

利用蜜蜂演算法訓練遊戲角色

林正弘
東海大學
資訊工程所 研究生
G97350046@thu.edu.tw

蔡清權
東海大學
資訊工程所 教授
cttsai@thu.edu.tw

廖啟賢
東海大學
資訊工程所 講師
liaw@thu.edu.tw

柯朝輝
修平技術學院
資訊管理系 助理教授
vikey@thu.edu.tw

摘要

近幾年來最佳化演算法在遊戲人工智慧(Artificial Intelligence) 產業領域的應用與探討，已經是國內、外研究所關注的重要議題。現今電腦遊戲過度注重畫面，遊戲畫面的關注超過 AI 應用的發展速度，玩家無法體驗到與電腦所控制角色(Non-Play Character, NPC)真實的擬真互動，而遊戲內容大同小異，遊戲耐玩度不長，玩家很快就失去熱忱。更突顯最佳化演算法在遊戲 AI 之中對電腦遊戲角色進行訓練與學習策略的重要性。

現今常用的最佳化演算法對於改良遊戲AI效能部份的缺點讓不少遊戲開發者所垢病，例如基因演算法 (Genetic Algorithm, GA) 的計算量大、編碼困難，粒子群最佳化演算法 (Particle Swarm Optimization, PSO) 易收斂速度過快常陷入局部最佳解而非全域最佳解的問題。蜜蜂最佳化演算法(Bees Algorithm, BA)是一種新型態的演算法具有收斂速度快、相互溝通且不易陷入局部最佳解的優點。本文提出以蜜蜂最佳化演算法經由遊戲平台〈雷神之鎚三：競技場〉來實作對電腦遊戲角色進行訓練與比較。

藉由本研究所提出的學習方法來訓練角色行為策略，相較於其他最佳化演算法更能快速與確實找尋到更佳的行為參數組合以提升所訓練的電腦遊戲角色能力。

關鍵詞：遊戲人工智慧、蜜蜂演算法、粒子群最佳化、互動式演化

1. 簡介

電腦遊戲是軟體工業市場上一個很大的產業，目前業界所用的 AI 不僅單調且不具學習能力，這樣的 AI 很容易就會被查知並預測其行為，因此不久玩家便會覺得單調且無趣，對於遊戲壽命的延展性，並沒有任何幫助。我們將針對於戲的 AI 進行研究，對於玩家來說，最好的 AI 就是最真實擬真的，也就是會多樣思考，具有學習能力的學習與適應系統，應是目前遊戲角色設計的方向[4-6]。

現今遊戲即時戰略性質的人工智慧演算法例如粒子群最佳化演算法(PSO)與基因演算法(GA)有著快速且穩定的特色。粒子群演算法的特點是：概念簡單、實作容易、收斂速度快、適合非線性等多變數複雜問題、運算成本低、沒有太多的參數需要調整，因此預計在許多的情況下所有的粒子可能更快的收斂於最佳解，然而 PSO 易收斂速度過快常陷入局部最佳解而非全域最佳解的缺點，這是令開發者頭痛的一點，現今很多 PSO 相關論文也針對此點進行改進。

常見的基因演算法的特性是在整體搜尋空間大時，較易求得全域最佳解也較精準、可以處理的資料型範圍極大。然而相

較之下較為人垢病的就是計算上大、運算成本高、收斂速度較慢、有些問題有編碼上的困難、而且它的解不保證最佳化。

針對 AI 作訓練改良，目的是藉由全方位的訓練，得到一個會多樣思考，具有自己情緒與學習能力的非玩家遊戲角色。本研究利用蜜蜂演算法作為遊戲 AI 訓練策略，BA 演算法最佳化需考慮到 BA 運算中如何使用、設定相關的運算條件、最佳化參數變化控制，針對不同的求解問題，讓遊戲 AI 中角色行為找出合適的最佳解。

2. 相關研究

目前已有一些遊戲公司願意將其遊戲的原始碼開放給玩家更改，以做為研究的用途或是豐富遊戲的人工智慧，所以有些學者開始利用商業遊戲作為其研究的平台。John Laird 發展了一套 Soar 的 AI 引擎，Soar 是一個可藉由修改少數的函式呼叫後，即可在不同類型中使用的 AI 引擎，而之後 Laird[7]並用它來設計 Quake II 的 BOT，使得遊戲裡的 BOT 可以預測玩家的動作，模擬人類高手的策略模式。另外有些學者則是選擇在自己開發的遊戲或是以模擬的方法做為研究的平台。

