

國立政治大學資訊科學系
Department of Computer Science
National Chengchi University

碩士論文

Master's Thesis

在機會網路上使用機率預測法搜尋行動代理人
之機制

Using Probabilistic Prediction Method in the Search of
Mobile Agents over Opportunistic Network

研究生：游筱慈

指導教授：連耀南

中華民國一百零二年五月

May 2013

在機會網路上使用機率預測法搜尋行動代理人
之機制

Using Probabilistic Prediction Method in the Search of
Mobile Agents over Opportunistic Network

研 究 生：游筱慈 Student：Hsiao-Tzu You

指 導 教 授：連耀南 Advisor：Yao-Nan Lien

國立政治大學
資訊科學系
碩士論文

A Thesis
submitted to Department of Computer Science
National Chengchi University
in partial fulfillment of the Requirements
for the degree of
Master
in
Computer Science

中華民國一百零二年五月

May 2013

在機會網路上使用機率預測法搜尋行動代理人之機制

摘要

在機會網路上，訊息的遞送遠比一般網路來得困難許多，溝通交換資訊效率很低。本篇論文以山文誌資訊系統為背景，假設在山區中已佈建完成控制節點並組成控制網路，以及行動代理人機制已導入在控制網路上用來搜尋移動的目標節點。其中行動代理人附屬於登山客所攜帶的設備上，欲搜尋的目標節點會沿著登山路徑不斷移動造成搜尋上的困難，若搜尋失敗不只拉長延後了搜尋時間，也可能錯失黃金救難時間造成極大的損失，如何增進搜尋效率是機會網路上相當重要的議題。為此，本文提出一個搜尋方法，在任意的時間點計算目標行動節點落在每個控制節點之間路段的機率，預測目標代理人的位置，就可依機率高低逐次搜尋各路段，以提高搜尋效率。我們以山文誌登山資訊系統，作為參考的機會網路，提出兩個模型，使用機率預測搜尋法，預測行動節點可能所在位置優先搜尋此路段來降低整體搜尋時間，透過一連串的實驗驗證機率模型之準確度，並評估本法之搜尋效率以及當各路段花費時間的機率分佈假設有誤時，搜尋效率的受損程度。在我們的實驗中，機率模型之準確度極高，誤差不超過 7.59%，搜尋效率都在 44.44 倍以上，即使機率分佈錯誤，搜尋效能仍高於二分搜尋法約 2 倍。

Using Probabilistic Prediction Method in the Search of Mobile Agents over Opportunistic Network

Abstract

Since transmitting data on an opportunistic network is more difficult than that on a general network, information exchanging is less efficient. Based on “CenWits” system, we assume that control point has entirely construed all over the mountains and a control network has completed altogether; meanwhile, the mobile agent mechanism has applied in the searching of mobile target nodes. With mobile agent attached on the equipment of hikers, the target agent moving constantly along the hiking path grows the difficulties in searching. The failure in locating the mobile agent possibly not only prolongs the searching time, but also misses the golden time of life saving, and causes enormously damages eventually. Therefore, figuring that “improving the efficiency of searching” is a major issue in opportunistic network, in this thesis we develop a searching method which enables us to calculate the probability where a mobile target agent locates in every edge between control points in any arbitrary time point. Through forecasting the location of the target agent, we can start searching from the edge with the highest probability, thus enhance the efficiency of searching. Using “CenWits” system as reference opportunistic network, we designed two probability models as well as associated search methods. We conducted a series of experiments to evaluate the accuracy of probabilistic models and the performance of the proposed search methods. In our experiments,

the error of probability models is no more than 7.59%. Our proposed methods outperform Basic Binary Search by 44.44 in average. Furthermore, assuming that there is a discrepancy between the probability assumptions and the real distribution of the traveling time spent on each edge, we evaluate the performance degradation too. The experimental results show that under such circumstance, our Probabilistic Prediction Method can even outperform Basic Binary Search by approximately 200%.

致謝辭

感謝連耀南教授帶領著我做研究，訓練學生獨立思考並詢詢善誘讓這份論文盡可能完善，在完稿的這一刻心中有著滿滿的成就感並對指導教授懷著感恩心情。做研究常有一種前不見古人，後不見來者的孤獨感，謝謝始終替我打氣，一起做研究、玩耍、打球的 MC 玩樂幫。更感謝讓我不需做家事，替我帶便當的家人，這段時間著實辛苦了，尤其是最近艱難的日子，我即將歸隊與你們患難與共，最好能榮華共享。親愛的父親願您身體康復，與母親完成不老騎士環島傳說。親愛的母親，感恩您始終樂觀，給我精神上最大的支持。最後將這份榮耀獻給我最親愛的家人、指導教授及親朋好友。

游筱慈 May 2013 政治大學

目錄

第 1 章	1
緒論	1
1.1 研究背景與目的	1
1.2 機會網路	2
1.2.1 機會網路的訊息交換機制	3
1.2.2 機會網路的研究議題	4
1.2.3 機會網路的應用	6
1.3 行動代理人	8
1.3.1 行動代理人的特點	9
1.3.2 行動代理人的優點	10
1.3.3 行動代理人的應用	11
1.4 運用行動代理人於機會網路	12
1.4.1 機會網路上的行動代理人運作	13
1.4.2 行動代理人平台的功能需求	13
1.5 論文組織架構	14
第 2 章 相關研究與文獻探討	16
2.1 機會網路上搜尋行動代理人之挑戰	16
2.2 「山文誌登山資訊系統」簡介	16
2.3 機會網路上使用控制網路搜尋行動代理人	20
2.3.1 控制網路概念	21
2.3.2 利用控制網路的搜尋策略	22
2.4 控制網路上控制節點選擇問題之相關研究	23

2.4.1	控制網路的建置	23
2.4.2	環境假設	24
2.4.3	設計考量與目標	24
2.4.4	控制點選擇問題—最大總流量模型	25
2.4.5	控制點選擇問題—最大涵蓋率模型	25
2.4.6	控制點選擇問題—最大加權涵蓋率模型	26
第 3 章 機會網路上行動代理人搜尋控制點選擇問題及解決方案		27
3.1	控制點選擇問題-最大複合加權涵蓋率模型	27
3.2	最大複合加權涵蓋率的數學模型	28
3.3	解決方案的演算法	29
3.3.1	控制點選擇問題的評估指標	30
3.3.2	CPSP-Coverage 改進式解決方案- NPF2 演算法	32
3.3.3	CPSP-Utility 改進式解決方案- NPF-U2 演算法	35
3.3.4	CPSP-Utility-Flow 解決方案- NPF-PF 演算法	38
第 4 章 機會網路上行動代理人位置之預測方法		40
4.1	預測方法設計理念與目標	40
4.2	問題定義	41
4.2.1	控制網路下搜尋名詞定義	41
4.2.2	問題模型	42
4.3	機率預測函數	44
4.3.1	Gamma 分配	45
4.3.2	模型一使用計算機率預測法之搜尋程序	47
4.3.3	模型二使用計算機率預測法之搜尋程序	49
第 5 章 效能評估		52

