

私立東海大學資訊工程學系
專題實驗聯合評審報告書

日期:99年12月18日

[雷神之鎚 3 - 人工智慧之優化]

THE OPTIMIZATION OF ARTIFICIAL INTELLIGENCE ON QUAKE3

指導教授 蔡清欉教授

專題學生 963915 邱為義

963832 廖啟瑞

963938 黃歆雅

摘要

玩遊戲已經成為現代人們在不景氣之下重要的一項休閒娛樂活動，電腦遊戲擁有高度的互動性以及整合各式媒體的能力。特別是在網路環境普及後，多人連線的玩家互動式線上遊戲更是引發了一股新的熱潮。現今角色行為設計普遍存在有角色行為規則訂定、調整耗時、行為不具學習性的缺點，比起以往面對著單調死板的電腦 AI，現在的電腦玩家更渴望的是擁有豐富多樣化的非玩家控制的角色（non-player character）。

本文是針對遊戲角色之行為設計來做探討，希望能針對不同場合、不同的 NPC，有著不一樣的行為模式使其能達到殺人與被殺之間的平衡，並且利用粒子群演算法最佳化技術調整影響角色行為參數，讓我們所設計的人物具有因應不同狀況下的能力。訓練的過程中不需要大量的訓練資料以及費時的運算，可以更適合遊戲環境的應用，並且可以輔助遊戲的人工智慧設計師調整行為策略時，策略參數的組合並且節省測試參數組合的時間，增加遊戲開發的效率。

最後我們將所提出的演算法套入雷神之鎚 3 (Quake III)遊戲中的死鬥競技場模式分別與原始的人工智慧做比較，來證明我們的方法在遊戲的結果中表現的確比較出色。

目錄

摘要.....	i
目錄.....	ii
第一章緒論	1
1.1 動機.....	1
1.2 目標.....	1
1.3 大綱.....	2
第二章相關研究.....	3
2.1 遊戲之人工智慧.....	3
2.2 非玩家控制角色.....	4
第三章粒子群演算法之探討.....	4
3.1 粒子群演算法.....	5
3.2 適應函式與參數控制.....	7
第四章電腦射擊遊戲之策略.....	9
4.1 雷神之鎚 3 之介紹.....	9
4.1.1 故事背景.....	9
4.1.2 遊戲簡介.....	9
4.2 遊戲角色之行為.....	10
4.3 競技場模式.....	12

4.4	QUAKE III 之 BOT 行為最佳化.....	14
4.4.1	QUAKE III 之 BOT 設計.....	14
4.4.2	BOT 行為之調整.....	15
4.4.3	最佳化相關參數與適應函數之設定.....	16
4.4.4	實驗設計與過程.....	18
	第五章結論	21
	第六章參考文獻	22

敘論

1.1 動機

我們觀察目前的遊戲類型後發現，目前市場上具有團隊合作的電腦遊戲通常受到很大的歡迎，例如：「雷神之鎚 3」裡玩家組對搶奪對手的旗幟或防守己方堡壘；以及「百變恰吉」中玩家組成 2 人、3 人、5 人隊伍去競技場中挑戰國內玩家所組成的隊伍，雖然遊戲中角色的技能都固定，但我們必須考量到角色內行為的參數與玩家的素質來應付不同的狀況。所以人類玩家在團隊遊戲的合作上會依據當時的狀況取用個人最佳的策略，若電腦角色所構成的行為毫無挑戰性可言，則遊戲的趣味以及互動性也會大打折扣，進而縮短遊戲的生命，因此我們提出一種方法，將電腦行為訓練為有效率的角色，讓角色中的行為兼具備高度殺人數及少量被殺數的功用，在不同的狀態下都能採取對應的最佳策略，將會是我們要探討的主要課題。

1.2 目標

現今的遊戲普遍存在以下缺點：

- 相同的遊戲角色均僅具有相同的動作
- 遊戲角色的行為不具有學習性

“相同的遊戲角色均僅具有相同的動作”指的是，隨著遊戲的進行，玩家會遇到許多各式各樣的非玩家控制角色(NPC)，尤其是阻礙玩家進行遊戲的敵人，依據遊戲腳本的設定，同類型的敵人通常會在一特定的區域重複出現，甚至是成群結隊的出現，這相較於現實生活中的情況是有些不合理的與不真實的。而“遊戲角色的行為不具有學習性”，也就是說遊戲角色無法藉由與玩家或是與虛擬環境的互動而改變既有動作，這是與由玩家所控制的角色之最大不同點，也是一般玩家更喜好與真實玩家互動的主要原因之一。因此我們希望利用演算法最佳化技術創造電腦遊戲中的角色，包括了有：

