基于雲端運算之多機器人控制方法 應用於垃圾清潔問題

徐銘鴻　Ming-Hung Hsu
玄奘大學資訊管理研究所學生
Email:wahaha1002@msn.com

姜正雄　Cheng-Hsiung Chiang
玄奘大學資訊管理研究所助理教授
Email:chchiang@hcu.edu.tw

摘要

本研究提出一種基於雲端運算的多機器人控制方法應用於垃圾蒐集任務。先由伺服器端處理收集任務相關資訊並計算每台機器人的路徑再發送至各機器人。在伺服器端，我們使用類神經網路來產生每台機器人的路徑。當機器人清潔完某個區域要移動至另一個區域時，我們應用類免疫式粒子群優化演算法產生一條較短路徑來使之移動。每台機器人將依照伺服器規劃好的路徑導航，途中發現垃圾時會撿起。當機器人的垃圾乘載量達到上限時，機器人會請求伺服器計算前往垃圾桶的路徑。由模擬結果顯示多機器人的執行效率優於單機器人，此外，在電影院的環境中，該方法也能有效的使四台機器人完成清潔任務。

關鍵詞：雲端運算、多機器人控制、類神經網路、類免疫式粒子群優化、路徑規劃

Abstract

This paper proposes a control approach of multiple robots based the concept of clouding computing. The proposed control method is applied to the trash collecting task. The server that guides the robots to execute tasks firstly collects related information and computes the path for each robot. In the server, we applied the artificial neural network to generate the navigation path for each robot. If the robot cannot move anymore and there exists the unclear area, the robot has to move to the unclear area for cleaning task. We applied the immune-based particle swarm optimization for generating a shorter path for above situation. Next, each robot will begin to navigate according to the designed path generated by the server. If the robot finds trash, it will pick it up. While the amount of trash the robot carried reaches the upper bound, the robot will request the server to compute a path to the garbage can. The simulation results showed that model of multiple robots required less time steps than the model of single robot for cleaning task. In addition, in the environment of movie theater, the proposed method could efficiently guide the four robots for cleaning task.

Keywords: Cloud computing, Multi-robot control, Artificial neural network, Immune-based particle swarm optimization, Path planning

1. 緒論

科技的進步日新月異，機器人的應用不僅限於軍事及工業上，也逐漸進入了人們的日常生活中分擔許多的工作。在環境清潔的問題中，市面上已經有許多居家清潔機器人的產品，如：日立清潔機器人、微星科技 iCleaner、益眾科技 RV-13 等商品。但目前市面上所販賣的這類清潔機器人的產品，並不是像想像中這麼的完善。這類產品由於他們的移動方式完全藉由小輪子的旋轉而帶動，往往只能侷限在平面的地板上運動，無法在一些顛簸路面甚至是階梯上運作。另外，這些產品都只有單機的特性，即使有兩三台清潔機器人同時運作，工作效率也不一定會比較好。如果要增加機器人的清潔效率，則需要加入一套可以使他們分工合作的方法。

清潔機器人在試圖造訪全部區域的移動中，路徑規劃、導航及避障一直是受到廣泛討論的問題。Howie Choset [10]提出牛耕田式(boustrophedon)的全域覆蓋路經規畫法(complete coverage path planning)，使用 8 方位的移動來達成在指定空間內拜訪所有正方格的目的，如圖 7 所示。而 Oh 等人[13]以三角方格為基礎的地圖取代傳統的以正方形方格為基礎的地圖，使移動方位由 8 向增加為 16 向，大大的增強了導航的精緻度。de Carvalho 等人

[14]是利用幾組已預先設定好的路徑範本(template)，針對不同的地形使用特定的範本完成全域覆蓋的路徑規畫。Yang 以及 Luo [16]則是使用類神經網路(artificial neural network)來做導航，將地圖上的任何一點都視為神經元，而機器人所在點在與其相鄰的 8 個點做運算，計算出適合的路徑。

為了達到更有效率的全域覆蓋路徑規劃，本研究採用多機器人同時進行巡航，以節省全域路徑規劃的時間。在機器人的控制與協調方面則採用雲端計算(cloud computing)的概念，用以指揮機器人巡航的任務。機器人的移動採用類神經網路(Artificial Neural Network, ANN)來控制，當機器人遇到無法移動而且尚有未拜訪的區域時，則以類免疫式的粒子群優化演算法(Immune-based Particle Swarm Optimization, IPSO)來規劃一條較短路徑，引導機器人移動至未拜訪的區域繼續清潔任務。

