

強化角度因子之啟發式蟻群演算法 應用於動態環境中存活力與適應力之研究

The Research of Survivability and Adaptability in Dynamic Environment of Heuristic Ant Colony System with Angle Factor for Mobile Robot Path Planning

劉豐豪
國防大學管理學院
資訊管理學系
lfh123@gmail.com

羅祥福
國防大學理工學院
國防科學研究所
superalf@gmail.com

張冠璋
國防大學管理學院
資訊管理學系
j7898521j@gmail.com

摘要

啟發式演算法求得之解較傳統演算法多變，藉此來增加非線性問題的值域，避免落入區域解的巢臼，螞蟻演算法(Ant System, AS)即屬於啟發式演算法的範疇，其藉由啟發資訊以及費洛蒙遺留與蒸發機制來跳脫區域最佳解，對於複雜度高、規模大的問題，可以有效找到最佳解或是近似最佳解，故被應用在機器人路徑規劃問題上。

但現實環境是多變而不可預測的，一般AS的啟發資訊僅依據相鄰節點之距離的長短來決定下一路徑，此一機制在多變的現實環境中並不足以適應環境，達到提升完成任務的比率。

本研究提出一種改良的螞蟻演算法，稱為狡猾的蟻群系統(Shady Ant Colony System, SACS)，結合角度因子改進啟發資訊參數，使機器人在變動的環境中可以規劃一條較快速的可行路徑。

實驗結果顯示 SACS 較傳統 AS 路徑總長短 38.3%、有效性優 85.7%、時效性優 35.3%，明顯較傳統 AS 更具適應力及存活力。

關鍵字：螞蟻演算法、機器人、路徑規劃、動態環境

1. 前言

自走機器人路徑規劃的研究起始於 70 年代，迄今已有大量的研究成果發表[10、17、20]，是機器人研究的核心問題之一，其目的在移動時尋找所處所必須經過的所有點之集合，且使機器人在運動過程中，能無碰撞地繞過所有障礙物[3]。一般來說，機器人路徑規劃方法可以分成 1) 人造力場路徑規劃技術(Artificial Potential Field Path

Planning)、2) 模版匹配路徑規劃技術(Template Path Planning)、3) 人工智能路徑規劃技術(Artificial Intelligent Path Planning)及 4) 地圖建構規劃技術(Map Building Path Planning)。

人造力場法[1、12、15]是模擬機器人處於一"力場"中，障礙物對機器人有"斥力"，目標點對機器人有"引力"，整個力場環境對機器人產生斥引作用，進而讓機器人避開障礙到達目標位置。模板匹配法[8]就是以曾走過的路徑建立模板庫，當在重新規劃路徑時，會比對當前與模板庫路徑的特徵，找出特徵最相似的路徑，再以為之基底修改成當前的路徑。人工智能路徑規劃[3]法是使用人工智慧方法-例如啟發式演算法、執行路徑規劃方法，其中包含基因演算法(Genetic Algorithm, GA)、粒子演算法(Particle Swarm Optimization, PSO)、爬山演算法(Hill-Climbing Algorithm, HC)、模擬退火法(Simulated Annealing, SA)及 AS 等。地圖建構規劃法[6、7、19]是根據機器人的感測器去搜尋障礙物資訊，並將所建構的地圖劃分為可行區及不可行區，再依據一定的規則規劃出最佳路徑。地圖建構規劃法與其他路徑規劃方法常合併使用，如 Najjaran, H.[13]提出的卡爾曼濾波器的地圖建構路徑規劃，Chaomin Luo[18]提出的基於生物啟發神經網路與地圖建構路徑集成的清潔機器人完全覆蓋路徑規劃技術等，Purian [14]提出基於螞蟻及模糊邏輯演算法在未知環境下的機器人路徑規劃等。

自然界中的螞蟻在覓食過程，會沿途分泌一種叫費洛蒙的化學物質，其他螞蟻則會依據前一隻螞蟻所遺留的費洛蒙去尋找食物，這種化學物質會吸引螞蟻呈正相關，其濃度會隨著時間不斷的減少。從雙橋實驗[17]可以看到有兩條不同長

度的路徑連接蟻巢和食物(圖 2-1 a)，一開始螞蟻隨機選擇路徑前進並留下費洛蒙，由於短的路徑在單位時間內螞蟻經過的次數較多(圖 2-1 b)，故費洛蒙濃度較長的路徑高。此消彼長之下，更多螞蟻選擇費洛蒙高的路徑(圖 2-1 c)。