- 讓遊戲動畫設計師能夠更快速的創造出多樣化的角色動作
- 讓遊戲人工智慧設計師能夠更方便的創造智慧、擬真與具有學習能力的角色行為
- 讓遊戲玩家藉由參與角色最佳化的過程，感受到角色行為的不可預測性。

1.3 大綱

本文以粒子群最佳化基礎，訓練角色行為的策略並同時讓團隊兼具殺人及被殺的功能。而我們則是以雷神之鎚 3 死鬥競技場 (Quake III :

Team Arena) 作為實驗平台，使用我們的方法來訓練出一個人物與電腦內建人物在死鬥競技場中對戰並且比較其結果。

相關研究

2.1 遊戲之人工智慧

在網路環境普及後，玩家互動式的線上多人遊戲(Multy players on-line game)逐漸形成一個大宗，比起單人遊戲中的電腦 AI 角色，在連線遊戲中所面對的，是真正的玩家，可以進行具有人性化的行為及互動，不但讓遊戲進行的過程更加有趣，還可成為玩家之間交流的平台。然而儘管網路遊戲的出現，非玩家控制的角色 (none-player character, NPC)仍有其持續存在的需求與不可替代性。NPC 電腦遊戲角色設計指的是對其角色進行造型 (model)、動作(motion)與行為 (behavior) 的設計，而 NPC 就是由電腦人工智慧(AI)所控制的角色。

現今大部份的遊戲，因為基於簡單、容易實作，仍是使用規則為基礎的傳統設計方法，但以規則為基礎的方法最大的缺點就是，一但讓玩家找到其中的弱點，就無法阻止玩家利用這個弱點不斷的擊敗電腦角色，如果可以讓電腦角色具有從錯誤中學習的能力，並且嘗試著解決

這個問題，所以將具有學習能力的學習與適應系統以及演算法導入遊戲人工智慧，將是未來遊戲角色設計的方向。

2.2 非玩家控制角色

提到電腦遊戲就不得不聯想到「非玩家控制角色」(Non-Player Character—NPC)，NPC 通常扮演七種角色，分別是：一、玩家角色協助者。二、玩家角色敵對者。三、玩家角色夥伴。四、玩家角色暴行的受害者（不論陣營）。五、引導玩家角色發生事件的關鍵虛擬角色。六、提供事件背景的資訊。七提供玩家任務並使玩家獲得獎勵。角色經由遊戲設計師賦予適當的人工智慧，讓這些 NPC 能夠表現出不同的特質，並且透過特有的情緒、脾氣、音樂、台詞，讓玩家更能融入遊戲的劇情當中，所以一款遊戲要令玩家回味無窮，除了豐富的劇情、感官的刺激、華麗特效外，靈活的人工智慧更能讓玩家每次進入遊戲都有不同的體驗。

粒子群演算法之探討

3.1 粒子群演算法(PSO)

在PSO 中，一個最佳化問題的解就像是一隻在空間中飛行的鳥一

樣，他們稱作“粒子(particle)”，在空間中移動的所有粒子都有一個由適應函式所決定的適應值，另外每個粒子還有一個速度來決定他們移動的方向與距離，一群粒子靠著追隨個人的成功經驗與目前最佳粒子的腳步在解空間中飛行。在初始階段，PSO 會隨機產生一組粒子（位置與速度），然後透過一次次的疊代找尋最佳解，在每次的疊代過程中每個粒子利用二個最佳值”來更新自己的速度，一個是粒子本身所找到的最佳解，稱為”個體最佳值(pbest)””，另一個最佳值是由 PSO 所記錄與更新，為全體粒子所找到的最佳解，稱為”全體最佳值(gbest)””，若一個粒子所能受影響的範圍是區域性的，則這裡的最佳解就要改稱為”區域最佳值(lbest)””。當粒子找到這二個最佳值之後，便利用下面這個式子來更新粒子的速度：

$$v_i^{k+1} = wv_i^k + c_1rand_1 \times (pbest_i - s_i^k) + c_2rand_2 \times (gbest - s_i^k)$$