2. 垃圾清潔問題

本研究設計一個垃圾清潔的問題，機器人必須完整清潔指定的區域，使該區域保有整潔的環境。機器人在指定的區域中巡航，當遇到垃圾的時候則撿起，當垃圾數量到達機器人可裝載數量時，機器人必須至垃圾桶丟棄，如圖 1 所示。在本研究中，地形資料、垃圾桶位置與機器人數量為已知的，垃圾的位置與數量皆為未知。因機器人本身並不知道環境中哪裡有垃圾，所以機器人的移動方式是以類似牛耕田的模式[10]掃描全部的可行區域以拾起垃圾。又因清潔機器人不只一台，若機器人重複經過其它機器人已清潔過的路徑，就會變成無意義的步驟。因此必須要使用一種合作的方法，使機器人之間不會重複導航於已清潔過的路徑。

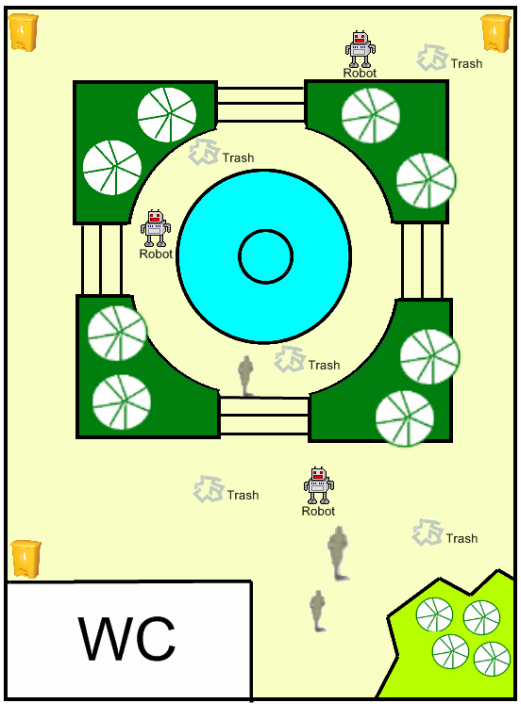

圖 1. 環境示意圖

3. 雲端計算的概念

早在 2004 年的暑假，經典科幻電影「機械公敵」裡就已經大量的使用雲端技術在控制機器人。像是軟件的更新、團隊合作、甚至是鎖定追殺目標都是雲端的主機所下達的指示。而這裡所說的雲端只的是一種抽象的概念，就像一個年長的智者，他的腦袋裡許多寶貴的經驗與智慧，但他的這些經驗與智慧你不一定全都用得到，當你需要知道某些經驗或某些智慧實在向他請教就可以了，亦即為一種資源共享的概念[5]。

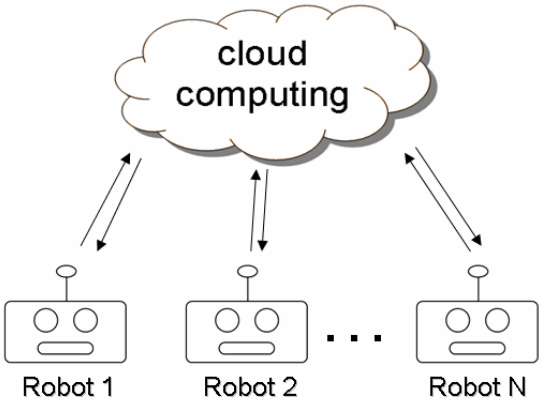

圖 2.雲端運算架構示意圖

雲端科技並不是什麼突破性的技術，早在好幾年前就開始使用的 Internet 就是雲端科技的應用之一，而我們常常用到的 ATM 也是。最近也有防毒軟體把雲端的技術應用於其中，這項技術叫做「雲端鑑識技術」[2]。而未來雲端技術發展重點也應該朝向於可以發展出什麼樣創新的應用。而在本研究中提出的雲端應用正是要解決機器人與機器人之間團隊合作的問題。試想想，機器人在一個已經排程好的路徑下移動時，若在移動的途中發生一個優先度高於移動指令的事件時該怎麼處理，在單機器人的導航環境中可能比較簡單處理，但在多機器人導航時就需要考慮到機器人與機

器人之間的互相配合了。

在本研究的設計裡，機器人彼此之間是不需要溝通的，每台機器人的導航移動都是由伺服器端事先運算，再把每台機器人個別的路線傳送至各個機器人。每台機器人只需要接收各自的路線以及執行它就可以，並不需要對路徑規劃做多餘的運算，這樣的方法也可以避免機器人與機器人之間複雜的溝通協調問題。