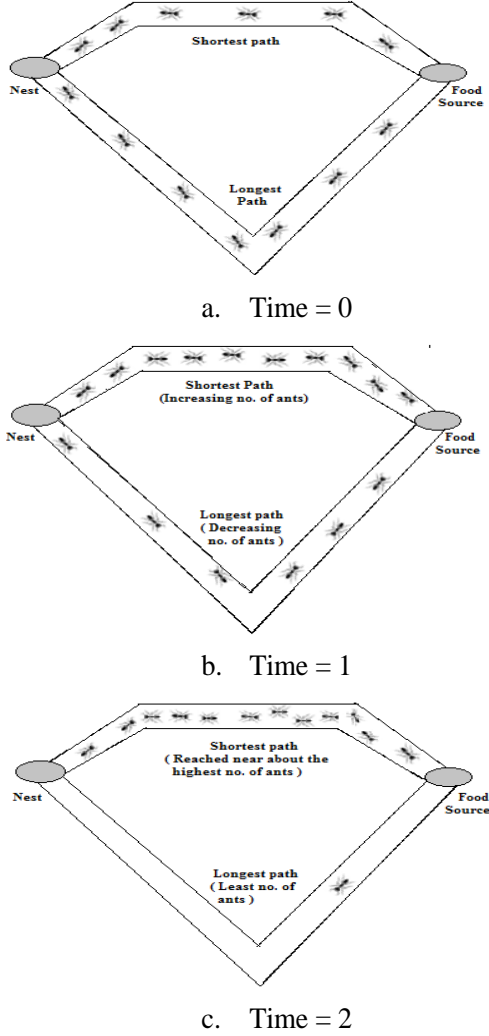


圖 2-1 雙橋實驗(資料來源：[17])

AS 是由 Dorigo[11] 等人於 1996 年發表，其原理是根據費洛蒙及啟發資訊來決定下一路徑的機率有多大，再隨機選取下一路徑，當螞蟻 k 由點 i 移往下一節點 j 的機率 P_{ij}^k 表示如下：

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \times [\eta_{ij}]^\beta}{\sum_{k \in \text{allowed}_k} [\tau_{ik}(t)]^\alpha \times [\eta_{ik}]^\beta} & \text{if } j \in \text{allowed}_k \\ 0 & \text{others} \end{cases} \quad (1)$$

$\tau_{ij}(t)$ ：為路徑 ij 在時間 t 時的費洛蒙濃度。
 η_{ij} ：為啟發資訊，通常為路徑 ij 之距離倒數，表示為 $\eta_{ij} = 1/d_{ij}$ 。
 α ：為費洛蒙對於路徑選擇的比重參數。
 β ：為啟發資訊對於路徑選擇的比重參數。

因此，就算 A 路徑的 P_{ij}^k 較於其他路徑大許多，也不代表一定會選擇 A 路徑，其他路徑也會有機會被選擇，也就不容易陷入區域最佳解。

AS 經過一段時間後，會對費洛蒙濃度進行更新，主要分成兩部分：

1) 遺留(Laying)：

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k} & \text{若第 } k \text{ 隻螞蟻經過}(i, j) \\ 0 & \text{否則} \end{cases} \quad (2)$$

Q 表示費洛蒙強度，它在一定程度上影響算法的收斂速度， L_k 為第 k 隻螞蟻在本次循環中所走路徑的總長度。

2) 揮發(Evaporating)：

$$\tau_{ij}(t+1) = (1 - \sigma)\tau_{ij}(t) \quad (3)$$

σ 代表費洛蒙蒸散比例， $0 < \sigma < 1$ 。

結合上述機制，費洛蒙數學模型更新為：

$$\tau_{ij}(t+1) = (1 - \sigma)\tau_{ij}(t) + \Delta\tau_{ij} \quad (4)$$

為了不干擾螞蟻選擇路徑，初始費洛蒙通常會設定成一極小值，爾後隨時間逐漸增加，費洛蒙才開始有所變化。

Dorigo[11] 將 AS、GA 與 SA 等以 TSPLIB 國際例題進行比較，結果顯示 AS 表現較佳，其執行結果與實際最佳解誤差皆小於 3.5%。

2. 啟發式演算法中角度因子探討

AS 雖然容易處理路徑規劃的問題，但在最佳化求解的過程中容易陷入區域最佳解及迭代過程需要耗費較多時間[4、5]，其原因為 AS 依賴相鄰節點之距離及費洛蒙濃度決定，相鄰節點之距離越短，則該節點被選擇的機率越大(公式 1)。然而，僅僅依據相鄰節點的距離來決定路徑，在路徑規劃的情況下會因為前進方向的差異，而導致遠離終點。為解決此問題，本研究加上下一節點在前進方向上的投影向量做為啟發資訊，換句話說，