在這個式子中， V_i^k 代表粒子*i* 在第*k* 次疊代中的速度；*w* 是權重函式；*c1* 與*c2* 是學習因子，通常，*c1=c2*；*rand* 是一個介於0~1 之間的隨機值； S_i^k 代表粒子*i* 在第*k* 次疊代中的位置；*pbest_i* 是粒子*i* 的最佳值；*gbest* 是全體的最佳值。

以下是上述式子所用到的權重函式：

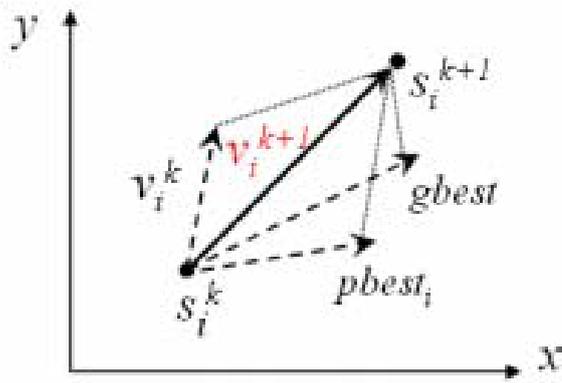
$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \times iter$$

在這個式子中， w_{\max} 代表初始權重， w_{\min} 代表最終權重， $iter_{\max}$ 是最大的疊代數目， $iter$ 則代表目前的疊代次數， w 的值將會隨疊代次數的增加越來越小，這表示PSO 希望在最佳化過程中的開始傾向讓粒子更積極地搜尋更佳的問題解，也就是“探索(exploration)”，而隨著疊代次數的增加，粒子則會傾向滿足現有的最佳解，也就是“利用(exploitation)”。

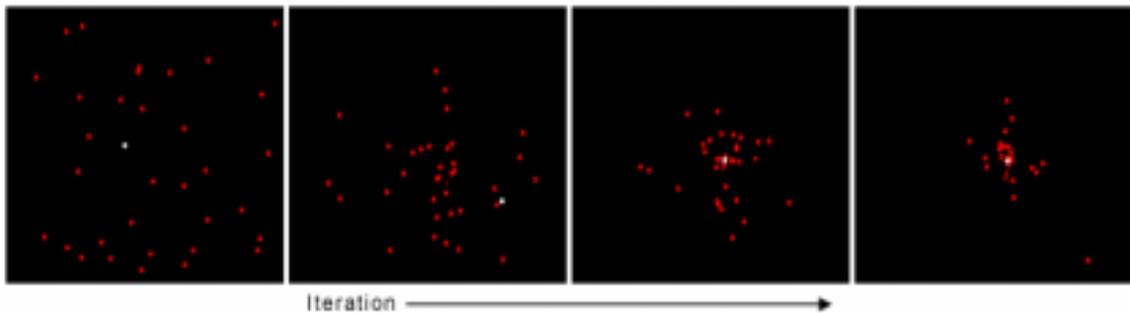
每個粒子在第 $k+1$ 次疊代中的位置（解空間中的搜尋點），可以簡單的利用下面的式子來更新：

$$S_i^{k+1} = S_i^k + V_i^{k+1}$$

以下為粒子在解空間中移動之示意圖：



以下則為整個粒子群搜尋最佳解過程之示意圖，粒子群透過不停的疊代程序逐漸收斂並找到問題的近似最佳解，此時大部分的粒子會處在一個相似的狀態，也就是提出類似的問題解。



3.2 適應函式與參數控制

應用 PSO 解決最佳化問題的過程中有二個重要的步驟：問題解的編碼與適應函式的制定，編碼的維度與長度將影響求解的速度而適應函式將直接影響粒子搜尋的方向。PSO 的一個優勢就是採用實數編碼，不需要像基因演算法一樣利用二進制編碼（或者採用針對實數的

基因操作)，舉個例子：如果我們希望求得 $f(x)=x_1^2+x_2^2+x_3^2$ 這個式子的最佳解，則可以直接將解編碼為 (x_1, x_2, x_3) ，而適應函式就是 $f(x)$ ，之後就可以利用標準的PSO程序直到達到終止條件，終止條件可以為到達最大的疊代數或是達到最小的錯誤。在 PSO 中一般並沒有什麼參數需要調整，一些關於參數上的經驗設定：

