58	80.14	77.00	80.14	77.78	80.14	77.78	80.14	77.76
CBPSOGA	82.84	78.58	83.94	79.12	84.07	79.15	84.07	78.03	84.07	78.48

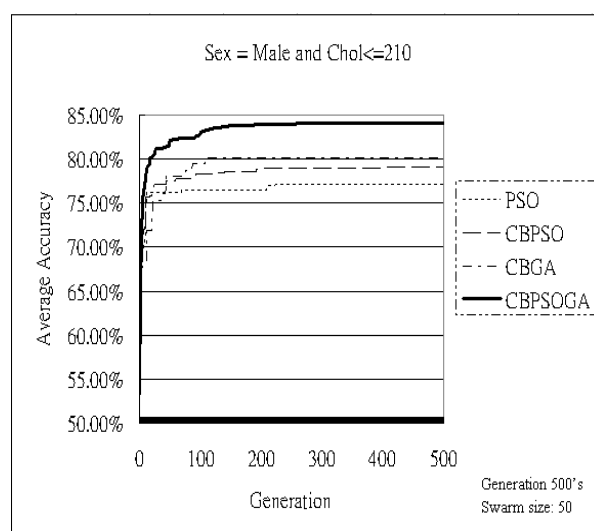


圖 7.限制式“Sex=Male and Chol<=210”之訓練實驗結果圖

表 2.限制式“Sex=Male and Chol<=276”之結果

Sex=Male and Chol<=276										
generations	100		200		300		400		500	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
PSO	74.53	68.75	75.62	70.63	76.72	68.13	76.72	74.38	77.03	66.88
CBPSO	77.35	76.88	79.53	77.50	79.69	76.88	79.85	76.25	80.00	78.75
CBGA	81.51	78.03	83.07	77.08	84.90	79.17	85.16	80.25	85.16	80.25
CBPSOGA	83.28	78.13	85.91	80.00	86.88	81.25	87.03	80.63	87.03	81.25

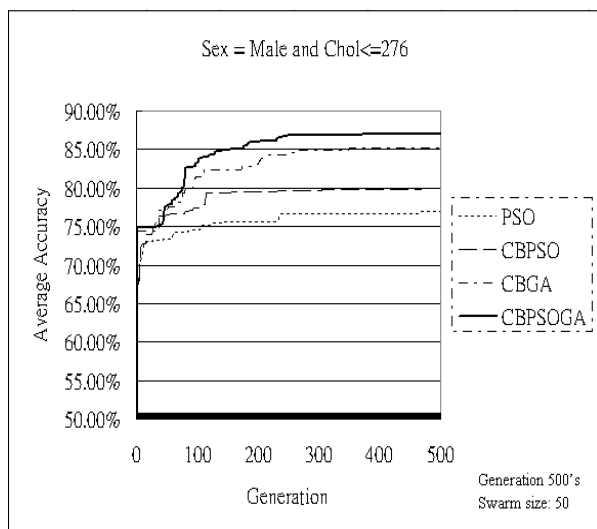


圖 8 限制式“Sex=Male and Chol<=276”之訓練實驗結果圖

六、結論

本論文提出限制推理型之粒子群與基因演算法於產生分類規則。主要方式是整合限制推理及粒子更新機制於粒子群演算法架構中，用來減少粒子產生的搜尋空間，使得粒子群演算法能夠更快速的找出符合限制之最佳解。而在粒子更新機制中，混合了基因演算法更新機制以補足粒子群演算法於區域搜尋能力不足的缺點，使粒子之搜尋範圍及效能得到良好的提升。實驗方面分別與傳統粒子群演算法比較，實驗結果顯示所提出的 CBPSOGA 能夠在相同的飛行次數中得到較高正確率的分類結果。

七、參考文獻

- [1] N. Barnier and P. Brisset, “Optimization by hybridization of a genetic algorithm and constraint satisfaction techniques,” in *Proc. IEEE World Congr. Computational Intelligence—Evolutionary Computation*, pp. 645–649, 1998.
- [2] S. C. Brailsford, C. N. Potts and B. M. Smith, “Constraint Satisfaction Problems: Algorithm and Applications,” *European Journal of Operational Research*, Vol.119, No.3, pp.557-581, 1999.
- [3] G. F. Cooper and E. Herskovits, “A Bayesian method for constructing Bayesian belief networks from databases,” in *Proc. 7th Annu. Conf. Uncertainty in Artificial Intelligence*, Los Angeles, CA, pp. 86–94, 1991.
- [4] I. D. Falco, A. D. Cioppa and E. Tarantino., “Facing classification problems with Particle Swarm Optimization,” *Applied Soft Computing*, Vol. 7, pp. 652-658, 2007.
- [5] [4] A. A. Freitas, “A genetic algorithm for generalized rule induction,” in *Advances in Soft Computing—Engineering Design and Manufacturing*, R. Roy, T. Furuhashi, and P. K. Chawdhry, Eds. New York: Springer-Verlag, pp. 340–353, 1999.
- [6] M. Garofalakis and R. Rastogi, “Scalable data mining with model constraints,” *SIGKDD Explorations*, vol. 2, no. 2, pp. 39–48, 2000.
- [7] J. Kennedy and R. C. Eberhart, “Particle Swarm Optimization” , *Proceedings of the*

- 1995 *IEEE International Conference on Neural Networks*, Vol. 4, pp. 1942-1948, 2005.
- [8] R. Kowalczyk, "Using constrain satisfaction in genetic algorithms," in *Proc. Australian New Zealand Conf. Intelligent Information Systems*, pp. 272–275, 1996.
- [9] C. J. Lin, and Hsieh, M. H., "Classification of mental task from EEG data using neural networks based on particle swarm optimization," *Neurocomputing*, Vol. 72, pp. 1121-1130, 2009.
- [10] B. Nadel, "Tree search and arc consistency in constraint satisfaction algorithms," in *Search in Artificial Intelligence*, L. Kanal and V. Kumar, Eds. New York: Springer-Verlag, pp. 287–342, 1988.
- [11] S. L. Salzgerg, "On comparing Classifiers: Pitfalls to Avoid and a Recommended Approach", *Data Mining and Knowledge Discovery*, Boston, Vol. 1, pp. 317-327, 1997.
- [12] S. C. Satapathy, J.V.R. Murthy , P.V.G.D. Prasad Reddy , B.B. Misra, P.K. Dash and G. Panda., "Particle swarm optimized multiple regression linear model for data classification," *Applied Soft Computing*, Vol. 9, pp. 470-476, 2009.
- [13] R. Srikant and R. Agrawal, "Mining sequential patterns: generalizations and performance improvements," in *Proc. 5th Int. Conf. Extending Database Tech. (EDBT'96)*, Avignon, France, pp. 3–17, Mar. 1996.
- [14] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," pp. 69-73 ,1998.
- [15] T. Sousa, et al., "Particle Swarm based Data Mining Algorithm for classification tasks," *Parallel Computing*, Vol. 30, No. 5-6, pp. 767-783, 2004.
- [16] Z. Wang, X. Sun, , and D. Zhang , "Classification Rule Mining Based on Particle Swarm Optimization," *Spring-Verlag Berlin Heidelberg*, LNAI 4062, pp. 436-441. , 2006
- [17] Z. Wang, X. Sun, and D. Zhang , "A PSO-Based Classification Rule Mining Algorithm," *Spring-Verlag Berlin Heidelberg, ICIC 2007, LNAI 4682*, pp. 377-384, 2007.