即在前進方向上的前進深度越大者被選擇的機率越大。如圖 2-1，今有向量 \vec{i} 及 \vec{j} 且 $\vec{i} \neq \vec{j}$ ，若 \vec{i} 與 \vec{j} 之夾角為 θ ，則 \vec{i} 在 \vec{j} 上之投影量 k 為公式(5)。

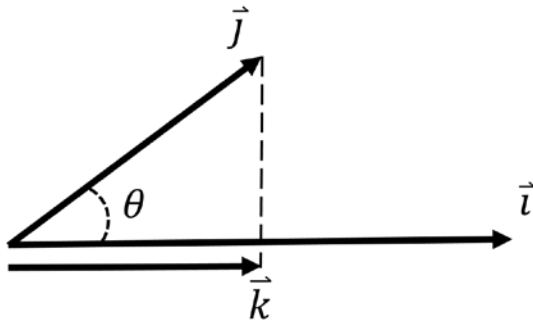


圖 2-1 向量投影

$$k = \vec{j} \cdot \vec{i} / \|\vec{i}\| = \|\vec{j}\| \cos\theta \quad (5)$$

由於 $\cos\theta$ 的值界於-1 到 1 之間，若僅僅是相乘則角度此項因素的影響程度不足，故經過修正後的啟發資訊參數的數學模型如下：

$$\eta_{ij} = (d_{ij})^{\gamma(\cos\theta_{ij}+1)} \quad (6)$$

- γ ：為角度影響啟發資訊的比重參數。
- θ_{ij} ：為 i 節點與 j 節點之夾角。
- d_{ij} ：為 i 節點與 j 節點之距離。

表 2-1 η_{AS} 及 η_{SACS} 比較

節點	與前進方向夾角	相鄰節點距離	η_{AS}	η_{SACS}
A(10, 7)	81.9°	12.2	0.081	1.051
B(13, 19)	11.56°	8.6	0.116	1.077
C(15, 28)	23.4°	14.87	0.067	1.094
D(23, 23)	63.59°	8.54	0.117	1.055
E(23, 10)	173.88°	5	0.2	1.000

本研究先將 γ 的值設為 1，且以下列例子說明本研究啟發資訊運作方式。假設有一個二維座標空間(如圖 2-2)，其中有 A、B、C、D、E、CP 及 EP 等點，CP 為機器人所在位置，EP 為終點，A、B、C、D 及 E 點均為此次掃描後所得之節點位置，虛線是此次任務前進的方向，實線則是可能前進方向。將各點座標位置帶入公式(6)及公式(1)後，可得到表 2-1 的結果。

從 η_{AS} 裡可看出 E 點的值為 0.2，均大於其他節點，故在 AS 裡被選上的機率較其他節點大，但是從圖 2-2 可以看出 E 點位於 EP 的反方向，若選

擇 E 點則意味著遠離終點。從 η_{SACS} 可得知最大值为 C 點 1.094，因其在前進方向上(虛線)之投影量最大，所以選擇 C 點較其他節點之機率大，從圖 3-2 中可以看出選擇 C 點較選擇 E 點合理。目前在 $\gamma = 1$ 時就有如此大的差異，倘若根據不同情境而調整 γ 參數，效能必然有更大的提升。

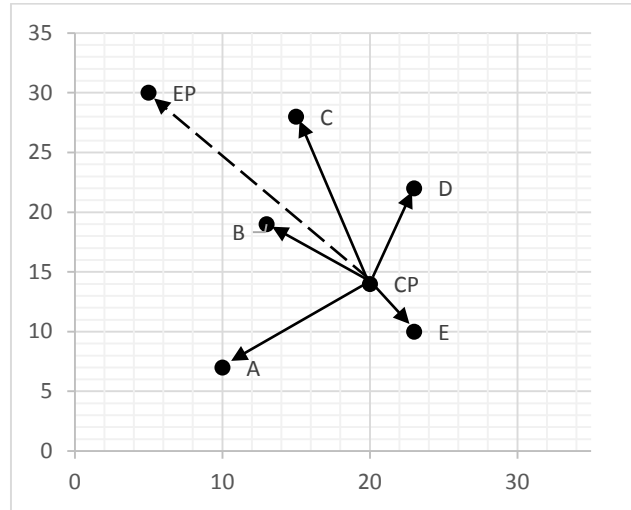


圖 2-2 機器人選擇節點圖

3. 實驗設計與效能分析

在現實環境中道路有可能因地貌的改變而產生移動、增加或消失，目標也可能因此而不同，故本研究所提出之生存力是指，當處於變化中的環境時，依然有能力可以找到路徑前進。而適應力是指能否找到較短的路徑前進，而不致陷入區域解。