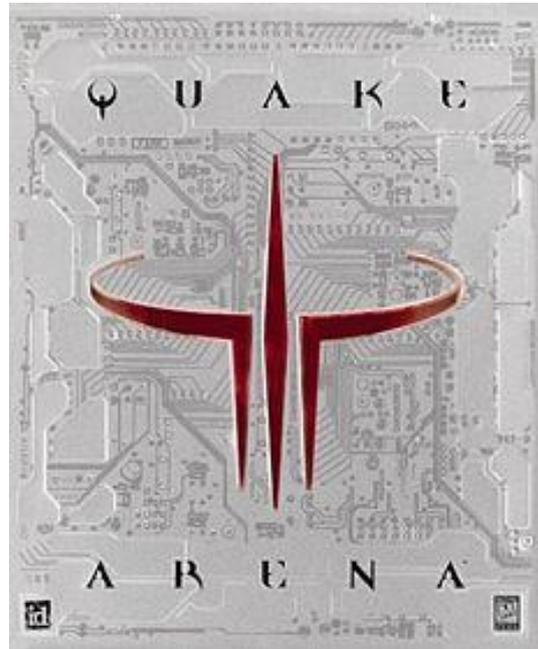
- 粒子數：一般取20-40，其實大部份的問題10個粒子已經足夠可以取得不錯的結果，不過關於比較難的問題或者特定類別的問題粒子數可以取到100或200。
- 粒子的最大速度：決定粒子在一次的疊代中最大的移動距離，通常設為粒子的範圍寬度。
- 學習因子： c_1 和 c_2 通常等於2，不過在其它的文獻中也有其它的設定，但一般 c_1 等於 c_2 且範圍在0~4之間。
- 全體最佳值或區域最佳值：前者速度較快，不過有時會陷入局部最佳解的情形。後者速度較慢，不過不容易陷入局部最佳解，在實際的應用中可以先用全體最佳值找到大致上的結果，再利用區域最佳值進行搜索。

電腦射擊遊戲之策略

4.1 雷神之鎚3(Quake III) 介紹

4.1.1 故事背景

玩家是一名士兵，被政府派往傳送門去阻止名為"Quake"的敵人。這個入口使用政府的新技術傳送各種可怕的魔鬼，傳送門可以即時傳送各種物品。一旦透過這扇傳送門，



你就必須同成群的魔鬼作戰並阻止它們。雷神之鎚中的無名英雄在雷神之鎚III競技場中再一次出現，此次是作為可選的角色之一，名為"Ranger"。

4.1.2 遊戲簡介

id Software 的雷神之鎚3(Quake)是我們所選擇的開放程式碼與角色展現行為的平台，id Software 所創造的動作射擊遊戲與遊戲引擎一向是業界的標竿，遊戲引擎開放授權與鼓勵玩家修改遊戲程式(MOD)的概念也是id Software 所首先提出。雷神之鎚3 是一款第一人稱動作

射擊遊戲，遊戲中玩家在不同的場地，使用不同的武器、道具與對戰策略擊倒AI 對手(稱作bot)或網路上的玩家以獲得射擊的樂趣。雷神之鎚3 程式中的遊戲引擎(繪圖與網路之核心程式碼) 是屬於需要授權才能取得的部份，而遊戲邏輯(與遊戲邏輯相關的程式碼) 則是開放讓玩家修改，並且鼓勵玩家修改，這樣的作法除了增加遊戲玩法的多元性之外，更增加了遊戲商品的壽命。

在Quake III裡利用人類高手的行為模式做為訓練資料，讓電腦學習高手的行為模式遊戲中玩家在不同場地，使用不同武器、道具與策略擊倒NPC 對手(稱作bot) 或網路玩家以獲得射擊的樂趣。bot 一詞源自robot，通常指的是第一人稱動作射擊遊戲中由電腦所控制的AI NPC 角色。而我們所要設計的即是Quake 的bot。

4.2 遊戲角色之行為

在遊戲模式中，個人行為扮演了十分重要的角色，而本身的決策也就是「行為的策略」，更是決定自己是否可以贏得比賽的關鍵。

遊戲提供了玩家九種不同的行為，可以分別代表隊員不同的特徵。

- 蹲伏 (Crouch)：隊員會有著類似現實生活中蹲伏射擊對手的行為。
- 跳躍 (Jump)：隊員會利用地形來進行跳躍以便攻擊或是躲避對

手。

- **武器切換 (Weapon Jump)**：當此種武器彈藥消耗完畢，並且在此時遇到對手時馬上切換其他種類武器來迎擊對方。
- **警覺 (Alerthness)**：隊員會在指定地點來回巡邏，避免遇到敵方的NPC。
- **復仇心 (Vengeful)**：被同一目標擊殺過多時，下一次會以此目標為第一擊殺為優先考量。
- **侵略 (Aggression)**：當遇到一群對手在相互對打時會積極的進攻對方。
- **自我保護 (Self preservation)**：當本身條件屬於不利的情況，例如：裝甲值過低、血量太少…時，盡量避免與對方戰鬥或是衡量是否直接放棄目前戰鬥直接等待復活，以便快速進行下一次的戰鬥。
- **躲藏 (Capm)**：配合其他行為再搭配自身的地形、條件，去評估是否躲藏起來偷襲對手。
- **擊火目標 (Frag easy target)**：若看到某個或多數個NPC同時攻擊同一目標，此時會判斷是否參予一同對付弱勢的一方。