4. 導航規劃方法

在 Bai and Low 的論文中[15]提出機器人路徑規劃系統應分為五個步驟執行，分別為任務規劃、路徑規劃、步態規劃、腿的運動規劃、驅動馬達控制。本研究不涉略硬體方面的控制，只著重於任務規劃、路徑規劃及初步的步態規劃。任務規劃是完成全區域環境清潔的全域式路徑規劃。路徑規劃即根據機器人特定任務生成一條可行的路線。步態規劃方面以機器人最大移動步伐內取一個中間值，若是機器人一步最多可以跨越 7 個方格時，將取 1 至 7 格之中的一個數作為一次跨越的距離，同時也要考慮到每個步伐跨越之高度是否為可行的。

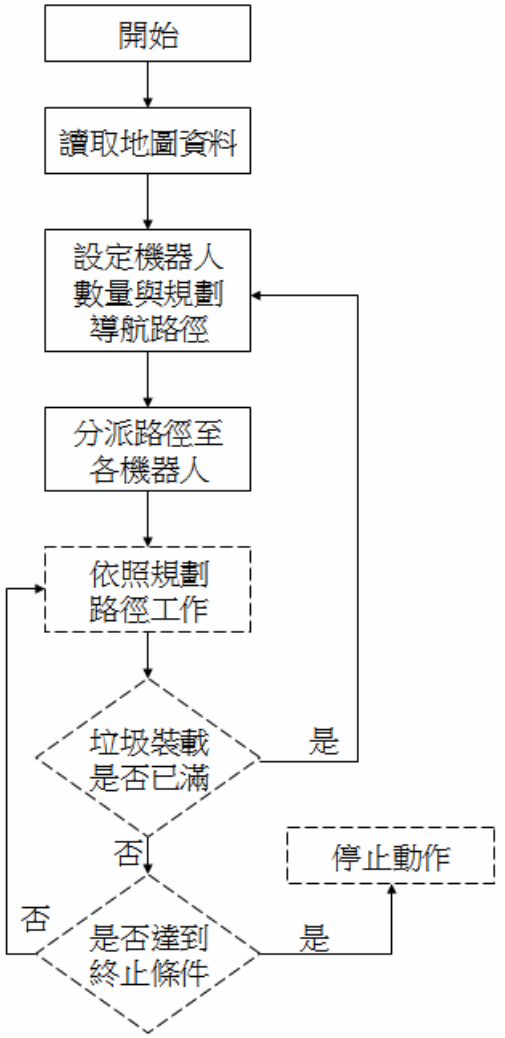

圖 3. 導航規劃流程圖

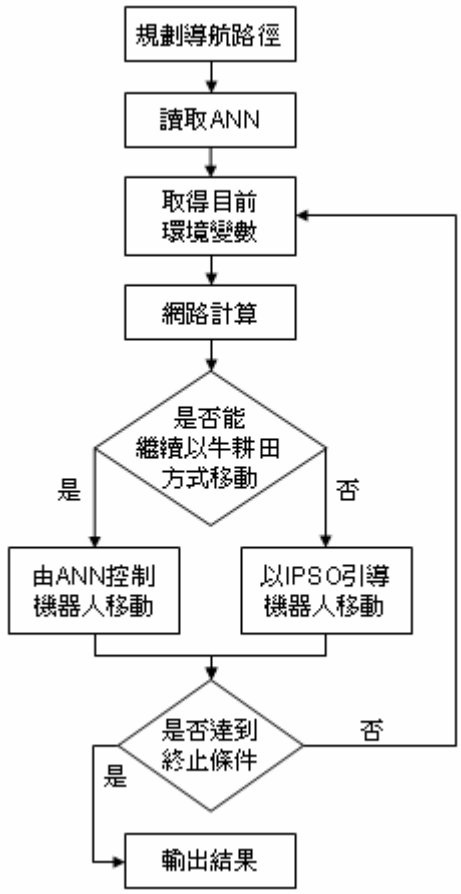

圖 4. 路徑規劃流程圖，為圖 3 之規劃導航路徑步驟

4.1 導航流程

本研究提出的導航規劃方法，如圖 3 所示。整個流程大致上分為兩個規劃步驟：1) 伺服器端的路徑計算及分派(如圖 3 實線部分)；2) 機器人的實際執行(如圖 3 虛線部分)。整個流程的一開始先讀取地圖資訊，至規劃導航的步驟(如圖 4)時讀取一個已訓練好的網路。以這個網路為基礎規劃路徑，路徑規劃的這個步驟是個迴路，需規劃 N 台機器人時，則每個步驟就需要 N 個迴路(如圖 4)。

路徑規劃的流程，整體的結構是以類神經網路(Artificial Neural Network, ANN)的判斷為主。ANN 會先判斷機器人是否能繼續使用已學習好的網路做導航(這是一種類似牛耕田的方式控制機器人的移動)。若機器人不能移動(例如：周圍的區域都已經拜訪過)，則呼叫類免疫式粒子群優化演算法(Immune-based Particle