5.1	實驗設計	52
5.2	控制點選擇問題效能評估	54
5.2.1	小型問題效能評估	54
5.2.2	實驗 1A 環境參數設定	54
5.2.3	實驗 1A 結果與分析	55
5.2.4	大型問題效能評估	60
5.2.5	實驗 1B 環境參數設定	61
5.2.6	實驗 1B 結果與分析	61
5.2.7	實驗 1C 環境參數設定	64
5.2.8	實驗 1C 結果與分析	64
5.3	機率預測搜尋法效能評估	67
5.3.1	單一路徑機率預測準確度實驗	68
5.3.2	單一路徑機率預測搜尋法效能評估	74
5.3.3	單一路徑錯誤容忍度實驗	78
5.3.4	多重路徑機率預測準確度實驗	94
5.3.5	多重路徑機率預測搜尋法效能評估	97
5.3.6	多重路徑錯誤容忍度實驗	99
第 6 章	結論與未來研究	107
參考文獻	108

圖目錄

圖 1-1: Disconnected Mobile Ad Hoc Network (Disconnected MANET)	4
圖 2-1:山文誌登山系統-登山客黑盒子	19
圖 2-2:山文誌登山系統-基地台(簡易型與太陽能型)	20
圖 2-3:在機會網路上搜尋行動代理人 (以「山文誌登山資訊系統」應用情境為例)..	21
圖 2-4：使用控制網路來輔助行動代理人的搜尋.....	22
圖 3-1: Node Potential First 2 (NPF2) 演算法之運作流程	34
圖 3-2：Node Potential First-U2 (NPF-U2) 演算法之運作流程.....	37
圖 3-3：Node Potential First-PF (NPF-PF) 演算法之運作流程.....	39
圖 4-1:控制網路底下行動代理人的搜尋及其名詞定義	42
圖 4-2:問題模型一情境	43
圖 4-3:問題模型二情境	44
圖 4-4:Gamma 分配機率密度函數.....	45
圖 4-5: $P_{e,t}$ 之推導.....	47
圖 4-6:機率預測搜尋演算法-單一路徑(PPM-S).....	49
圖 4-7: $q_{e,t}$ 之推導	50
圖 4-8: 機率預測搜尋演算法-多重路徑(PPM-M).....	51
圖 5-1：CPSP-Coverage—NPF vs. NPF2 vs. Optimal solution (Edge Coverage).....	55
圖 5-2：CPSP-Coverage—NPF vs. NPF2 vs. Optimal solution (Mean MCD).....	55
圖 5-3：CPSP-Coverage—NPF vs. NPF2 vs. Optimal solution (Max MCD).....	56
圖 5-4：CPSP-Utility—NPF-U vs. NPF-U2 vs. Optimal solution (Weighted Edge Coverage).....	57

圖 5-5 : CPSP-Utility—NPF-U vs. NPF-U2 vs. Optimal solution (Mean Weighted MCD)	58
圖 5-6 : CPSP-Utility-Flow—NPF-PF vs. Optimal solution (Populated Edge Coverage)	59
圖 5-7 : CPSP-Utility-Flow—NPF-PF vs. Optimal solution (Mean Populated MCD) ...	59
圖 5-8 : 各 NPF 演算法之效能比較 (Edge Coverage) (Sensitivity of edge coverage on the number of nodes)	62
圖 5-9 : 各 NPF 演算法之效能比較 (Populated Edge Coverage) (Sensitivity of populated edge coverage on the number of nodes)	62
圖 5-10 : 各 NPF 演算法之效能比較 (Mean Weighted MCD) (Sensitivity of mean weighted MCD on the number of nodes)	63
圖 5-11 : 各 NPF 演算法之效能比較 (Mean Populated MCD) (Sensitivity of mean populated MCD on the number of nodes)	63
圖 5-12 : 各 NPF 演算法之效能比較 (Weighted Edge Coverage) (Sensitivity of weighted edge coverage on the number of trails)	65
圖 5-13 : 各 NPF 演算法之效能比較 (Populated Edge Coverage) (Sensitivity of populated edge coverage on the number of trails)	66
圖 5-14 : 各 NPF 演算法之效能比較 (Mean Weighted MCD) (Sensitivity of mean weighted MCD on the number of trails)	66
圖 5-15 : 各 NPF 演算法之效能比較 (Mean Populated MCD) (Sensitivity of mean populated MCD on the number of trails)	67
圖 5-16 : Gamma Distribution PDF (shape=2,scale=2)	69
圖 5-17 : Gamma Distribution PDF (shape=3,scale=2)	69
圖 5-18 : Gamma Distribution PDF(shape=4,scale=2)	70

圖 5-19 : Gamma Distribution PDF (shape=5,scale=2).....	70
圖 5-20:實驗 2A-1 出發後 10-100 分鐘 計算值與實驗值平均偏差(預測準確度實驗)	71
圖 5-21: 實驗 2A-1, 出發後 110-200 分鐘 計算值與實驗值平均偏差(預測準確度實驗)	72
圖 5-22: 實驗 2A-1, 出發後 210-300 分鐘 計算值與實驗值平均偏差(預測準確度實驗)	72
圖 5-23: 實驗 2A-1, 出發後 310-400 分鐘 計算值與實驗值平均偏差(預測準確度實驗)	73
圖 5-24: 實驗 2A-2, 出發後 10-100 分鐘 (搜尋法效能評估實驗).....	75
圖 5-25: 實驗 2A-2, 出發後 110-200 分鐘 (搜尋法效能評估實驗).....	75
圖 5-26: 實驗 2A-2, 出發後 210-300 分鐘 (搜尋法效能評估實驗).....	76
圖 5-27: 實驗 2A-2, 出發後 310-400 分鐘 (搜尋法效能評估實驗).....	76
圖 5-28: 2A 各實驗之平均偏差比較 出發後 10-100 分鐘.....	81
圖 5-29: 2A 各實驗之平均偏差比較 出發後 110-200 分鐘.....	81
圖 5-30: 2A 各實驗之平均偏差比較 出發後 210-300 分鐘.....	82
圖 5-31: 2A 各實驗之平均偏差比較 出發後 310-400 分鐘.....	82
圖 5-32: 實驗 2A-3-1, 出發後 10-100 分鐘 (錯誤容忍度實驗, Against Normal distribution).....	85
圖 5-33: 實驗 2A-3-1, 出發後 110-200 分鐘 (錯誤容忍度實驗, Against Normal distribution).....	85
圖 5-34: 實驗 2A-3-1, 出發後 210-300 分鐘 (錯誤容忍度實驗, Against Normal distribution).....	86
圖 5-35: 實驗 2A-3-1, 出發後 310-400 分鐘 (錯誤容忍度實驗, Against Normal distribution).....	86

圖 5-36: 實驗 2A-3-2, 出發後 10-100 分鐘 (錯誤容忍度實驗, Against inaccurate Gamma distribution).....	88
圖 5-37: 實驗 2A-3-2, 出發後 110-200 分鐘 (錯誤容忍度實驗, Against inaccurate Gamma distribution).....	88
圖 5-38: 實驗 2A-3-2, 出發後 210-300 分鐘 (錯誤容忍度實驗, Against inaccurate Gamma distribution).....	89
圖 5-39: 實驗 2A-3-2, 出發後 310-400 分鐘 (錯誤容忍度實驗, Against inaccurate Gamma distribution).....	89
圖 5-40: 實驗 2A, 出發後 10-100 分鐘 (錯誤容忍度實驗, Search Ratio).....	91
圖 5-41: 實驗 2A, 出發後 110-200 分鐘 (錯誤容忍度實驗, Search Ratio).....	91
圖 5-42: 實驗 2A, 出發後 210-300 分鐘 (錯誤容忍度實驗, Search Ratio).....	92
圖 5-43: 實驗 2A, 出發後 310-400 分鐘 (錯誤容忍度實驗, Search Ratio).....	92
圖 5-44: 實驗 2B-1, 出發後 10-100 分鐘 (預測準確度實驗).....	95
圖 5-45: 實驗 2B-1, 出發後 110-200 分鐘 (預測準確度實驗).....	96
圖 5-46: 實驗 2B-1, 出發後 210-300 分鐘 (預測準確度實驗).....	96
圖 5-47: 2B 各實驗之平均偏差比較 出發後 10-100 分鐘.....	101
圖 5-48: 2B 各實驗之平均偏差比較 出發後 110-200 分鐘.....	102
圖 5-49: 2B 各實驗之平均偏差比較 出發後 210-300 分鐘.....	102
圖 5-50: 實驗 2B, 出發後 10-100 分鐘 (錯誤容忍度實驗).....	105
圖 5-51: 實驗 2B, 出發後 110-200 分鐘 (錯誤容忍度實驗).....	105
圖 5-52: 實驗 2B, 出發後 210-300 分鐘 (錯誤容忍度實驗).....	106