我們針對這九種行為以及死鬥競技場模式去進行歸類，並且我們的 PSO演算法套入，其中我們以「殺人」以及「死亡」來代表我們演算法中的兩個目標，我們的bot會在遊戲的進行中根據當時自身的狀況採取不同的策略，因此，我們研究目標便是讓我們的隊員不論是在處於有利或是不利的狀態下，都能夠達到盡量少死亡多殺人這兩方兼具特性，讓我們獲勝；或是比較不同操作點對於競賽有何不同處的特性，例如：當我們選擇的操作點為死亡數的值較高而殺人數的值較少時與電腦NPC比賽有何成效，或是死亡數的值較少而殺人數的值較高時，又有怎樣的結果。

4.3 競技場模式

Quake III本身提供的遊戲模式有四種，分別是：「Free for all」、「Team Deathmatch」、「Tournament (Duel)」、「Capture the flag」。

下列為這四種模式的主要說明：

- 「Free for all」：競技場模式，同時給多個玩家同時進行遊戲，已殺死其他玩家來獲取分數，殺死一個其他玩家就得一分，自殺或者從高處掉下等並非被對方玩家殺死而死亡的情況則扣一分。
- 「Team Deathmatch」：簡稱「TDM」，將遊戲中玩家分為紅藍兩隊的遊戲模式。玩家以殺死對方隊伍中玩家為目標，由於TDM模式的玩家數目相對較多，因此在此模式下地圖中的武器與盔甲等資源便得尤其重要，所以玩家在射殺對方玩家的同時還需要與本隊的玩家進行配合。
- 「Tournament (Duel)」：一對一的單挑模式，在此模式中考量著玩家與對手間的各项素質，例如：武器的選擇、盔甲值的多寡、地形的認知…等
- 「Capture the flag」：簡稱「CTF」，將隊伍分成紅藍兩邊在一個對稱的地圖中競賽，競賽的目的是將對方的旗子帶回來並且碰觸未被移動過的我方棋子，我對就得一分，此模式一般會由時間的設定來結束，避免雙方勢均力敵而難以得分。

而在這四種模式中，我們選擇「Free for all」，競技場模式來作為我們實驗的平台。此外，我們將殺人數設定在15人，也就是說，只要有一個NPC 在殺人數上達到15人則就停止這回合的戰鬥。遊戲每「回

合」的過程，會記錄著我們的隊員與其他NPC 的對戰紀錄，因此讓我方的隊員存活的時間越長越好，反之，我方被對手擊殺的次數越少越好，這樣才能讓我方時常保持在有利的狀態，進而快速結束此回合的戰鬥。

4.4 QUAKE III之BOT 行為最佳化

4.4.1 QUAKE III 之BOT設計

id Software 的雷神之鎚3(Quake III: Arena)是我們所選擇的開放程式碼與角色展現行為的平台，遊戲中玩家在不同的場地，使用不同的武器、道具與對戰策略擊倒AI 對手(bot) 或網路上的玩家以獲得射擊的樂趣。

而bot AI 運作的方式為，由bot AI 開放程式碼(bot AI open Source)與bot AI 函式庫(bot AI library)經由編譯與連結形成動態連結函式庫(DLL)並由遊戲主程式於遊戲期間呼叫，而外部bot 檔案(external bot files)則會在每個bot 加入遊戲之前由bot AI 函式庫中的程式所負責載入。

bot AI 函式庫所負責的是較低階的AI 工作，包括外部bot 檔案的載入、bot 的感知與動作展現以及部份的行為控制。bot AI 開放程式碼則負責大部份的bot 行為控制、長期與短期目標的決定、行為規則的