Swarm Optimization, IPSO)，搜尋從目前位置到未拜訪(覆蓋)區域的較短路徑。並且讓機器人依據所搜尋到的路徑移動。

伺服器端將所有路徑規劃完成之後會把個別的路徑派給個別的機器人，機器人只要依照規劃好的路徑工作。即使不必相互溝通依然能達成分工合作的目的。當然，在路徑規劃的步驟時已經有避免相互碰撞的機制了，亦不用擔心互相碰撞的發生。

而在機器人執行全域清潔時，可能會發生需要改變路徑的情況。例如：垃圾的乘載量已滿，需移動至垃圾桶丟棄；或是在風景區遊客請求帶路等等的情況時。此時機器人只需要對伺服器端發送一個請求，伺服器端就會依照目前執行到第幾個步驟，重新規劃其它 N-1 台機器人的導航路徑。並以 IPSO 方法規劃該台機器人移動至目的地的路徑，而該機器人完成特殊指令需重回排程時，也是依照上述方式加入序列當中。

4.2 類神經網路

ANN 自 1943 年 McCulloch 與 Pitts [17]提出第一個人工神經元模型至今，已經應用得非常廣泛。例如：最佳化問題、聯想問題、預測問題、辨識等問題，舉凡非線性的問題或是非常複雜難以以數學方式描述求得解答的問題都能使用 ANN 來完成。Rumelhart 等人[9]在 1986 年提出基於倒傳遞學習法則的倒傳遞類神經網路(Back Propagation Neural Network, BPNN)，這是一種多層的網路架構，以誤差倒傳遞的方法來學習調整加權值。由於 BPNN 具有容易使用的特性且能夠有效解決非線性的問題，因此本研究採用此種 BPNN 作為機器人移動的控制機制(如圖 5 所示)。

ANN 內的神經元具有非線性的運算功能，神經元與神經元彼此間互相連結，這些神經元通常是以平行且分散的方式來進行運算。ANN 中常使用的活化函數有四種：步階函數、片段線性函數、S 型函數和雙曲線函數，本研究以 S 型函數作為活化函數。網路輸出的算法如公式(1)，X_i 為第 i 神經元的輸入值，W_i 為第 i 神經元輸出的加權值，公式(2)為 S 型活化函數，倒傳遞學習的原理是利用誤差(目標輸出值和實際輸出值的差)來調整加權值，公式(3)為誤差的計算公式，d 為理想輸出值 y 為網路實際輸出值，為了要使誤差最小化，定義加權數值改變量如公式(4)。

$$y_j^n = f(net_j^n)$$

$$net_j^n = \sum_{i=1}^{n} W_{ij}^n X_i^{n-1} - \theta_j^n \quad (1)$$

活化函數為：

$$f(net) = \frac{1}{1+e^{-net}} \quad (2)$$

計算誤差 E 公式：

$$E = \frac{1}{2}\sum_n (d_n - y_n)^2 \quad (3)$$

修正加權值 w 公式：

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} \quad (4)$$

本研究提出的 ANN 為兩個輸入及一個輸出值，如圖 5 所示。輸入值 X1 為與機器人目前位置相對映的周遭地形狀況；X2 為機器人上一步動作之回傳值，作為輔助判斷的用途；Y 為網路判斷後的輸出結果，因為區域填滿的導航通常只會使用到上下左右的四，很少會用到左上、左下、右上、又下的四個指令，因此將 Simon X.Yang 等人[16]提出的八向相對點運算縮減為四向相對點運算，這樣的作法不僅能大量縮減培養神經網路所需的資料，且因為每次的移動需計算所有神經元的活動值也因輸入量的減少，所需使用的神經元相對可以在不影響收斂的情況下減少，而使得每次所需的計算量減少不少；當需要使用到非牛耕田方式導航時，神經網路會呼叫 IPSO 以點對點的方式完成移動，這個動作將會在下一章節介紹。

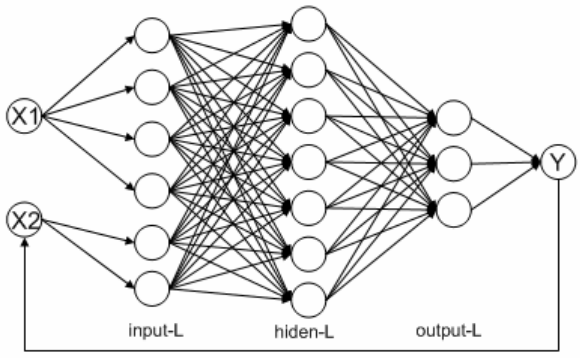

圖 5. 類神經網路架構圖

4.3 類免疫式粒子群優化演算法

粒子群優化演算法(Particle Swarm Optimization, PSO)最早是由 Kennedy 和 Eberhart [11], [12]藉由觀察鳥群覓食的行為得到啟發而提出的方法。是一種具有群體智慧概