表目錄

表 3-1：控制點選擇問題 (CPSP) 的評估指標.....	30
表 3-2:Node Potential First (NPF)及 Node Potential First 2 (NPF2) 演算法參數表.....	32
表 3-3: NPF、NPF2 演算法比較.....	35
表 3-4：Node Potential First U (NPF-U)及 Node Potential First U2 (NPF-U2)演算法參數表.....	36
表 3-5:NPF-U、NPF-U2 演算法比較.....	37
表 3-6:NPF-PF 演算法之參數表.....	38
表 4-1:名詞定義.....	41
表 5-1：實驗一控制點選擇問題實驗描述.....	53
表 5-2：實驗二機率預測搜尋法實驗描述.....	53
表 5-3：實驗 2A-1 參數.....	68
表 5-4:實驗 2A-1 預測準確度實驗.....	73
表 5-5：實驗 2A-2 參數.....	74
表 5-6 實驗 2A-2 搜尋法效能評估實驗.....	77
表 5-7：實驗 2A-3-1 參數.....	78
表 5-8：實驗 2A-3-2 參數.....	78
表 5-9: 實驗 2A-3-1 預測準確度實驗.....	83
表 5-10:實驗 2A-3-2 預測準確度實驗.....	84
表 5-11: PPM-S 錯誤容忍度實驗(Against Normal distribution).....	87
表 5-12: PPM-S 錯誤容忍度實驗(Against inaccurate Gamma distribution).....	90
表 5-13: PPM-S 錯誤容忍度實驗(Search Ratio).....	93
表 5-14:實驗 2B-1 參數.....	94

表 5-15:實驗 2B-1 預測準確度實驗	97
表 5-16: 實驗 2B-2 參數.....	97
表 5-17 實驗 2B-2 搜尋法效能評估實驗.....	98
表 5-18:實驗 2B-3-1 參數	99
表 5-19:實驗 2B-3-2 參數	99
表 5-20:實驗 2B-3-1 預測準確度實驗	103
表 5-21:實驗 2B-3-2 預測準確度實驗	103
表 5-22: PPM-M 錯誤容忍度實驗(Against Normal distribution).....	104
表 5-23: PPM-M 錯誤容忍度實驗(Against inaccurate Gamma distribution).....	104

第1章

緒論

1.1 研究背景與目的

旅行者時間位置的預測在許多研究領域中都是個重要的議題，例如：交通運輸、物流管理、車載網路、地理資訊系統...等。以旅行者本身的觀點來看，旅行者可依照預測的旅行時間評估到達目的地所需花費的時間。以管理者而言，預測目標旅行者在某個時間點所在位置對於掌控旅行者行蹤及安全性有極大的幫助。本篇的研究背景在山區機會網路環境上，機會網路傳遞資料遠比一般網路困難，機會網路本身的特性使得傳遞訊息的效率很低，因此我們將在機會網路上使用行動代理人來傳遞溝通，希望可利用行動代理人自主性的性質，增加訊息交換的效率。而我們要對一個行動代理人下達指令，無論是暫停、終止、恢復，最重要的第一件事，就是要能夠先找到這一個目標行動代理人，因此我們提出一個預測及搜尋方法，來預測此目標節點在某個給定的時間其所在山區路段各段機率並依機率高低依序搜尋，期望提升訊息溝通的效率或是進一步支援緊急山難搜救。

為此，本篇論文以「山文誌登山資訊系統 (Cen Wits System)」為背景，在登山客追蹤與環境監控系統中，每個登山客都攜帶一個配備有 GPS 功能的小型設備，在登山客的移動過程中，自動收集位置資訊，並使用短程無線電來和鄰近的登山客交換彼此的移動資訊。本研究假設山區已佈建完成內部能快速交換資訊的高速網路（稱為

控制網路)，並假設已導入行動代理人技術於機會網路中。行動代理人附屬於登山客所攜帶的設備上，當兩個登山客互相接近時，行動代理人才可從其中一個設備轉移到另一個設備，故其行動力受制於登山客的移動行為。因此，本研究提出一個預測方法，利用機率計算的方式來預測目標代理人在山區健行路徑上各路徑之機率，接著提出一個簡單的搜尋演算法，將搜尋代理人利用控制網路發送至預測路段中機率最高路段前方的控制節點，再利用反向的行動節點來逐步靠近目標節點。讓行動代理人藉此控制網路的轉送，以快速接近並搜尋到特定的行動代理人。

1.2 機會網路

機會網路 (Opportunistic Network, OppNet) [1]是由許多移動式節點 (mobile node, MN) 所組成，在機會網路環境中，可能因移動式節點的分佈位置不均、移動性等原因，造成整個網路分成幾塊互不相連的區塊。

以下為機會網路的特性。

- 網路聯繫是斷斷續續的。
- 可能不存在來源端到目的端點對點的路徑 (end-to-end path)。
- 兩端點之間連線中斷與重新連線的情況可能時常發生。

由於以上幾個特性，使得原本的資料傳輸機制，無法直接使用在機會網路中，因此產生了一個新的訊息交換機制—store-carry-forward，藉由這個訊息交換機制，可讓不存在點對點路徑的兩個節點，藉由中間節點代為轉送，使訊息透過接力的方式傳送到目的端，當沒有機會使資料往目的端發送時，中間代傳的節點會為來源端暫存這份資料，等待適合的代傳節點出現，再將資料透過接力方式往目的端發送。機會網路的資料傳輸沒有一個固定的路線，即使是位於不同網路區塊的節點，也有機會透過中間節點的代傳，得以互相通訊。

在機會網路中，藉由節點本身的移動性，來讓不同的節點之間有「機會」互相接觸，以傳送、接收或代傳資料，但也因為此特性，使得在資料傳輸的過程中，可能因為環境因素、節點的移動性等原因，而造成移動式節點之間產生間歇性、臨機性的接觸與連結，這將造成每一次的封包傳遞常會有延遲的狀況，所花費的傳輸時間也難以預測，因此，機會網路是屬於耐延遲網路 (Delay Tolerant Networks, DTN) 的一種，不適合使用在即時性需求高的服務，例如：網路電話 (Voice over IP, VoIP)。

1.2.1 機會網路的訊息交換機制

運作在傳統網路下的資料交換機制，一般都假設在來源端與目的端之間存在有點對點的路徑 (end-to-end path)，然而在機會網路中，不一定存在這條點對點的路徑，因此，store-carry-forward 機制被提出作為機會網路上的訊息交換機制。