本研究修改 Thomas Stützle. 在 2002 年開發之 ACSTSP-1.03[16]開放程式，擴充啟發資訊及角度並在動態地圖中檢視 SACS 是否能展現生存力，規劃出有效路徑，並與 AS 在路徑規劃的平均路徑長、有效性、時效性進行適應力的比較。

3.1 動態模型下 SACS 之表現

為驗證 SACS 在動態環境下，是否仍可規劃出可行路徑，設計實驗於輸入地圖參數時讀入兩個測試檔。實驗共跑 500 try，於 2 try 時進行地圖變換，藉此模擬動態環境。因實際情況中，感測器測量距離能力有限，故於實驗中設定兩節點距離超出 10 unit 者，判定為不可行走之路徑。

本研究實驗組共分為四個測試檔，test32_move.tsp、test32_add.tsp、test32_lose.tsp、test32_dif_obj.tsp，分別進行節點移動、節點增加、節點消失及目標變換等實驗，對照組則使用 test32.tsp。測試檔內各個節點均以平面座標表示

之，並標上 0~31 等編號以利識別。圖中兩節點間為一路徑，若有經過則以實心線段表示之，越粗則表示經過此路徑次數越多；越細則表示經過此路徑次數越少。

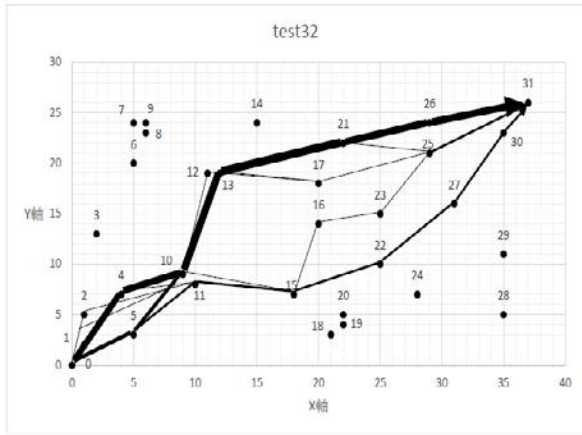


圖 3-1 SACS 於 test32 路徑行走紀錄

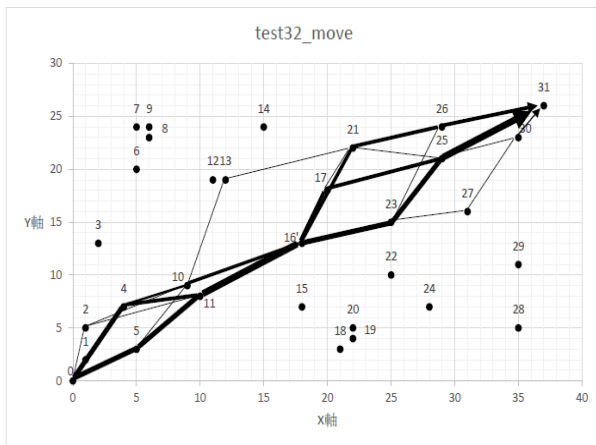


圖 3-2 SACS 於 test32_move 路徑行走紀錄

節點移動：

此實驗欲證實地圖中的節點若發生移動，則 SACS 演算法仍可正常運作。實驗中起點為節點 0(0,0)，終點為節點 31(37,26)，如圖 3-1 原路徑 10→16 長度為 12.8 unit、11→16 為 11.66 unit，兩者路徑均超出 10 unit。今將節點 16 (20,14) 移至 16' (18,13) 後如圖 3-2，路徑 10→16' 長度變為 9.84 unit、11→16' 為 9.43 unit，按照實驗設定均判定為可行路徑。

實驗結果，原地圖中行徑節點 10 及節點 11 時，都因為是不可行路徑而不經過節點 16，改走其他路徑。新地圖則因 16' 的位置是前進方向上的捷徑，使選擇節點 16' 的次數增加。