定訂等，使用到了模糊狀態機的概念，在長期與短期目標的決定上，模糊狀態機會根據遊戲狀態以決定目標與行為是否符合執行的條件，並且根據外部bot 檔案中關於bot 行為特色的權重與一個隨機值來決定是否訂出目標與做出行動。在武器與物品的選擇與取用上，模糊狀態機會根據狀態與經驗法則動態決定一組權重，並且與外部bot 檔案中關於bot 行為特色的權重決定來決定角色的行為。

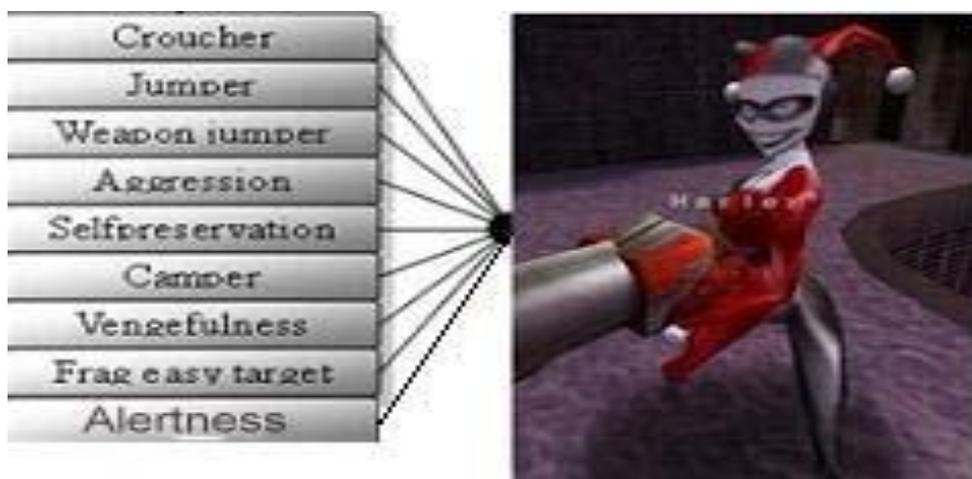
4.4.2 BOT行為之調整

外部 bot 以四個檔案為一組描述一個bot 的各項能力、行為偏好之權重、武器與物品取用之權重、bot 之對話能力等，這樣的設計最主要是方便AI 設計人員調整與測試角色的行為，一般的玩家幾乎都是透過修改這些外部檔案來產生新的bot，不過接近無限的行為權重排列組合是需要長時間嘗試、測試與調整的。此外在外部bot 檔案中包含了有bot 的各項能力描述例如射擊準確度、移動速度、反應時間等。我們將會在bot AI開放程式碼中加入適當的模組來評估角色在遊戲中的行為結果，並且利用粒子群最佳化演算法產生新的權重來修改外部檔案中關於bot 行為描述的部份。

[botName]_c.c 這個外部檔案主要是用來描述一個bot 之各項能力值與行為偏好之權重，然而對我們來說，能力並不是我們希望調整的

值，策略與行為選擇之偏好才是應調整的權重，在這個檔案中與行為相關的權重包括了有，對戰時該採蹲伏或跳躍、該積極進攻或採取防守、是否埋伏來狙擊對手、是否恃強凌弱、是否採取報復行為、武器切換的頻率等等，因此問題解的表示則為：

(Croucher, Jumper, Weapon jumper, Aggression, Selfpreservation, Camper, Vengefulness, Frag easy target,)



4.4.3 最佳化相關參數與適應函數之設定

這些影響行為的權重可以由0到1之間的實數來表示。在這裡我們所要解決的問題是粒子的表現無法由個人的主觀感受來判斷，而是更需要有更嚴謹的適應函數來決定，此外除了適應值能夠客觀決定之外，bot在遊戲中的表現還會受到能力與運氣等不確定因子的影響，適當的參數設定也相對來說更加的重要。

以下為參數與適應函數之設定：

- 粒子數：該使用多少粒子數將與解空間的大小有直接的關係，解空間越大需要越多的粒子，也需要更多的運算資源，因此我們建議盡量採用較關鍵的行為權重進行編碼來減少解空間的維度。

此外，我們實驗的方式是將修改之後的bot 放入遊戲競技場中與QUAKE 本身的bot 對戰並評估結果，為了讓遊戲中不確定因子的影響減低，一個粒子必需經過多次的對戰，由於我們無法對QUAKE 的遊戲速度進行調整，因此這是花時間的一項程序，在以一粒子群代表一個bot 行為的實驗上，我們取10左右之粒子數，如有能力加速遊戲進行的速度或不在意這項程序所需的時間，可以取更多一點的粒子數來求得更好的結果。