利用 store-carry-forward 訊息交換機制，節點需依靠本身所具備的移動性，且節點扮演的角色必需能隨需求適時地做切換。每個節點都身兼數職，可以承擔 host、router、或 gateway 的任務，當它作為 host 時，具有儲存及保管資料的能力，作為 router 時，此節點便同時具有儲存、攜帶、轉送資料的能力，可以做為同一個網路區塊中其他節點的代傳節點，而作為 gateway 時，則可以視為不同網路區塊之間的代傳節點，如此一來，即使各節點之間處於互不相連的網路區塊，也能藉此機制順利溝通，這種以 store-carry-forward 技術來傳送資料的方式，是機會網路資料傳輸的最大特色。

機會網路可能的應用環境，可以以一個分離的無線隨意網路 (disconnected Mobile Ad Hoc Network, disconnected MANET) [2]來說明，如所示，該網路由許多移動式節點所組成，當整個網路因節點的移動性，而被切成許多大大小小、互不相連的網路區塊時，由於來源端和目的端之間不存在點對點的路徑，故位於不同區塊的節點是無法互相溝通的，這樣的網路型態即被稱為分離的無線隨意網路。

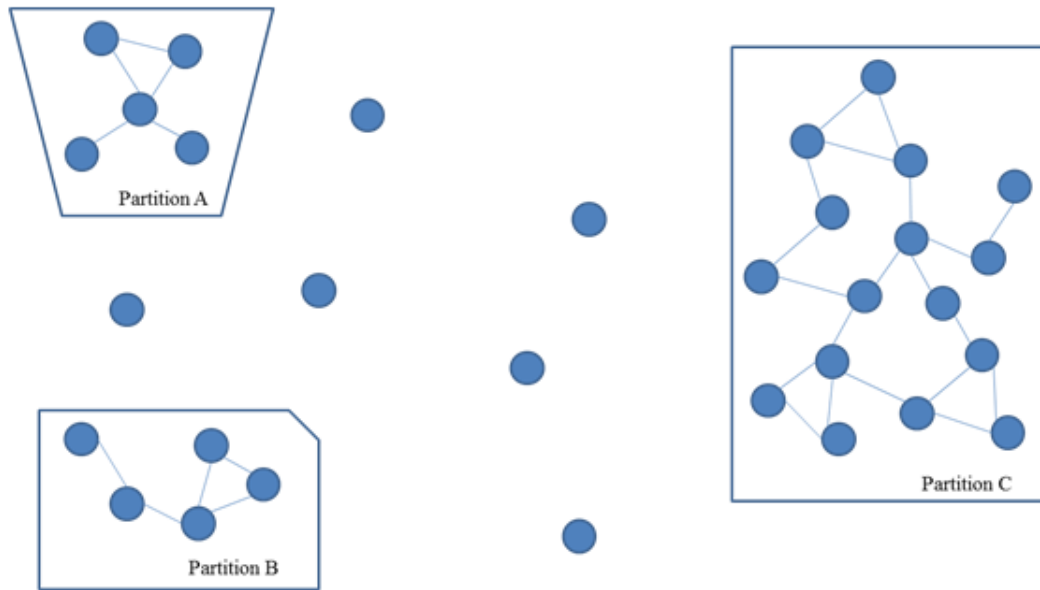


圖 1-1: Disconnected Mobile Ad Hoc Network (Disconnected MANET)

在此分離的無線隨意網路中，若可以利用移動式節點的移動性及區域發送訊息 (local forwarding) 的能力，讓訊息藉由其他節點的代傳來傳送，那就可以達到資料交換的目的。

1.2.2 機會網路的研究議題

由於機會網路本身的特性或限制，使得在資料交換方面的研究，面臨了許多挑戰，以下分別依照不同的網路層級來說明[3]。

1. 網路層

機會網路上的資料交換需依靠移動式節點之間的接觸，且節點同時擁有 host 及 relay 的身份時，整個機會網路才能運作得起來。在網路層中，由於每一節點皆肩負轉送任務，故將額外消耗儲存空間，因此，節點本身的儲存空間大小必需被列入考慮，基於此，產生了兩種完全不同的路由設計理念，分別是 flooding-based (multiple copies) approach 及 forwarding-based (single copy) approach，分別說明如下。

- Flooding-based (multiple copies) approach :

發送端或中間節點在發出或轉送一個訊息時，會將同一個訊息的多個複本，透過不同的路由路徑來轉送給鄰近節點，直到將訊息送達目的端為止。由於訊息是循著許多不同路徑轉送，較有機會使用到較短的路程，可以降低資料傳送的延遲率，並提高資料傳輸的成功率，但也會造成網路上充斥著同一個訊息的多個複本，耗費轉送節點大量的儲存空間，且佔用較多資料傳輸所需的網路頻寬。

- Forwarding-based (single copy) approach :

發送端或中間節點在發出或轉送一個訊息時，僅轉送到一個鄰近節點，重覆資料轉送的動作，直到將訊息送達目的端為止。這種作法可以減少儲存空間的使用量，並避免網路上相同訊息的過度泛濫，然而，對同一個訊息而言，由於同時間只有一個負責保管此訊息的代傳節點，而訊息轉送路徑無法保證為最短路徑，故也會造成延遲時間加長及資料傳遞效率偏低的情形。

2. 傳輸層

在機會網路中，節點之間發生連線中斷的狀況是相當頻繁的，且點對點路徑也極少是存在的，故使用在傳統網路上的傳輸層協定（例如：TCP），由於在設計之初，都是假設在點對點路徑已存在的環境下，因此，無法直接轉移到機會網路上使用。

3. 包裹層 (Bundle Layer)

這一層是為了支援 store-carry-forward 資料傳輸機制而產生，主要負責資料包裹（或稱為 bundles）的儲存、攜帶及轉送，當只有一個目的端節點時，需提供 unicast bundle delivery 的技術，當有多個目的端節點時，則需提供 multicast、anycast delivery 的方式來傳輸資料。

4. 應用層

在機會網路中，由於無法預期節點之間的連線何時會被中斷，故不容易妥善地利

用節點接觸時所產生的通訊機會，因此，將傳統網路上的應用直接移植到機會網路中，仍頗具挑戰性。

即使是本質上具有耐延遲特性的網路應用，在傾向於分離的網路環境中，整體的效能也會顯著地下降，舉電子郵件這個高度耐延遲的網路應用來說，雖然使用者通常習慣為了一封回覆去等數小時或數天的時間，然而，由於現有支援電子郵件應用的傳輸層協定（例如：TCP），在設計之初，並沒有被考慮使用在機會網路這樣的環境中，因此，在機會網路中支援電子郵件的應用，仍是相當具有挑戰性的。

即使如此，在機會網路中支援電子郵件的應用，仍是相對簡單的，因為它耐延遲的特性，可以忍受機會網路的高延遲傳輸特性，然而，支援 Web 應用則要更複雜得多，因為高互動性的應用協定（例如：HTTP），不能忍受高延遲。

除了以上所探討的幾點外，另有兩個在機會網路中不可忽略的特殊挑戰。

- Contact – 應該要考慮當兩個移動式節點互相接觸時，可以成功傳送的資料量多寡。
- Storage constraint – 應該要考慮在移動式節點中，是否有足夠的儲存空間，用來暫存準備代傳的資料。