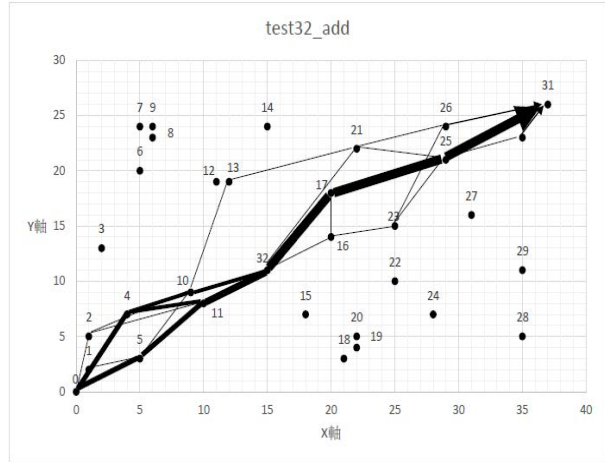


圖 3-3 SACS 於 test32_add 路徑行走紀錄

節點增加：

此實驗欲證明環境增加新節點是否仍可有效規劃路徑前進，故在圖 3-3 中節點 11 及節點 16 之間加入一新節點 32(15,11)。實驗中起點為節點 0(0,0)，終點為節點 31(37,26)，如圖 3-1 原路徑 10→16 長度為 12.8 unit、11→16 為 11.66 unit，兩者路徑均超出 10 uni，加入新節點後如圖 3-3，使 10→32→16、10→32→17、11→32→16 及 11→32→17 四條路徑長度均小於 10 unit。

實驗結果如圖 3-3 所示，由於經過節點 32 之總距離比經過節點 13 之總距離要短，故經過節點 32 的次數增加。

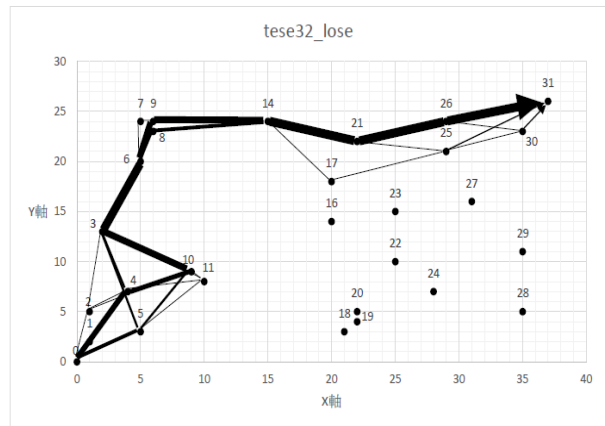


圖 3-4 SACS 於 test32 Lose 路徑行走紀錄

節點消失：

此實驗欲證明地圖中原本可行路徑消失，SACS 仍可規劃出路徑前進。如圖 3-1 原地圖中起點為節點 0(0,0)，終點為節點 31(37,26)，大部分經過的路徑為 0→4→10→13→21→26→31，今將節點 12、13 及 15 去掉如圖 3-4，則節點 11 右邊的

路徑長度都會超出 10 unit，原本可行之路徑也就消失。實驗結果顯示，原本路徑經過節點 13 及結點 15 因為節點的消失，故路徑改變為 0→4→10→3→6→9→14→21→26→31，繞過了不可行區域。

目標變換：

此實驗欲證明 SACS 若遇到目標變換之情況，仍可規劃出可行路徑。圖 3-1 中起點為節點 0(0,0)，終點為節點 31(37,26)，圖 3-5 則將終點改為節點 28(35,5)。實驗結果如圖 3-5 所示，由於終點變為節點 28，前進方向變換，故主要前進路徑變為 0→5→11→15→24→28，證明 SACS 即便在目標變換情況下，仍可有效規劃路徑。

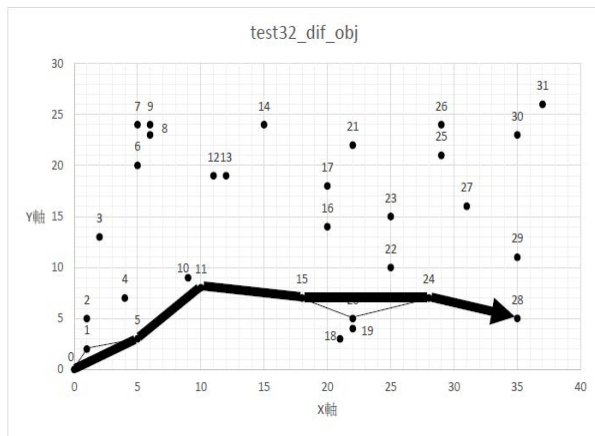


圖 3-5 SACS 於 test32_dif_obj 路徑行走紀錄

3.2 SACA 與 AS 之比較

為驗證 SACS 適應力較佳，本研究設計一個在相同的環境條件下的實驗，分別執行 SACS 及 AS 並分析其結果，輸入參數值如表 3-1：

表 3-1 實驗參數表(資料來源：[16])