念的一種計算方式，在人工智慧領域中擁有相當不錯的評價[6]。在演化過程中，每個粒子各代表一個解，藉由評估函數計算出每個粒子的優劣，每個粒子會記住自己的最佳位置以及群體目前的最佳位置，根據這兩個位置計算出每個粒子的速度以決定未來的位置和方向。以下為計算速度的公式：

$$V_{id} = w \cdot V_{id} + c_1 \cdot \text{Rand}() \cdot (P_{id} - X_{id}) + c_2 \cdot \text{Rand}() \cdot (P_{gd} - X_{id}) \quad (5)$$

$$X_{id} = X_{id} + V_{id} \quad (6)$$

其中 V_{id} 為第 i 個粒子在第 d 維度之速度，w 為一慣性權重。c_1 和 c_2 為學習常數，通常設定為 c_1=c_2=2，Rand()為在[0, 1]範圍裡的隨機變化值。P_{id} 為第 i 個粒子在第 d 維度到目前為止所出現的最佳位置，P_{gd} 為所有粒子到目前為止所出現的最佳位置。X_{id} 為第 i 個粒子在第 d 維度目前之位置。每個粒子在搜索空間中以一定的速度飛行，這個速度根據它本身的飛行經驗(P_{id})和同伴的飛行經驗(P_{gd})來動態調整。

與其它的演化方式相比，PSO 演算法具有收斂速度快、設置參數少、程序結構簡潔等優點，特別是和科學研究及工程應用[1][3][6]。但是由於 PSO 在優化過程中所有粒子都會往 P_{gd} 的方向飛去，群體的多樣性就逐漸喪失。這樣的情況一旦遇到全域最佳解與區域最佳解的位置在整個搜尋空間的兩極時，只要初次灑粒子時無法灑在全域最佳解的方向時，就會因為 PSO 有快速收斂的特性而錯誤的收斂至區域最佳解的位置[1][3][6]。如圖 6 所示

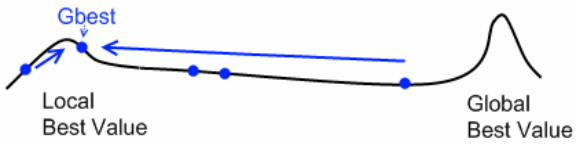

圖 6. 粒子群優化演算法的錯誤收斂

類免疫式的粒子群優化演算法(Immune-based Particle Swarm Optimization, IPSO)以類免疫演算法(artificial immune algorithm)為基礎來避免粒子陷入區域最佳解中。藉由免疫反應中抗體辨認抗原的專一性而將問題目標值視為抗原，另外將問題之設計解視為抗體。針對抗原來搜尋可與之相結合的抗體亦即尋找問題之解答，抗體必須針對未知的抗原來做演算，以找出最符合最佳解之抗體[6]。本研究採用陸克中等人[3]所提出來的 IPSO 方法，流程如下：

步驟 1. 產生初始的粒子族群；
步驟 2. 計算每個粒子的位置與適應值；
步驟 3. 計算並更新 P_{id} 值；
步驟 4. 計算並更新 P_{gd} 值；
步驟 5. 判斷粒子的活性，若粒子活性小於一個閥值則將最佳解視為抗原並給予激活；
步驟 6. 若未達到停止條件則返回步驟 2。

本研究提出的粒子激活的演算法修改自[3]，如下所示：

```
For i = 1 to popsize,
 If P_cownt(i) >= iter_num,
  If rand()< Pm ,
   For j = 1 to m,
    X(i,j)=(up(i,j)-low(i,j))*rand()+ low(i,j)
    V(i,j)=0;
   End_For
  Else
   For j=1 to m,
    X(i,j)=(rand()-0.5)/Pa*X(i,j)+X(i,j)
    V(i,j)=(rand()-0.5)/Pa*V(i,j)+V(i,j)
   End_For
  End_If
  Regenerate Pid randomly
 Else
  Update Vid and Xid by Eq.(5) and (6)
 End_If
End_For
```

其中，rand()為一個在[0,1]範圍裡的隨機變化值，P_m 表示粒子的異變機率，popsize 為粒子群種的大小，m 為粒子的維度，Pa 是一個常數用來控制激活粒子的範圍，up(i,j)與 low(i,j)分別表示當前演變過程中第 i 個粒子在第 j 維的上限與下限，P_cownt(i)用於記錄粒子 i 未取得進展的次數，iter_num 為一閥值。