在機會網路中，即使存在了 store-carry-forward 訊息交換機制讓節點之間能順利交換資料，然而，由於與生俱來的網路連線特性，使得資訊交換的傳輸率及延遲時間難以估計，並且由於每個節點內部儲存容量的限制，故也可能造成資料未成功送出前就已被丟掉，因此，對保證資料成功交換的挑戰性極大。

1.2.3 機會網路的應用

1. 登山服務與山難搜救

由中研院、台灣大學與美國科羅拉多大學合作研發出的「山文誌登山資訊系統

(CenWits System)」[3-5]，主要目的為收集遊客行走路徑與山區的氣候資訊，並將所收集到的資料做進一步的分析、處理與應用，現已實際應用於玉山國家公園，稱為玉山網 (YushanNet) [6, 7]。

將「山文誌登山資訊系統」應用於山難搜救時，可先經由控制中心所搜集到的資訊，來定位待援登山客的行蹤，並將搜救範圍縮小至最有效的區域，以協助搜救人員抓緊黃金救援時間，提升救援速度，並節省搜尋救援工作所耗費的人力與物力成本。

而關於登山服務方面，也可讓國家公園管理處得知遊客活動較頻繁的區域，作為日後增修遊客服務設施的參考，另外，登山客的家屬也可以透過網際網路，來掌握登山客目前的位置與行進狀況。

2. 野生動物追蹤

由於野生動物沒有固定路徑，故只能在特定地點找到牠們，例如：水源、巢穴等，在這些野生動物的身上裝置省電、無須額外設定的 GPS 收發器後，研究人員便可藉由裝置之間資料交換的所得，來分析出野生動物的活動範圍，以及不同物種間的互動及遷徙等關係，相關的研究如美國普林斯頓大學的 ZebraNet Project[8,9]。

3. 外太空傳輸

美國國家航空暨太空總署 (NASA) 在 2011 年啟用的 InterPlanetary Interent Project 就是這個應用的實際案例[10,11]。

網際網路 (Internet) 使用 TCP/IP 協定來保證網路中資料傳輸的可靠度，在這個協定中，傳輸資料的來源端和目的端必須在傳輸的過程中保持連線的建立，然而，這在外太空傳輸的過程中是很難做到的，因為距離太遠，再加上行星的運行及衛星的漂浮等情況，故在傳輸過程中，連線的間歇性中斷情形是很普遍的，因此，在這樣的環境下，就相當適合使用機會網路的傳輸特性，當連線中斷時，可以繼續暫存資料，直到下一個可能的節點再傳輸，並且，容許傳輸時間的長時間延遲及高資料錯誤率。

4. 開發中國家或偏遠地區的網路傳輸

以往許多技術都是為了已開發國家而研究、發展，然而，這些技術可能無法使用在開發中國家或偏遠地區，因為這些地區的基礎建設可能並不完善，甚至該地的人們可能也沒有一定程度的知識水平，在這樣的應用環境下，成本、部署 (deployment)、功耗等問題就需要被考慮，例如：美國柏克萊大學所研究的 Tier Project[12]，主要的研究目標就是要解決這些開發中國家或偏遠地區的網路通信問題。

5. 戰區、災區的通訊網路

以往 Mobile Ad Hoc Network (MANET) 的研究應用，常被設定在無建置基礎建設 (infrastructure)，或基礎建設已遭受損壞的戰區、災區環境中，然而，在這樣的環境中，由移動式節點所組成的許多互不相連的網路區塊，可能會因為移動式節點的移動性與無線傳輸範圍的限制，而造成不同的網路區塊之間無法互相通訊的情況，此時，若能利用節點的移動性，以及機會網路最重要的傳輸機制—store-carry-forward 訊息交換機制，就能適用於戰區、災區環境[13,14]。

1.3 行動代理人

所謂的行動代理人 (mobile agent) 可以視為一支可在異質系統之間移動、執行，並具有特定功能的小型程式。當原本的節點 (home node) 送出行動代理人時，代理人就像使用者的分身一樣，從原本的節點中分離出來，自主地決定要訪問的節點，無須和 home node 持續保持連絡，它將代表使用者在網路中的節點進行多次跳躍 (multiple hops) 移動與執行任務，最後將使用者所交待的任務結果回傳到 home node。

行動代理人其實也是代理人技術 (agent technology) 的一種。其基本原理就是當行動代理人 (mobile agent) 接收到終端使用者的請求命令時，便即時地處理使用者所交付的任務，並依實際的情況，可從網路一端的主機移動至其他另一端的主機去處理，最後再將結果回傳給使用者。實際上，在網路上移動的行動代理人，其狀態不僅包含本身的程式碼和任務處理資料，還包括了 thread 的狀態及其擁有者的權限。

另外，在終端間移動中的行動代理人，具備了自主性與機動性，即使是在無法和使用者相互通信的環境中，也能將移動前的程式憑藉自己本身的知識，或所收集到的情報，自主地加以分析判斷，並繼續執行處理。

1.3.1 行動代理人的特點

代理人通常扮演一個使用者或另一個代理人的角色，並負責和其他代理人之間的溝通 (communication) 與合作 (collaboration/cooperation)。以下為「行動代理人」所必備的特質[15]：

- 自適應性 (Adaptivity)：代理人具備學習能力並能隨經驗增長
- 自主性 (Autonomy)：代理人必須具有主動性，在朝達成目標的方向下，有自主能力採取啟發性之行為
- 協作行為 (Collaborative behavior)：代理人具備可以與其它代理人合作並一起達成目標之能力與行為
- 推論能力 (Inferential capability)：代理人具備有針對抽象定義之目標採取行動之能力，換句話說就是具備推論能力
- 知識水平的通訊能力 (Knowledge-level communication ability)：代理人在與其它代理人溝通時，具備使用人性化語言與溝通方式之能力
- 個性化 (Personality)：代理人可以具有個性化之特質，即具有可以扮演出逼真行為之能力
- 反應性 (Reactivity)：代理人具備選擇性感測與反應之能力
- 持續性 (Temporal continuity)：代理人具備保持身份扮演一致性之特質，並能夠長時間的維持
- 移動性 (Mobility)：代理人具備機動引導之能力，可以從一個主機移至另一個。

1.3.2 行動代理人的優點

使用行動代理人技術的好處，包括克服客戶端計算機能力的限制、客制化、與生俱來的存活能力、作為離線用戶的代表，和易於擴展。以下將逐一說明[16]。

- 克服客戶端計算機能力的限制 (Overcoming the limitations of a client computer)：舉例來說，先把代理人送到遠端主機的資料庫，並根據自定義的演算法進行搜尋後，再將資料回傳客戶端，比直接存取遠端資料庫更有效率，並能克服客戶端本身的通訊延遲、總處理能力、儲存容量大小、計算能力等限制，此外，客戶端也可能沒有足夠的儲存空間來暫存大量資料，或者可能沒有足夠的網路頻寬和計算能力。
- 客制化 (customization)：在 client/server 架構中，是很難適應快速變化的，相反的，行動代理人則可以很容易地依照使用者需求客制化，再將此新的需求送到伺服器端執行。其中，伺服器端的代理人系統代表一種解譯器 (interpreter)，負責接受並執行收到的代理人。
- 與生俱來的存活能力 (inherent survivability)：因為行動代理人能攜帶著本身的程式碼和狀態移動，故相較於 client/server 架構有更高的存活率，當一個節點或網路的一部分發生問題時，行動代理人可以離開這個節點，到其他地方再繼續執行原有的任務，雖然有其他方式可以提高 client/server 架構的執行效率，但仍無法適用於大規模的系統架構中，反之，行動代理人原本就是被設計用於 Web 上。
- 作為離線用戶的代表 (representation of a disconnected user)：現今很多使用者都具有移動性，時常在辦公室開始一個工作，而後在另一個地方利用筆記型電腦來繼續之前的工作，並希望在使用者離線的期間仍能繼續執行工作，故可利用送出行動代理人，讓使用者離線後，仍由代理人繼續處理原來的工作，