Nmae	Value	Denotation
max_tries	10/500	number of independent trials
max_time	10	maximum time for each trial
tapfile	lin318.tsp	inputfile
n_ants	10	number of ants
alpha	1	alpha (influence of pheromone trails)
beta	2	beta (influence of heuristic information)
rho	0.5	pheromone trail evaporation

max_tries 為執行程式次數，實驗中執行 10 次

及執行 500 次的狀況；max_time 為執行 1 次程式的最大時間，實驗中設定為 10 秒；tapfile 為載入的測試檔，實驗載入測試檔為包含 318 個節點 lin318.tsp；n_ants 為執行搜索的螞蟻數量，設定 10 隻；alpha 為費洛蒙的比重值，實驗中設為 1；beta 為啟發資訊的比重值，實驗中設為 2。Rho 為費洛蒙的揮發值，實驗中設為 0.5。

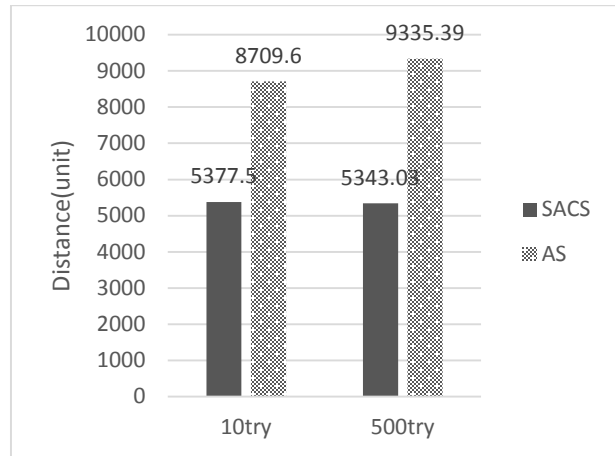


圖 3-6 SACS 及 AS 之平均路徑長

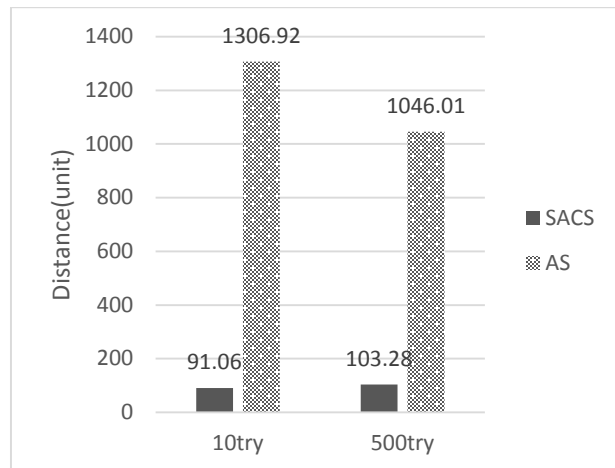


圖 3-7 SACS 及 AS 平均路徑長之標準差

平均路徑總長：

從圖 3-6、3-7 中顯示執行 10 次時 SACS 平均長度為 5377.5 unit，標準差為 91.06 unit；AS 平均長度為 8709.6 unit，標準差為 1306.92 unit。SACS 平均長度比 AS 縮短 38.3%，標準差低了 1215.86 unit。

執行 500 次時 SACS 平均長度為 5343.03 unit，標準差為 103.28 unit；AS 平均長度為 9335.39 unit，標準差為 1046.01 unit。SACS 平均長度比 AS 縮短 42.8%，標準差低了 942.73 unit。