而 ANN 與 IPSO 的合作方式大致如圖 7 所示，藍色實線表示 ANN 規劃之路徑，紅色虛線表示 IPSO 之規劃路徑。首先由 ANN 以牛耕田的方式控制機器人的導航。當機器人無法再移動時(如圖 7 的左圖)，則需搜尋未拜訪區域中，距離目前位置最短的路徑，以便繼續清潔

這部份的區域。這時候，IPSO 便用以搜尋一條較短的路徑，如圖 7 右圖所示。接著，再由 ANN 繼續控制機器人的導航工作。重複以上步驟，直到所有區域都清潔過為止。

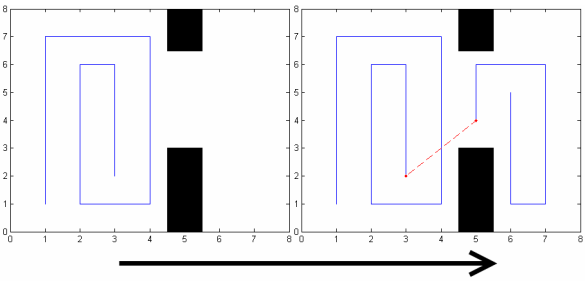

圖 7. ANN 與 IPSO 配合方式示意圖，藍色實線為 ANN 之導航，紅色虛線為 IPSO 之導航

5. 模擬實驗與結果

5.1 模擬實驗(一)

我們以一 100*100 的無障礙區域測試多機器人合作是否比單機器人運作的效率更好。每次實驗機器人的出發位置為分別為四個角落，並且隨機場地中灑垃圾，而每台機器人的垃圾負載量為 5 個。每當機器人撿滿 5 個垃圾時就必須移動至垃圾桶丟棄，而垃圾桶的位置也放置於場地四個角落。機器人每一步最多可跨越 7 個方格；以 ANN 導航時，機器人每一步伐固定以 4 個方格為單位，IPSO 導航時則視情況以 1 至 7 個方格為單位。每台機器人可偵測到障礙物以及垃圾的範圍為 7×7 個方格，也就是以機器人所在位置為中心點，往上下左右各延伸 3 個方格的正方形範圍。實驗中除了機器人數量不同外，其餘條件皆相同。圖 8 為導航結果。