而後，當使用者再度上線時，再將代理人拉回。

- 易於擴展 (ease of development)：當大量的資料被儲存在分散的伺服器中，遠端存取所需的資料並非一個可行的選擇，甚至基於安全因素，本地資料可能不提供遠端存取，因此，可利用管線技術 (pipelining) 讓不同類型的代理人先後拜訪這些伺服器，並執行必要的動作。

1.3.3 行動代理人的應用

行動代理人在接收使用者的任務後，可以提供資訊處理之服務，包含資料的收集、檢索、過濾、交換等功能，在[17]中已提出幾個原因來說明行動代理人在許多應用中，是一個有效率的選擇，其優點有改善 client/server 架構中的頻寬限制、網路延遲，以及減少網路負載、網路斷線的弱點等。以下就現存的行動代理人的應用逐一說明 [17-19]。

1. 電子商務 (Electronic commerce)：使用者可告知行動代理人欲購買的商品名稱、價格、品牌等資訊，並委託其在虛擬商店中，代使用者做搜尋、過濾、下訂單等動作。
2. 軟體發行 (Software distribution)：

當機器的數量不斷地增加，在其上安裝和維護軟體變得更困難，可利用行動代理人依照軟體包的軟件清單，並檢查本機安裝版本，來自定義安裝，以自動完成此動作。

3. 資訊檢索 (Information retrieval)：

使用者僅需輸入欲獲取資料的關鍵字，便可藉由行動代理人的移動，來檢索分散在網路上各主機中資料庫的資料，行動代理人便會在目的端主機資料庫，過濾出所需的資料，並依照檢索結果來選擇移到其他主機繼續處理，或回傳結果給使用者。

4. 系統管理 (System administration) :

行動代理人可以獨立地在大型系統中移動，並執行定期性的管理，例如：備份或其他系統相關的操作，並可在問題發生時，於人為操作的干預前做初步的問題評估，以幫助系統管理員釐清問題發生的原因及後果，此外，也可用於自動執行常規任務，及定期檢查系統和網路。

5. 網路管理 (Network management) :

網路本質上是分散式的，且提供一個適合部署代理人的環境，例如：在主動式網路 (Active Network) 中，可透過行動代理人來移動、傳送通信協定的處理程式，並安裝在原本沒有此通信協定的主機上，使其主機得以運作，即無須事先在各主機上做各通信協定標準化的動作。

6. 網路安全 (Network security) :

將非公開機密化的資訊安置於行動代理人，而後予以加密，並隨著行動代理人在網路中移動，可免於在傳送中遭受竊取、竄改的顧慮，特別是能將解密程式也置於代理人中，因此，在目的地端主機無須事先準備解密程式，則我們可以採用各種雙方已協調好的任意加密形式來作加密動作。

近年來，行動通信和無線網路技術的大幅躍進，使得行動代理人的利用不僅限於遠端資料查詢，還包括了分散式即時系統 (Real Time System) 和工作流程系統 (workflow system) 等環境，促成了其應用的多樣化。

1.4 運用行動代理人於機會網路

由於在機會網路上，訊息在傳遞的過程中，相較於傳統網路有更多的不確定性、不可預測性，故訊息的傳遞速度較慢，在這樣的環境中，若採用傳統的集中控制模式並不恰當，因此，我們希望將行動代理人技術導入機會網路中，提升機會網路的功能，以期產生更多可行的應用。例如：利用行動代理人去尋找其他行動代理人，遠比由中控

中心去找行動代理人要來得方便、快速，也無須耗費許多無謂的資訊來回時間。

我們以「山文誌登山資訊系統」為例，在此系統上導入行動代理人技術，可擴大此系統原有的功能，以下列舉幾個可能的應用情境：

- 變更既有登山行程：不同日期出發的登山客，能在發生臨時事件而改變行程時，通知對方新的集合時間及地點。例如：原本約定在兩天後於某定點集合，卻臨時得知有颱風警報或山中起濃霧，此時便可利用行動代理人通知所有人，取消原本的計畫，並馬上趕到另一個地點集合。
- 互動通訊：當某個登山客發現隊友脫隊時，可以先傳送訊息給走失的隊友，並提供初步的協助，或約定下一個集合地點。

1.4.1 機會網路上的行動代理人運作

行動代理人於機會網路中的功能與特性如下。

- 存在於移動式節點上，並可進行決策運算。例如：資料匯整，資料搜尋，變換路徑等。
- 可以決定何時轉移到其他特定的移動式節點。
- 可以決定何時啟動、暫停、恢復自己的工作。
- 當轉移到其他的移動式節點時，可被視為一個在節點之間交換的訊息。

1.4.2 行動代理人平台的功能需求

雖然將行動代理人技術應用於機會網路中，將帶來許多彈性與優勢，但不可否認地，由於行動代理人的本質仍是一支程式，而只要是軟體就可能有程式撰寫上的缺陷存在，導致破壞系統正常運作，因此，當某些代理人發生執行期的錯誤，而造成系統資源的損耗，影響系統之運作，或其他危及系統的情況（例如：系統當機）時，行動代理人平台必須能在不危及系統安全及正常運作的前提下，去解決這些問題。

行動代理人平台，具有創建、解譯、執行、複製、轉移、終止行動代理人的能力，

且必須提供代理人的控制機制，來讓其擁有者能監控代理人的執行狀態、增加新的任務或撤回舊任務等，同時，行動代理人之間也必須能在任務中進行溝通合作，此外，代理人的控制機制也作為處理代理人發生錯誤，或在其他例外處理時使用。

要實現行動代理人的控制機制，首先，必須存在代理人管理員 (Agent administrator) 元件，此元件具備代理人行為之控制權限，可提供代理人的狀態查詢及其他控制代理人行為之指令。

以上所提到的代理人控制功能，至少需具備以下幾項：

- Status inquiry：代理人狀態查詢功能，用來查詢代理人唯一的識別碼、來源及目的地節點之資訊、被分派執行的命令名稱、優先權值、佔用的系統資源多寡、已耗費的執行時間與完成度，及目前的執行狀態、中間結果等資訊。
- Stop：中止當前任務。
- Suspend：暫停當前執行的任務。
- Resume：恢復之前所執行的任務。
- Update assignment：分派其他新任務。
- Change its itinerary：改變代理人原來的移動路線規劃。
- Kill：強制刪除當前代理人及其任務。

以上所列出的幾點控制機制，最大的困難與前提，即是在於「如何找到所欲控制的行動代理人」，只有先找到某個特定的行動代理人，方能對其進行控制。

1.5 論文組織架構

本篇論文章節架構如下：第一章說明本研究動機與目的，並介紹背景知識及架構。第二章討論搜尋行動代理人之研究及所面臨的挑戰，介紹相關研究所提出的控制網路概念及其基於「山文誌登山資訊系統」為背景的控制網路之建置模型。第三章，本文加入第四個控制網路的模型，並針對控制網路建置模型相關研究所提出的啟發式演算法