Poropudas[8]提出了一種新的遊戲理論，應用統計技術於估計遊戲數據產生的模擬狀態上，對於飛機模擬空戰遊戲與模擬最優化的方法。Chambers[9]網絡遊戲是一個迅速發展的網路應用，為了運行一個成功的線上遊戲，遊戲公司和遊戲基礎設施供應商必須妥善管理的工作量和遊戲內容，以便他們能夠最大限度地提高玩家滿意度，同時盡量減少自己的成本。Weng[10]經典的 Pac-Man 遊戲，有明確的布林邏輯

遊戲規則和簡單的遊戲場景，藉由改變遊戲規則(Boolean logic)來改變遊戲比賽情況。

粒子群最佳化演算法(PSO)[11]與基因演算法(GA)[12]可用於遊戲角色的演化 對此藉由相關研究先行評估其優劣點[1-3]，以適應函數值比較單人角色 AI 演化行為為參數，適應函數值越高代表角色越強大，以 PSO 為例其適應函數值方面會比 GA 部份收斂速度來快很多，每一代適應函數值的大小分佈均勻，計算量也較小，因為 GA 在基因編碼上的複雜困難使得計算量大增，自然收斂速度較慢、運算成本高，然而並不是 GA 就比不上 PSO，經長時間場數的疊代演化適應函數值是有機會比 PSO 來的優秀，驗證 GA 在搜尋範圍大時較容易取得最佳解、並且可以處理極大範圍資料的優點。

對遊戲開發者而言，PSO 似乎是不錯的選擇，但因為粒子的移動受到權重值所影響，若權重值太高會使全體最佳值收斂過快而陷入局部最佳解而非全域最佳解的情形，這也就是令遊戲開發者困擾的一點，所以在此提出新型態的演算法進而比較效能與改善。

3. 蜜蜂最佳化演算法

3.1 蜜蜂演算法

蜜蜂演算法主要是摹仿蜜蜂行為群體智慧的一種演算法[13]，是一種新型態的智能最佳化演算法。本研究採用蜜蜂演算法來針對遊戲 AI 角色行為的訓練演化，進而求取最佳的角色行為參數，創造出不斷從遊戲中錯誤經驗學習與改進的能力，多樣

性的電腦角色 [14-18]。

然而蜜蜂演算法跟一般既定的最佳化演算法不太一樣，沒有所謂標準的數學公式與未固定的參數配置，主要經由演算法搜尋過程加上參數間的配置藉由各蜜蜂區域間與區域間的搜尋，圖 1 顯示搜尋的過程與參數配置。

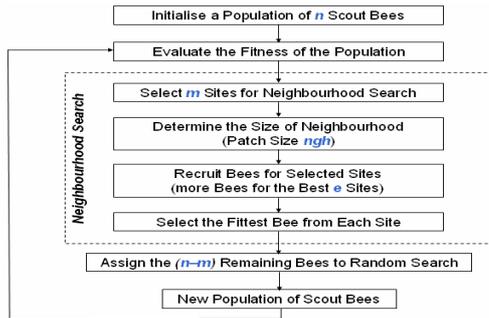


圖 1 遊戲平台所演練 BOT 的搜尋的流程圖

由圖 1 可知蜜蜂演算法搜尋的流程藉由所訂定的範圍不斷的縮小搜尋範圍進而找尋最佳解，再者如下圖 2 所示，以一個 X,Y 的二維空間尋找最佳解為例偵察蜜蜂即為我們演練的 BOT 在全域中搜索，圖中可了解到各參數之間的 e 與 m 配置先找出較佳的範圍 m 進而在對於最佳的範圍 e 慢慢縮小搜尋最後求取最佳解與藉由演算法的搜尋過程流程圖。