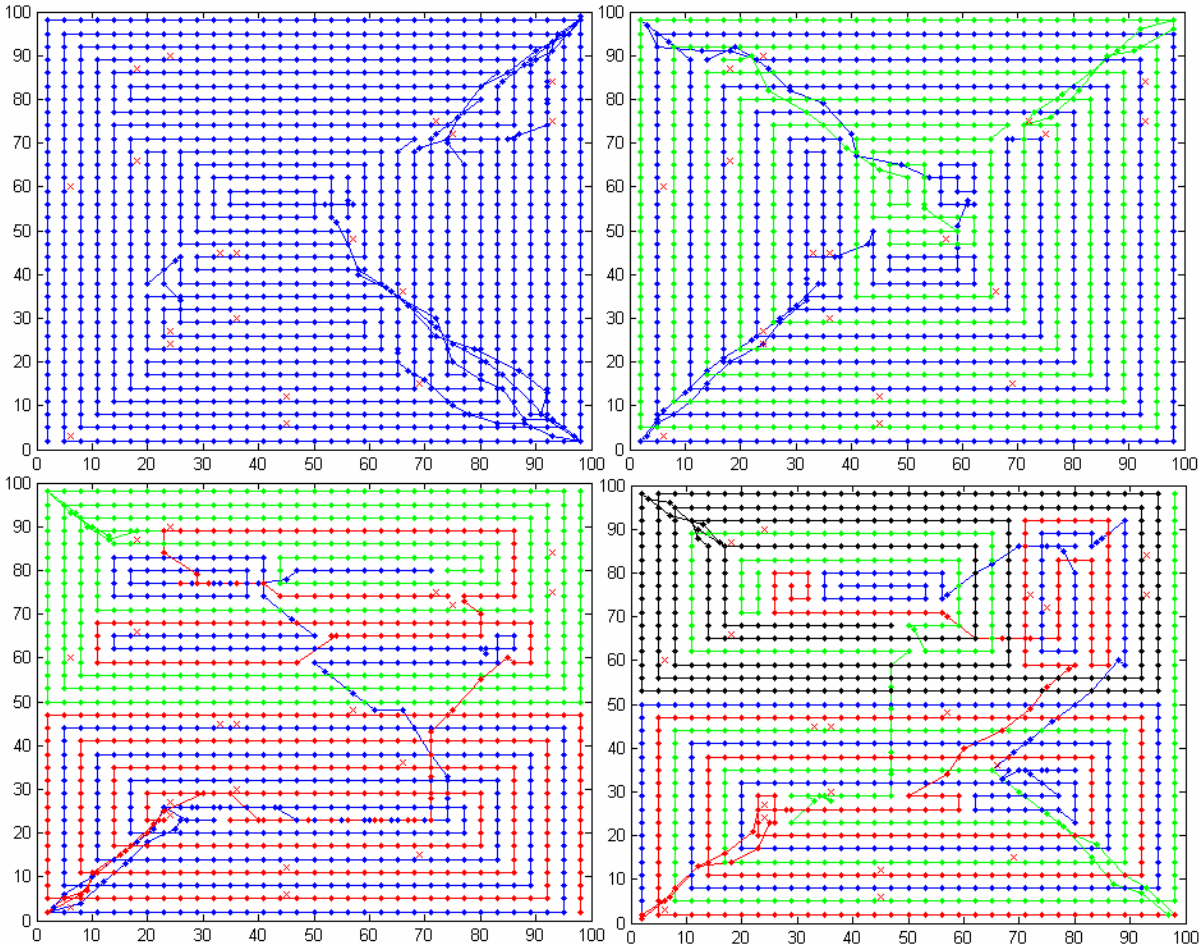

圖 8. 模擬結果：左上圖、右上圖、左下圖、右下圖分別為 1、2、3、4 台機器人的模擬 (×為垃圾)

圖 8 依序為 1 至 4 台機器人的導航結果，其中移動所需花費的時間，以 time steps 為單位。單機器人導航時需要 1201 個 time steps 來移動，雙機器人導航時需要 589 個 time steps，三機器人需要 407 個 time steps，四台機器人導航需要 306 個 time steps。可發現相較於單機器人的狀況下，每增加一台機器人則所需的 time steps，將會以接近 1/N (N 為機器人數量)的方式縮短。這證明了多機器人的導航確實可以有效降低全域清潔所需花費的時間。

5.2 模擬實驗(二)

實驗場景：電影院
機器人數量：4

此實驗將原本 2D 的場景推廣至 3D 的場景做模擬，其的目的為測試多機器人導航是否能成功的合作導航於多障礙的 3D 環境。我們以一個電影院的場景來測試機器人的導航能力。其中場地平面圖是擷取自威秀影城[7](圖 9 左)，再依照此圖的比例將影院虛擬化，如圖 9 右。就如同模擬實驗(一)，我們首先隨機在電影院中灑垃圾。每台機器人的垃圾負載量為 5 個，每當機器人撿滿 5 個垃圾時就必須走至垃圾桶丟棄。垃圾桶共有 2 個，分別位於螢幕前的最左邊及最右邊(如圖 10 的左下角與右下角位置)。每台機器人的出發位置為地圖的四個角落，機器人每一步最多可跨越 7 個方格。以 ANN 導航時每個步伐固定以 4 個方格為單位，IPSO 導航時則視情況以 1 至 7 個方格為單位，每台機器人的偵測範圍為 7×7。

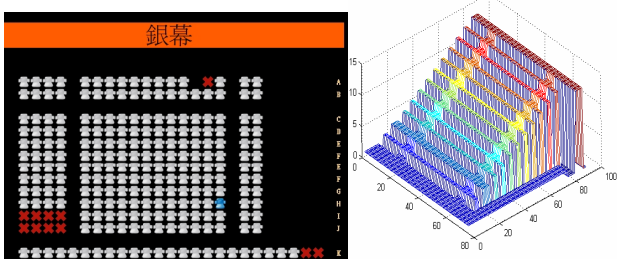

圖 9. 模擬實驗場地模型

圖 10 為導航結果，因導航路徑較難以立體圖表示，故以模擬鳥瞰的方式以平面圖來做顯示。由此圖可知，在場地已知且垃圾未知的情況下：四台機器人除了在倒垃圾的路徑及一些必經的路徑之外，很少會遇到有重複走的路徑。表示此路徑規劃可以有效率的控制多機器人，不但能夠有效的分工合作於電影院之環境；另外，由模擬實驗(一)可知，多機器人的合作模式也能夠減少清潔所需花費的時間。