- 粒子的移動範圍與最大速度：粒子的移動範圍設定為行為權重之大小，在0~1之間，而粒子的最大速度則設定為在0.2~0.5之間，不設定為粒子的最大移動範圍是為了要讓粒子更具有穩定學習的能力。
- 學習因子：依據一般文獻的建議，將c1與c2 設為 2。
- 在全體最佳值(gbest)或區域最佳值(lbest)的選擇上，混合的作法仍然是個不錯的選擇，也就是一開始採用區域最佳值進行較廣域的搜尋，後期採用全體最佳值來讓結果較能收斂，不過由於效率

問題本實驗所取的粒子數並不多，僅採用全體最佳值來進行搜尋。

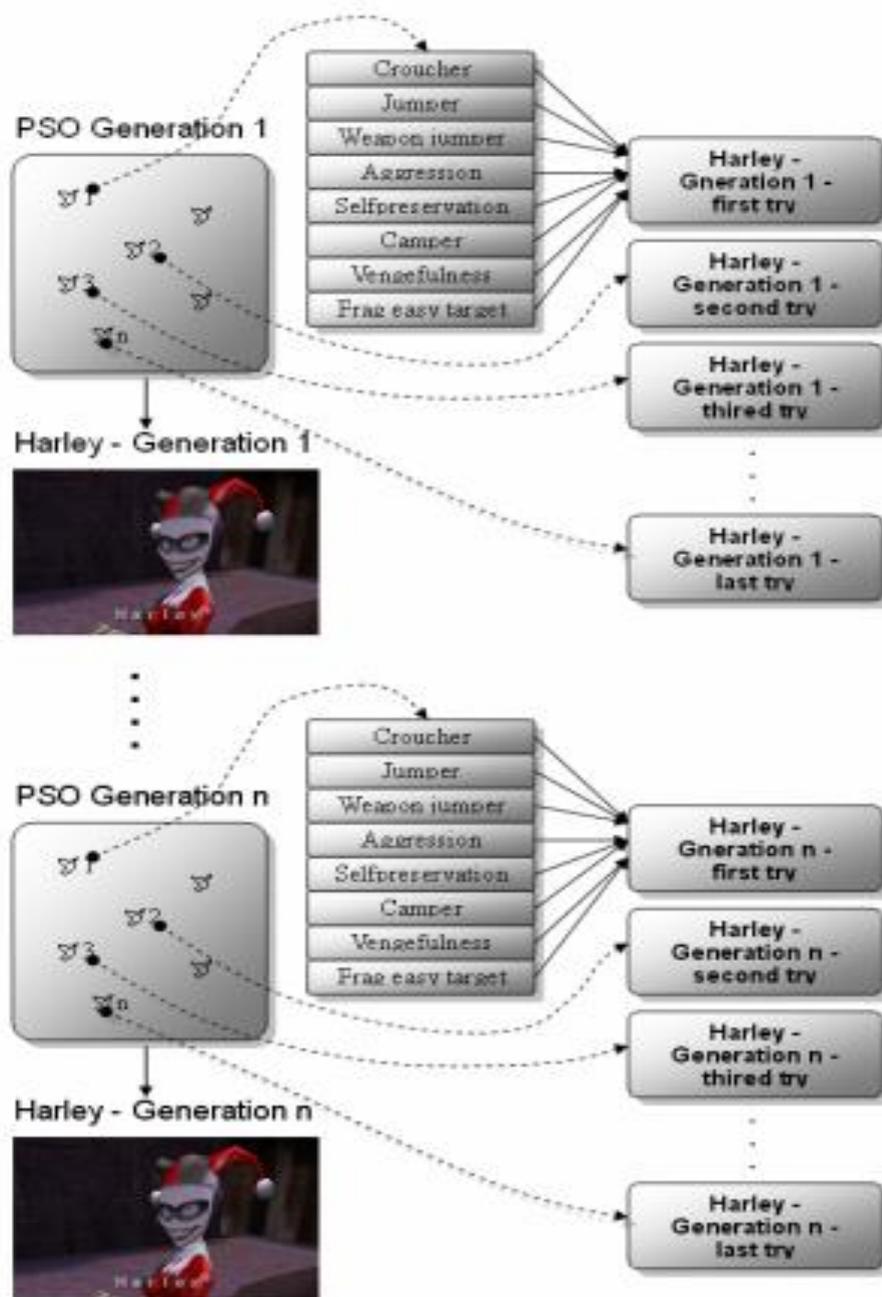
在適應函數的決定上我們採用下列的公式：

$$\text{FitnessValue} = 2.0 \times \text{num_kills} - 1.0 \times \text{num_deaths}$$