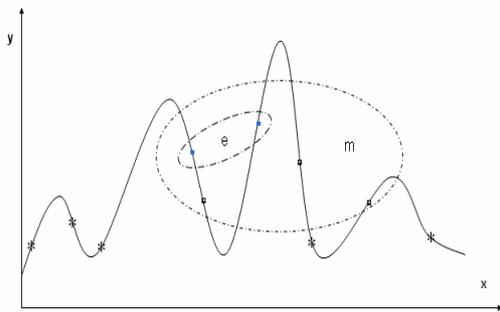


圖 2 搜尋演化過程

藉由圖 1 搜尋過程流程圖所表示，參數 n 表示偵察蜜蜂的數量，在 n 個偵察蜜

蜂搜尋過區域中選擇 m 個較好搜尋區域。在 m 個較好偵察蜜蜂搜尋區域之中選擇最好的 e 個搜尋過區域。 $nep(n2)$ ：派遣蜜蜂到最好 e 搜尋區域之中的數量，稱為 nep 。 $nsp(n1)$ ：派遣蜜蜂到 $m-e$ 的剩餘搜尋區域的數量，稱為 nsp 。 ng ：初始的搜尋半徑大小。

3.2 適應函數

適應函數對於遊戲角色的行為表現有很大的關聯性，而適應函數依據我們所演練的 BOT 經由每一代與 5 個電腦 NPC 進行對戰，遊戲模式為【死鬥模式】以單人率先擊殺到 15 人次當作一回合也就是我們所操作的 BOT 當代的結果產生一個適應值 (fitness value)，此適應值用來評估我們所演練的演算法產生的策略是否得宜，進而影響當代所演練的 BOT，越高的值代表 BOT 表現越好，以下所使用的公式作為我們多目標蜜蜂演算法的適應函數：

$$FitnessValue = 2.0 \times kills(num) - 1.0 \times deaths(num) \quad (1)$$

其中 $kills$ 表示我們所操作 BOT 的殺敵次數， $death$ 表示 BOT 被擊殺的次數，以公式(1)所示，針對 FPS 射擊遊戲所操作的 BOT 在戰場上殺敵數越高越強悍，自身死亡率越低，存活的時間越久，為我們的目標進而演練出強大的 BOT，適應函數會直接影響 BOT 的行為表現，越高的值表示演算法的演練找出每代 BOT 的更佳行為參數的策略組合，藉此不斷的演化在每個疊代找尋最佳的策略組合，直到收斂為止找到最佳的適應值，

以達到最佳的策略演練 BOT。

3.3 演算法效能評估

經由上述的介紹可知蜜蜂演算法、粒子群最佳化演算法與基因演算法的特色。粒子群演算法粒子之間本身雖然會相互溝通訊息但粒子整體間的移動是靠全區最佳解來做為前往的方向，再者 PSO 易收斂速度過快常陷入局部最佳解而非全域最佳解，因為粒子的移動深受權重值所影響，如果權重值太高將會造成全體最佳值收斂過快而陷入局部最佳解的情形。

本文採用蜜蜂演算法概念去解決最佳解的問題，蜜蜂演算法主要的特點是：蜜蜂群體之間相互溝通訊息，每隻個體蜜蜂均獨立運算以外還會跟其他蜜蜂分享自己運算過後的資訊，不同於 PSO 經由全區最佳解粒子來做為方向前往找尋，易陷入局部最佳解，這方面蜜蜂是個別各區搜尋溝通有效改善此缺點，本身演算法是採用優生策略，改善權重值對演算法的影響。這樣有著收斂快速、穩定且沒局部解問題的優點，缺點方面由於一開始所訂定的參數是非固定的，較不易一開始就找尋到較合適的參數，但經由經驗法則研究還是可以訂定出初始的有效參數。

3.4 BOT 演化流程

使用蜜蜂演算法於遊戲平台上演化我們 BOT 的過程如圖 4，偵察蜜蜂即為我們演練的 BOT，一隻蜜蜂代表一個 BOT 行為，而一個行為所提到有 9 種行為模式如圖 3，由對戰結果算出適應函適值。在圖