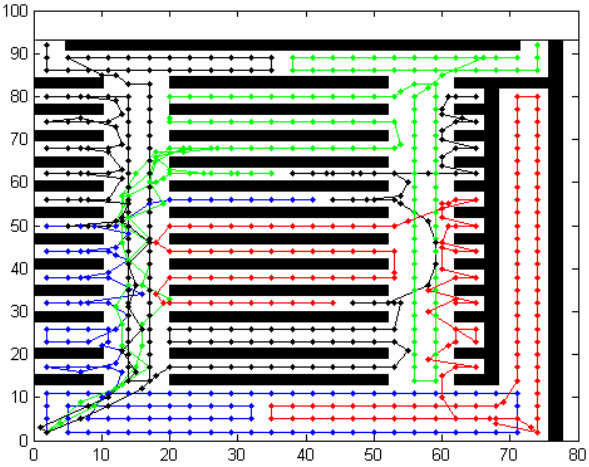
圖 10. 實驗結果（鳥瞰圖）

6. 結論

本研究提出一種基於雲端計算概念的多機器人控制方法，並應用於垃圾清潔問題。透過伺服器端的資料運算處理，運用類神經網路與類免疫式粒子群優化演算法計算出路徑，再將計算完的資料傳送至各機器人。這樣的方法不僅使機器人達成分工合作的目的，也解決了機器人之間複雜的溝通問題。多機器人的控制技術使他們比單一機器人有著更高的效率，能有效的縮短全域清潔所需的時間，除此之外也使其能有更多的時間提供更多清潔問題以外所需的服務。此外，本研究還提出一種以模擬步行移動的導航方法，利用模仿人類步行移動的方式藉以克服高低參差不齊的地形，雖然目前市面上幾乎還沒有此類的機器人在量產販售，但相關的研究正在不斷的進行[4][8]，相信有朝一日此方法定能在生活中實踐。

而目前本研究提出的路徑規劃方法都是以地形環境已知的情況下去做規劃的，在未來期望可加入以多機器人聯合探索環境的方式，使本研究的路徑規劃方法可用於未知的環境中。

參考文獻

[1] 程培新, 王亞慧, 2009, “應用 IPSO 的無線傳感器網絡分簇路由算法”, *北京建築工程學院電氣與信息工程學院. 北京*。

[2] 諾頓官方網站, http://tw.norton.com.

[3] 陸克中, 王汝傳, 帥小應, 2007, “保持粒子活性的改進粒子群優化演算法”, *計算機工*

程與應用，第 43 卷，第 11 期，35–38 頁。

[4] 謝銘原, 陳建升, 莊承鑫, 2010, "人形機器人步態偵測及穩定控制系統之研究", *第五屆智慧生活科技研討會*。

[5] 陳宗天, 張志峰, 2010, "雲端運算之主要研究主題", *國立台北大學資訊管理研究所碩士論文*。

[6] 蔡清欉, 藍卞鴻, 2008, "以免疫演算法為基礎之 PSO 於電腦遊戲團隊人工智慧之研究", *東海大學資訊工程與科學系碩士論文*。

[7] 威秀影城,http://www.vscinemas.com.tw.

[8] 余佳擁, 陸冠群, 2004, "二步足行機器人的設計製作與軌跡規劃", *大同大學機械工程研究所碩士論文*。

[9] D. Rumelhart, G. Hinton, and R. Williams, 1986, "Learning representations by back-propagating errors," *Nature,* Vol. 323, pp. 533–536.

[10] H. Choset, 2000, "Coverage of Known Spaces: The Boustrophedon Cellular Decomposition", *IEEE Autonomous Robots,* Vol.9, No.3, pp.247 – 253.

[11] J. Kennedy and R. Eberhart, 1995, "Particle swarm optimization", *In Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, USA, pp.1942 – 1948.*

[12] J. Kennedy and R. Eberhart, 1995, "A New Optimizer Using Particle Swarm Theory", *IEEE Micro Machine and Human Science,* pp.39 – 43.

[13] J. S. Oh, Y. H. Choi, J. B. Park, and Y. F. Zheng, 2004, "Complete Coverage Navigation of Cleaning Robots Using Triangular-Cell-Based Map", *IEEE Transactions on Industrial electronics,* Vol. 51, No.3, pp. 718 – 726.

[14] R. N. de Carvalho, H. A. Vidal, P. Vieira, M. I. Ribeiro, 1997, "Complete Coverage Path Planning and Guidance for Cleaning Robots", *Proceedings of the IEEE International Symposium on Industrial Electronics,* Vol.2, pp.677 – 682.

[15] S. Bai and K. H. Low, 2001, "Terrain evaluation and its application to path planning for walking machines", *Advanced Robotics,* Vol. 15, No. 7, pp. 729 – 748.

[16] S. X. Yang and C. Luo, 2004, "A Neural Network Approach to Complete Coverage Path Planning", *IEEE Transactions on Systems, Man, and Cybernetics, Part B,* Vol.34, No.1, pp.718 – 724.

[17] W. S. McCulloch and W. Pitts, 1943 "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biology,* Vol. 5, No. 4, pp.115-133.