另提出改進之解決方案。在第四章提出一個以計算機率的預測方式來搜尋行動代理人位置的演算法。第五章針對所提出控制網路建置的模型解決方案之演算法及機率預測，兩者各別進行效能評估，最後，在第六章總結本論文，並提出未來可能的研究方向。

第2章

相關研究與文獻探討

文獻探討的部份，依據研究題目之相關領域作一深入之瞭解與學習，以作為我們本身論文研究之基礎，並以其作為依據，提出本論文策略與架構，並作進一步之分析與比較。主要分為四個主題做介紹。2.1 小節將介紹機會網路上搜尋行動代理人之挑戰；2.2 小節介紹「山文誌登山資訊系統」；2.3 小節為控制網路之相關研究；2.4 則為控制網路上控制節點選擇問題之相關研究。

2.1 機會網路上搜尋行動代理人之挑戰

在機會網路上，欲搜尋行動代理人主要有以下幾點困難。

- 在機會網路中，行動代理人存在於移動式節點上，但由於移動式節點具有「移動性」的特質，故即使知道欲搜尋的行動代理人位於哪一個移動式節點，也難以預測此移動式節點目前的位置。
- 由於機會網路本身長延遲的特性，可能導致行動代理人的位置更新速度慢，故搜尋代理人所掌握的被搜尋代理人的位置資料可能與實際位置相差甚遠。

2.2 「山文誌登山資訊系統」簡介

中央研究院週報[20]，指出臺灣雖四面環海，卻是全球高山分佈密度最高的島嶼之一，這也使得登山成為國人熱門的休閒活動。然而，臺灣山區氣象多變，加上地勢複雜，登山意外時有所聞，如何保障登山客生命財產之安全，已成為重要的討論議題。根據

內政部消防署所公告的一項委託研究報告指出，登山意外可以區分為十種類型，其中迷途佔全部山難發生事件的 34%、墜崖佔 32%，其中又以迷途事件發生在積雪時期，其死亡比例竟高達 86%，由此可知當類似事件不幸發生之時，如何能在黃金救援時間內，迅速界定山難發生的地點，並成功營救受困的山友，已成為一項十分重要，且性命相關的課題。

「山文誌登山資訊系統 (CenWits System)」[4,7,21]為一個山區遊客追蹤與環境監控系統，主要目標為收集遊客行走路徑與山區的氣候資訊，再將所收集到的資料做進一步的分析、處理與應用，故可追蹤遊客在國家公園內之分佈狀況，及了解國家公園內之環境變化。此系統由中研院、台灣大學與美國科羅拉多大學所合作研發，目前已被建置於玉山國家公園，使用於遊客的登山活動中，相關的應用有登山服務、山難搜救等。

「山文誌登山資訊系統」除了提供玉山國家公園管理處了解遊客在國家公園內之活動區域，以作為增修各項遊客服務設施之參考外，更可在不幸發生失蹤山難事件時，快速地調閱出「失蹤者的最後出現時間、地點」，有效地為相關救援單位縮小搜救範圍，避免被口耳相傳的模糊資訊所誤導，以提升在七十二小時黃金救援時間內，救援成功的機率，加強保障遊客之生命安全，並節省大範圍搜尋救援工作所耗費之人力、物力成本。

在國家公園內，各地區地形的差異很大，故無線電訊號的傳送會受到環境狀況與遮蔽物的影響，而影響無線通訊的能力。這是因為電磁波傳播所經過的路徑，可能被許多不同的地形或障礙物所影響 (None Line of Sight, NLOS)，而不再只是很簡單的直線傳播 (Line of Sight, LOS)，無線訊號可能會經過建築物、山丘、或樹葉的反射，而產生多重路徑效應，造成訊號放大或衰減。因此，將使得國家公園內的許多地區都無法接收到無線網路的訊號，若要達到每個地點都能收到無線網路的訊號，將會大幅提升建造成本，困難度也會超乎想像。

此外，山區的人口少、自然景觀珍貴，建置過多基地台除了浪費成本，也嚴重破壞自然景觀與生態，故在這種環境下，傳統行動電話系統的覆蓋率通常不高，使得傳統行動電話系統在山區並不可靠，故可進而假設大多數登山客並無行動電話可用。

又因為目前的網路存在一個很大的缺點，即通信端與接訊端必須要在連線狀態才能傳遞資料，如此當遊客在通訊的死角則無法被追蹤到，為了解決此問題，「山文誌登山資訊系統」使用最新的無線微型收發器，搭配全球衛星定位系統，並藉由機會網路這種「耐延遲網路」技術，讓每位遊客彼此於山徑中相遇時，能彼此交換其所攜行的路徑與位置等資訊，最後利用架設於各山頭的無線網路中繼站，來收集、儲存遊客的行走路徑等資料，並傳回控制中心，使控制中心能掌握遊客的動向。

「山文誌登山資訊系統」的運作流程如下：

- 每個登山客攜帶一個約 70 克重、火柴盒般大小的無線微型收發器（內含 CPU、記憶體、天線，可執行使用者指定的程式），並搭配全球衛星定位系統（GPS）。
- 每個無線微型收發器會自動記錄各別登山客的移動路徑，換句話說，無線微型收發器會定時由 GPS 感測器取得登山客的位置，並暫存在模組的快閃記憶體中。
- 當登山客在山徑中相遇時，彼此的收發器便會自動記錄相遇的時間及經緯度，並經由 Zigbee 短距無線接取技術，來互相交換對方收發器中所儲存的路徑與位置資訊。
- 而當登山客行經預先架設於各山頭登山口、涼亭等處的無線網路中繼站（基地台）時，所有儲存在無線微型收發器中的資料（包含自己的路徑紀錄，及所有曾相遇過的登山客所攜之收發器中的紀錄），就會自動藉由基地台來傳回到系統內部的控制中心。

在以上的運作流程中，由於考慮到登山客往往是一天或數天的行程，以及山徑並不如平面道路寬敞，又遊客在山徑中相遇的時間是相當短的，因此，基於無線微型收

發器的電池耗電量、傳輸距離的限制，以及 Zigbee 基地台數量有限等考量，目前在此系統使用 Zigbee 短距離無線傳輸技術，並結合接力傳送的方式，來達到訊息傳遞的目的。此外，由於是使用 Zigbee 短波通訊協定的公共頻道，來傳輸登山客的軌跡資料，故相較於利用現有的衛星天線、3G，其通訊費用是完全免費的。而雖然 Zigbee 傳輸距離短，但在正式通訊前建立連線的速度比使用藍芽技術快，也較使用 IEEE 802.11 無線通訊協定省電，故更適合使用在這種情境中。

在此系統中，基地台的主要功能是接收無線微型收發器中的登山客 GPS 資料，再透過 3G 網路上傳到系統控制中心。此處的基地台可分為簡易型與太陽能型兩種，簡易型是架設在電源供應充足的管理站或是排雲山莊，太陽能型則是利用太陽能板與充電電池，架設在步道旁的涼亭或其他沒有電源供應的地區。此外，基地台也會收集風速、雨量、風向等氣象資料，並利用網路攝影機來做生態的觀測。

此系統目前所使用的最新無線微型收發器，除了體積小、重量輕（約 70 克）以外，也具有防水、防摔、耐熱等特點，以及緊急呼救的功能，實際使用時，登山客只要在出發前充飽收發器的電力，在登山口啟動收發器，夜宿時關閉，不需要使用者手動操作，無線微型收發器就可以自動提供登山客的路徑資訊給系統控制中心。