實驗數據說明 SACS 所規劃的路徑比 AS 可以

更快抵達終點，且規劃之結果相當平均，長度較 AS 短。

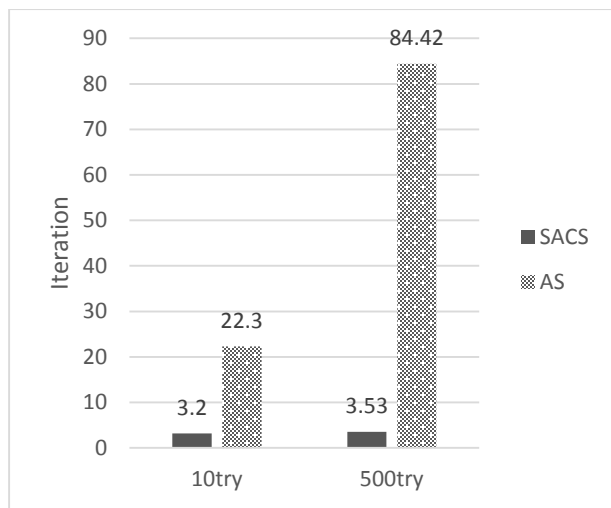


圖 3-8 SACS 及 AS 之平均迭代次數

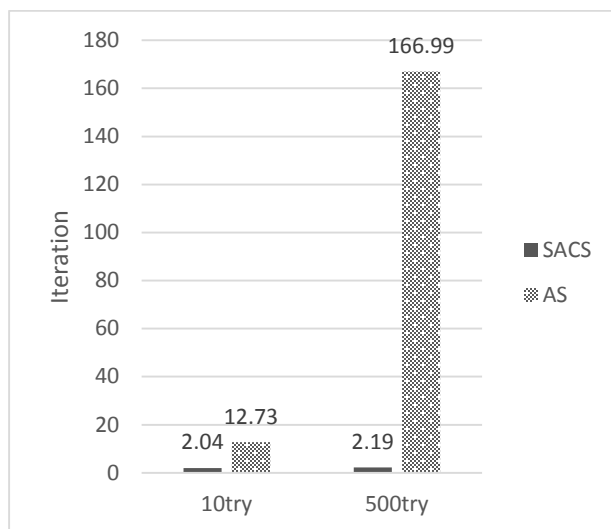


圖 3-9 SACS 及 AS 平均迭代次數之標準差

路徑規劃有效性：

從圖 3-8、3-9 中顯示執行 10 次時，SACS 找到最佳解的平均迭代次數為 3.2 次，標準差為 2.04 次；AS 平均迭代次數為 22.3 次，標準差為 12.73 次。SACS 找到最佳解的平均迭代次數比 AS 少 85.7%，標準差低了 10.69 次。

執行 500 次時，SACS 找到最佳解的平均迭代次數為 3.53 次，標準差為 2.19 次；AS 平均迭代次數為 84.42 次，標準差為 166.99 次。SACS 找到最佳解的平均迭代次數比 AS 少 95.9%，標準差低了 164.8 次。

說明 SACS 的規劃結果穩定，且路徑規劃較 AS 有效。

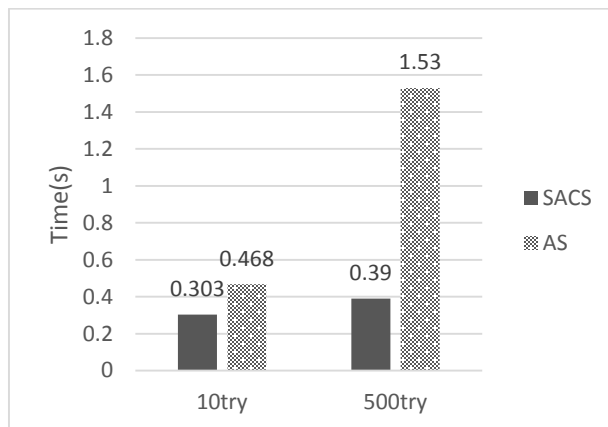


圖 3-10 SACS 及 AS 找最佳解之平均時間

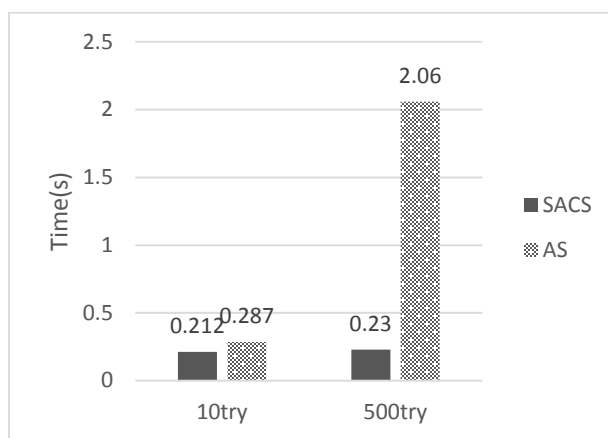


圖 3-11 SACS 及 AS 找最佳解平均時間之標準差

路徑規劃時效性：

從圖 3-10、3-11 可看出執行 10 次時 SACS 找到最佳解的平均時間為 0.303 秒，標準差為 0.212 秒；AS 平均時間為 0.468 秒，標準差為 0.287 秒。SACS 找到最佳解的平均時間比 AS 少 35.3%，標準差低了 0.057 秒。

執行 500 次時 SACS 找到最佳解的平均時間為 0.39 秒，標準差為 0.23 秒；AS 平均時間為 1.53 秒，標準差為 2.06 秒。SACS 找到最佳解的平均時間比 AS 少 74.6%，標準差低了 1.83 秒。

說明 SACS 收斂速度較佳，尤其在大量次數測試時更加明顯。

4. 結論與未來研究

本研究提出一種改良式螞蟻演算法，藉由改進啟發資訊參數，使機器人於現實環境中可以快速的規劃出一條可行路徑，實驗結果顯示，SACS 在路徑規劃的距離總長度、有效性及時效性，明顯較傳統 AS 演算法存活力更高、更具適應力。本