中的搜尋過程中，於 n 個行為參數組合方法中，選擇 m 個較好參數組合，進而再從這 m 個中選出 e 個更優良參數組合， e 與 $(m-e)$ 各自加入 $n1$ 、 $n2$ 的隨機參數值，目的是希望找尋到更好的行為參數組合，讓 BOT 不斷的嘗試不同的參數組合，並根據適應函式評斷每個 BOT 的行為表現，而表現較差 BOT 的隊員行為參數，會向表現較好 BOT 的參數做修正，以此求得更佳適應值解與更強大的 BOT，經由不斷疊代找尋較佳的角色行為參數組合。

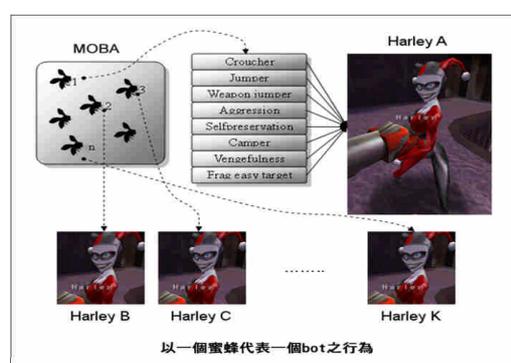


圖 3 一個蜜蜂代表一個 BOT 之行為

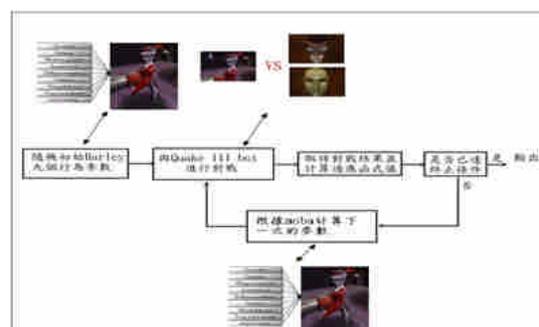


圖 4 BOT 的搜尋的流程圖

4. 實驗結果

經由蜜蜂演算法與粒子群最佳化演算法所訓練的 BOT 兩者在 Quake III 平台分別與平台本身的 BOT 進行對戰，並針對所求得適應值高低來搜尋，並經由實作實驗來比較其結果。關於蜜蜂演算法初始的參數

是可任意制定的，參數 n 在此我們設為8，參數 m 設為4、參數 e 設為2、參數 nep ($n2$)設為3、參數 nsp ($n1$)設為2、參數 ngh 設為0.25。首先以兩演算法分別訓練20場取得最佳的適應值，找到最佳的策略組合，再分別進行實戰20場相比較，結果顯示於圖5，綠色線為當疊代數的平均適應值，紅色線為此疊代目前的最佳值。

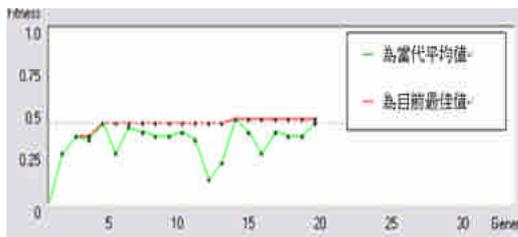


圖 5 PSO訓練20場的適應函數值

藉由 PSO 演算法訓練我們的 BOT (Harley) 在找到局部最佳解後容易陷入局部解，不易跳脫進而找尋到全域最佳解，此訓練得到最佳值為0.48之行為參數。圖 6 為蜜蜂演算法訓練20場的適應函數值，由蜜蜂演算法不只取得更佳全域最佳解為0.58之行為參數，更快收斂更有效避免陷入局部最佳解。

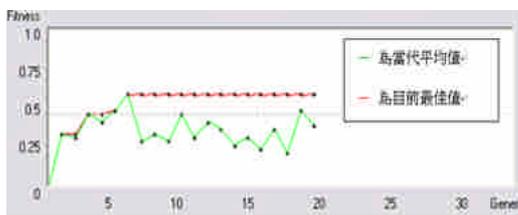
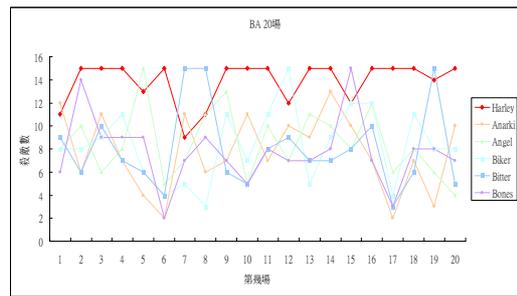
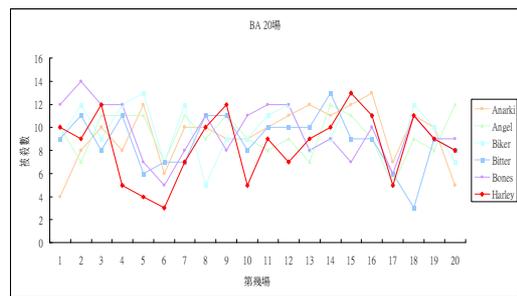