也就是希望我們的bot 能夠擊倒越多的敵人，並且盡量讓自己不被敵人擊倒。適應函數會直接引響bot 的行為表現，而適應函數的選擇則要考慮到遊戲性的設計，我們可以藉由不同的適應函數來設計出具有各種不同特色之電腦對手，例如聰明、勇猛、膽小與難纏之電腦對手。

4.4.4 實驗設計與過程

在本 QUAKE III 之BOT 行為最佳化的研究我們利用一個粒子群來代表一個bot 之行為。以一個粒子代表此bot 的一次嘗試，將此bot 放入競技場中與其它QUAKE內建的bot 進行對戰，也就是說每一代的bot 會選擇 n 次不同的方法來嘗試擊倒對手，每一次的嘗試會得到一個適應值，最好的嘗試結果會成為下一代中各個嘗試的參考，每一次的嘗試也會受到之前幾代中相對應嘗試中最好一次的引響，過了幾代之後bot 將會找到擊倒對手之較好策略，這與PSO 的概念是間接的對應。



以一個粒子群代表一個bot之行為



從本實驗的結果可以發現，我們所產生之bot 確實具有因學習而成長之能力，在三人自由對戰模式(free for all)以及地形稍微複雜的場景(q3dm13: lost world)中經過二十代左右的演進，我們所產生之bot行為普遍與上個實驗的結果相同，不同的是此實驗所產生的bot 在對戰時僅量採蹲伏的姿勢以增加射擊穩定度、不積極進攻並且僅可能採取防守以自我保護武器切換的頻率高等。

此外成長曲線方面，此次的實驗結果比起上個實驗bot 的學習能力較

為穩定，雖然仍然具有一些運氣因素影響適應值的評定，不過少了同時具有學習能力之bot 一同對戰可讓適應值的評定更加準確，以一個粒子群代表一個bot 之方式會更適合在測試階段輔助AI 設計師找到bot 較佳之行為權重。

結論

隨著科技的進步，遊戲設計也越貼近現實，相對的考慮到的因素也就相當多樣化，取得勝利是我們的目標，伴隨這目標讓我們想要繼續遊玩下去的動力就是多樣化的互動，感受著與NPC之間互動不再是像以往單調死板而是充滿著趣味以及挑戰性。第一人稱射擊遊戲考慮的因素越來越多，但不管如何最終一定存在著「殺人」與「被殺」的基本元素，本文把問題回歸到最基本排除其他外在條件，單純考慮「殺人」與「被殺」兩者間對整體遊戲生態的影響，並且利用粒子群演算法最佳化技術動態調整人物腳色的行為參數的值，讓人物腳色面對這兩問題時具有決策的能力，最後將我們選取的操作點分別與電腦NPC對戰20場、30場、40場並且觀察他們的結果。經由適當的應用探討射擊遊戲領域中關於角色行為設計上的部分問題，並有良好的成果。

我們期望這樣的方法可以在遊戲開發期間輔助遊戲人工智慧設計師在調整角色行為之間相互平衡的問題，節省測試不同參數組合的時間，增進遊戲開發的效率，並在遊戲進行時讓玩家感受角色具有調整行為之學習能力與多樣化的行為能力，並且豐富玩家在面對這款遊戲的挑戰度與互動性。

參考文獻

Quake III: Arena, id Software, www.idsoftware.com

S. Cass, "Mind Games: To beat the competition, video games are getting smarter," IEEE SPECTRUM,
<http://www.spectrum.ieee.org/WEBONLY/publicfeature/dec02/mind.html>

S. Rabin, "AI Game Programming Wisdom: Preface," AI Game Programming Wisdom, Charles River Media Press, pp. xi-xiv

R. Adobbati et al., "GameBots: A 3D Virtual World Test-Bed for Multi-Agent Research

"First-person shooter engine"

<http://zh.wikipedia.org/zh-tw/%E7%AC%AC%E4%B8%80%E4%BA%BA%E7%A8%B1%E5%B0%84%E6%93%8A%E9%81%8A%E6%88%B2%E5%BC%95%E6%93%8E>

<http://zh.wikipedia.org/zh-tw/%E9%9B%B7%E7%A5%9E%E4%B9%8B%E9%94%A43>