論文未來研究方向，將針對演算法在無預期變動環境及不同因素下參數所持有比重之效能進行探討，使其更趨於完備。

致謝

感謝科技部會對本研究之補助，計畫名稱「行動擴增實境之巨量資料環境下雲端計算啟發式跨層負載模型分析與優化之研究」，計畫編號：MOST 104-2221-E-606-001。

參考文獻

- [1].朴松昊、洪炳熔，”一種動態環境下一動機器人的路徑規劃方法”，*機器人*，第二十五卷，第一期，pp. 18-21，2003。
- [2].朱大奇、顏明重，”移動機器人路徑規劃技術綜述”，*控制與決策*，第二十五卷，第七期，pp. 961-967，2010。
- [3].李孟軒，”輪型機器人之路徑追蹤與避障”，*國立中央大學電機工程研究所碩士論文*，2011。
- [4].林志剛，”改良螞蟻最佳化演算法求解VRPTW問題”，*大同大學資訊經營研究所碩士論文*，2013。
- [5].賴智銘、郭躬德，”基於自適應閾值蟻群算法的路徑規劃算法”，*計算機系統應用*，第二十三卷，第二期，pp. 113-119，2014。
- [6].Avneesh Sud, Erik Andersen, Sean Curtis, Ming Lin, and Dinesh Manocha, “Real-time Path Planning in Dynamic Virtual Environment Using Multiagent Navigation Graphs”. *IEEE Trans on Visualization and Computer Graphics*, Vol. 14, No. 3, pp. 526-538, 2008.
- [7].Canny J. F., *The Complexity of Robot Motion Planning*. Boston, MIT Press, 1988.
- [8].Christian Hofner and Gunther Schmidt, “Path Planning and Guidance Techniques for an Autonomous Mobile Robot”. *IEEE Intelligent Robots and Systems*, Vol. 1, pp. 610-617, 1944.
- [9].Chaomin Luo, and Yang Simon X., “A Bioinspired Neural Network for Real-time Concurrent Map Building and Complete Coverage Robot Navigation in Unknown Environment”, *IEEE Transactions on Neural Networks*, Vol. 19, No. 7, pp. 1279-1298, 2008.
- [10].De Carvalho, R. N., Vidal, H. A., Vieira, P., and Ribeiro M.I., “Complete Coverage Path Planning and Guidance for Cleaning Robots”. *IEEE Int Conf Industry Electronics. Guimaraes*. Vol. 2, pp. 677-682, 1997.
- [11].Dorigo, M., Maniezzo, V., and Colorni, A., “The Ant System: Optimization by a Colony of Cooperating Agents”. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol. 26, No. 1, pp. 29-41, 1996.
- [12].Kikuo Fujimura, Hanan Samet, “A Hierarchical Strategy for Path Planning Among Moving Obstacles”. *IEEE Trans on Robotic and Automation*, Vol. 5, No. 1, pp. 61-69, 1989.
- [13].Najjaran, H., and Goldenberg, A., “Real-time Motion Planning of an Autonomous Mobile Manipulator Using a Fuzzy Adaptive Kalman Filter”. *Science Direct on Robotics and Autonomous Systems*, Vol. 55, No. 2, pp. 96-106, 2007.
- [14]. Purian, F.K., and Sadeghian, E., “Mobile robots path planning using ant colony optimization and Fuzzy Logic algorithms in unknown dynamic environments”. *IEEE International Conference on Control, Automation, Robotics and Embedded System*, 2013.
- [15].Robert A. Conn and Moshe Kam, “Robot Motion Planning on N-Dimensional Star Worlds Among Moving Obstacles”. *IEEE Trans on Robotic Automation*, Vol. 14, No. 2, pp. 320-325, 1998.
- [16].Romain Hendrickx and Leonardo Bezerra "About Ant Colony Optimization", *March 2012 (available online at <http://www.aco-metaheuristic.org/about.html>)*. [visited in 2015/09/07]
- [17].Subhadeep C., “Ant Colony System: A New Concept to Robot Path Planning”. *International Journal of Hybrid Information Technology*. Vol. 6, No. 6, pp. 11-30, 2013.
- [18].Schmidt, G., and Hofner, C., “An Advanced Planning and Navigation Approach for Autonomous Cleaning Robot Operation”. *IEEE International Conference on*

Intelligent Robots System. Vol. 2, pp. 1230-1235, 1998.

- [19].Takahashi, O., and Schilling, R. J., “Motion Planning in a Plane Using Generalized Voronoi Diagrams”. *IEEE Trans on Robotics and Automation*, Vol. 5, No. 2, pp. 143-150, 1989.
- [20].Vasudevan, C., and Ganesan, K., “Case-based Path Planning for Autonomous Underwater Vehicles”. *Proceedings of the 1994 IEEE International Symposium*, pp. 160-165, 1994.