圖 6 BA訓練20場的適應函數值

圖 7(a)與(b)為 BOT (Harley)(紅色線段為我們BOT)以BA演算法分別對其他5個BOT實戰20場的對戰結果，一回合結束設定為一方擊殺數達到15人為條件。



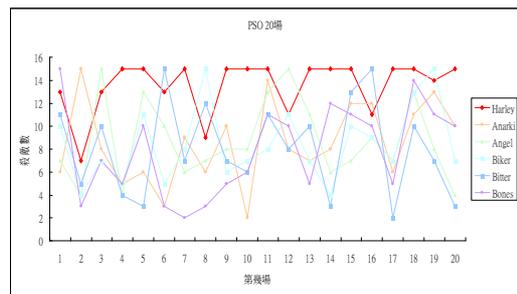
(a)



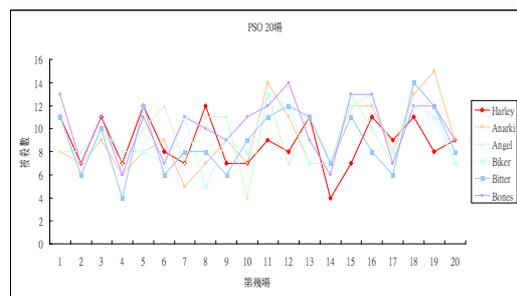
(b)

圖 7 BA實際對戰20場的殺敵數成績(a)與被殺數成績(b)

圖 8(a)與(b)為 BOT (Harley)(紅色線段)以BA與PSO分別對其他5個BOT實戰20場的對戰結果。



(a)



(b)

圖 8 PSO實際對戰20場的(a)殺敵數結果與(b)被殺數結果

BA在20場中有14場第一，PSO在20場中有12場第一，BA比PSO對戰結果好。由表 1可知不管蜜蜂演算法在殺敵率(Kill rate)與被擊殺率(Death rate)都比PSO來的好，BA演算法在各方面均比PSO演算法來的優異，AI部分更能提升與演練出更強更聰明的BOT。

表 1 BA與PSO各個BOT之間殺敵數(率)與被擊殺數(率)

BOT	[BA]	Harley	Anarki	Angel	Biker	Bitter	Bones
Number of kills	277	155	170	167	161	155	
Kill rate %	25.5	14.2	15.6	15.3	14.8	14.2	
Number of death	169	188	187	195	176	193	
Death rate %	15.2	16.9	16.8	17.5	15.8	17.4	
BOT	[PSO]	Harley	Anarki	Angel	Biker	Bitter	Bones
Number of kills	271	171	175	169	162	158	
Kill rate %	24.5	15.4	15.8	15.2	14.6	14.2	
Number of death	176	183	189	185	180	203	
Death rate %	15.7	16.4	16.9	16.5	16.1	18.1	

5. 結論

現今最佳化搜尋演算法眾多，本研究對於時下較廣泛應用的演算法探討與比較其優缺點，並提出新形態的蜜蜂演算法來改善解決PSO最佳化演算法易陷入局部最佳解的缺點，實驗結果顯示不僅收斂快速且沒局部解的困擾，進而求取全域最佳解。藉由此論文研究，讓遊戲人工智慧開發工程師能夠更方便的創造智慧、多樣、擬真與具有學習能力的角色行為。

本文所提出的方法可以幫助遊戲開發者更快速、更有效率的開發遊戲，節省調整參數所需的時間。將學術上蜜蜂演算法

實際應用於遊戲角色AI與行為設計，並展現此演算法在遊戲設計領域具有優越的使用價值，可實際應用於商業性之遊戲設計。

參考文獻

- [1] 陳柏仲、蔡清樞、廖啟賢、林灶生，以粒子群最佳化演算法為基礎之第一人稱射擊遊戲角色設計，智慧生活科技研討會，2006。
- [2] 郝國平、蔡清樞、廖啟賢，訓練電腦遊戲團隊的策略，2010 資訊與管理應用研討會，March 2010。
- [3] 黃奐禎、蔡清樞、廖啟賢、廖元勳，利用粒子群演算法在電腦遊戲訓練團隊策略，銘傳大學 2007 國際學術研討會，March. 2007。
- [4] C. Browne, F. Maire, Member, IEEE, “Evolutionary Game Design,” IEEE transactions on computaional intelligence and AI in games, 2010, pp.1 – 16.
- [5] M. Preuss, Member, IEEE, Nicola Beume, Holger Danielsiek, Tobias Hein, Boris Naujoks, Nico Piatkowski, Raphael Stüer, Andreas Thom, and Simon Wessing, “Towards Intelligent Team Composition and Maneuvering in Real-Time Strategy Games,” IEEE transactions on computaional intelligence and AI in games, 2010, pp.82 – 98.
- [6] H. Wang, Y. Gao, and X. Chen, “RL-DOT: A Reinforcement Learning NPC Team for Playing Domination Games,” IEEE transactions on

- computational intelligence and AI in games, 2010, pp.17 – 26.
- [7] J. E Laird, “It knows what you’re going to do: adding anticipation to a quakeBot,” Proceedings of the Fifth International Conference on Autonomous Agents, ACM Press ,Montreal,Canada, 2001, pp 385-392.
- [8] J. Poropudas and K. Virtanen, "Game-Theoretic Validation and Analysis of Air Combat Simulation Models", IEEE transactions on systems, Man, And cybernetics —part A: systems and humans, 2010 , pp.1057 – 1070.
- [9] C. Chambers, W.C. Feng, S. Sahu, D. Saha, D. Brandt, "Characterizing Online Games", IEEE/ACM transactions on networking, 2010, pp.899 – 910.
- [10] J.F. Weng, S.S. Tseng, T.J. Lee, "Teaching Boolean Logic through Game Rule Tuning", IEEE transactions on learning technologies, 2010, pp.319 – 328.
- [11] R. C. Eberhart and J. Kennedy, “A New Optimizer Using Particle Swarm Theory,” *proc. Sixth International Symposium on Micro Machine and Human Science*, Nagoya,1995, pp. 39-43.
- [12] T. E. Revello, “Generating War Game Strategies Using A Genetic Algorithm,” Proc. IEEE Evolutionary Computation '02, 2002, pp. 1086-1091.
- [13] D. T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim and M. Zaidi, “The Bees Algorithm,” Technical Note, Manufacturing Engineering Centre, Cardiff University, UK, 2005.
- [14] A. Jevtic, P. Gazi, D. Andina, M. Jamshidi, “Building a swarm of robotic bees,” World Automation Congress (WAC), 2010 , pp.1 – 6,2010.
- [15] C. Sumpavakup, I. Srikun, S. Chusanapiputt , “ A solution to the Optimal Power Flow using Artificial Bee Colony algorithm ,”Power System Technology (POWERCON), 2010 International Conference on, 2010,pp. 1 – 5.
- [16] N. Othman, I. Musirin, M.A. Rahim, Z.Othman,“Bees algorithm technique for loss minimization in power transmission network using Static Var Compensator ,” Power Engineering and Optimization Conference (PEOCO), 2010 4th International 2010 · 2010 , pp. 164 – 169.
- [17] H. Li, K. Liu, N. Li, “Improved bees algorithm for the large-scale layout optimization without performance constraints ,” Bio-Inspired Computing: Theories and Applications (BIC-TA), 2010 IEEE Fifth International Conference on, 2010, pp. 459 – 463.
- [18] D. T. Pham ,M. Sholedolu, “Using a Hybrid PSO-Bees Algorithm to train Neural Networks for Wood Defect Classification,” in 4th International Virtual Conference on Intelligent Production Machines and Systems (IPROMS 2008): Whittles, Dunbeath, Scotland, 2008.