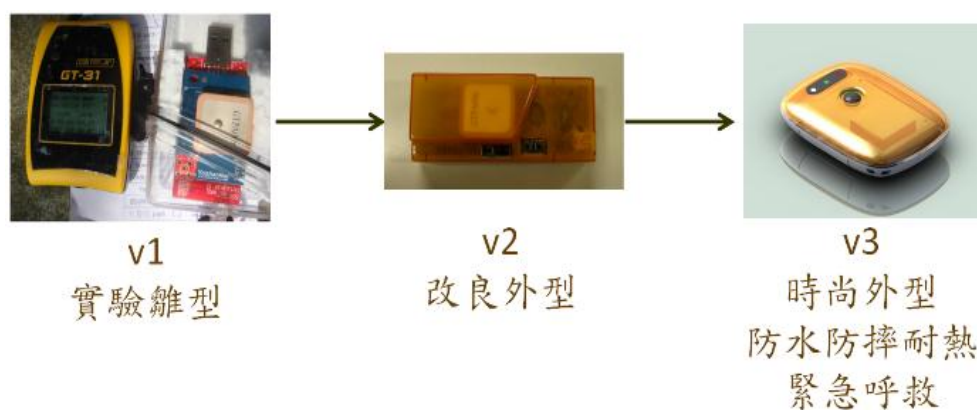


圖 2-1:山文誌登山系統-登山客黑盒子

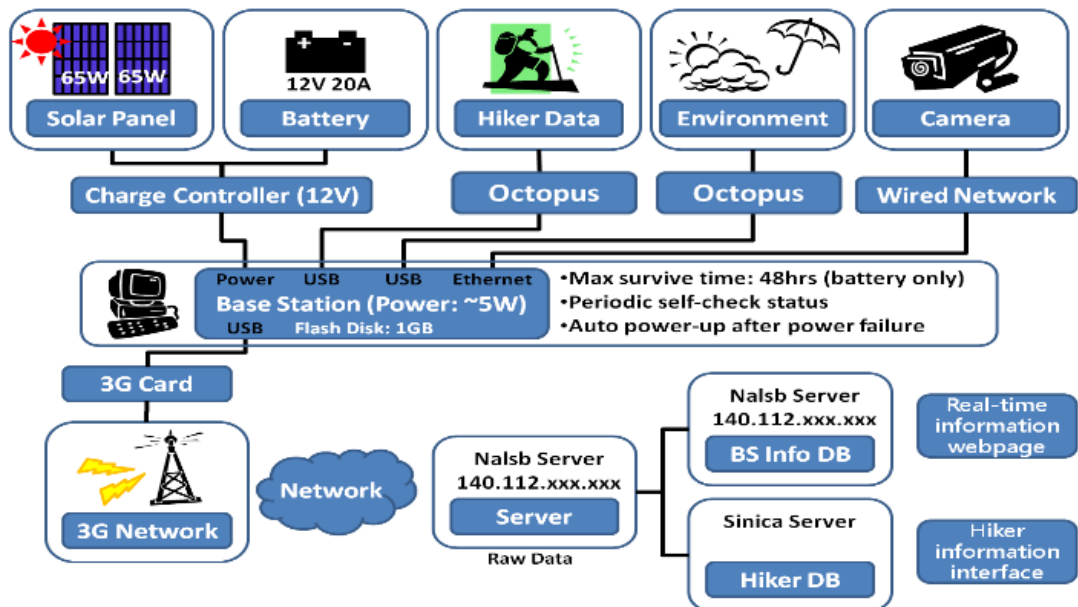


圖 2-2:山文誌登山系統-基地台(簡易型與太陽能型)

2.3 機會網路上使用控制網路搜尋行動代理人

在機會網路上的行動代理人之搜尋，存在著極大的挑戰性。在 Placement of Control Network for Mobile Agents over Opportunistic Networks [22]，提出以「山文誌登山資訊系統」應用情境為例，已知每個登山客都會攜帶一個配備有 GPS 功能的小型設備，簡稱為「行動節點 (mobile node, MN)」，此行動節點會在登山客的移動過程中，自動收集位置資訊，並使用短程無線電來和鄰近的登山客交換彼此的移動資訊。

然而，由於行動代理人是附屬於登山客所攜帶的行動節點上，故當兩個行動節點 (登山客) 互相接近時，行動代理人才可以從其中一個行動節點轉移到另一個行動節點，其行動力受制於登山客的移動行為，由於登山客行走速率差異不大、所在位置分佈不均，故行動代理人的轉移將極為緩慢且需仰賴不確定的碰面機會，使得行動代理人之搜尋效率極低。如圖 2-3 所示，若不存在控制網路 (Control Network) 的情況下，若

目的端實際上比來源端提早一天出發，不管怎麼走，走在後面的來源端，都很難讓行動代理人經由中間節點的跳接來到達目的端。

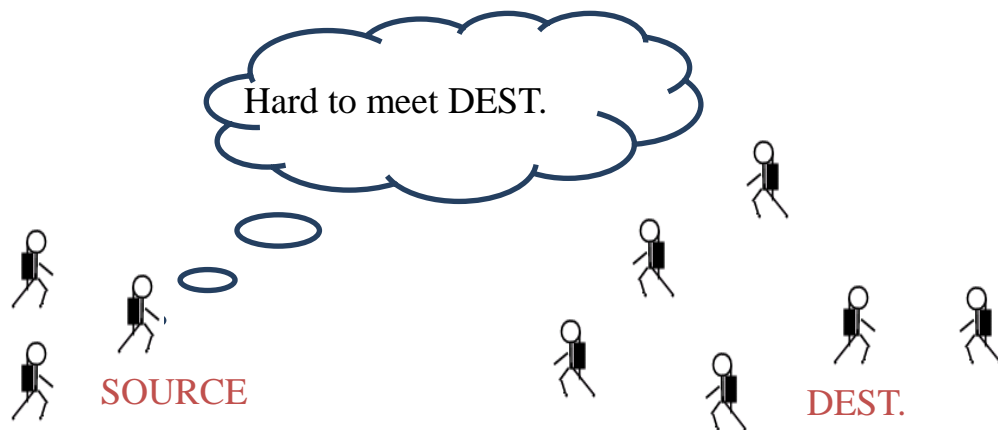


圖 2-3:在機會網路上搜尋行動代理人 (以「山文誌登山資訊系統」應用情境為例)

2.3.1 控制網路概念

為解決上述在機會網路環境上搜尋行動代理人的困難，[22]提出一個高速控制網路的概念，來輔助搜尋代理人的移動，以提高行動代理人的搜尋效率。

假設在山區存在有固定網路 (例如：Internet)，可利用此網路 (在可收到 3G、4G 等系統的信號之處，亦可用 3G、4G 等系統連網) 建立控制點 (Control Point, CP)，形成控制網路 (Control Network)，讓行動代理人在控制點之間快速移動到所欲搜尋的代理人的前方控制點上，再跳至反向登山客所攜帶的裝置 (行動節點) 上，隨著反向登山客的行進，與迎面而來的登山客所攜帶的行動代理人互相溝通，進行搜尋任務。

如圖 2-4 所示，假設目的端比來源端提早一天出發，故走在後面的來源端，極難讓行動代理人經由中間其他行動節點的跳接，來到達目的端。因此，需透過控制網路的輔助，先將來源端送出的搜尋代理人傳送到其最近的控制點，再利用控制網路的快速轉送，將行動代理人傳送到「最接近目的端，且位於目的端未來可能會經過的移動

路線上」的控制點，以上動作是藉由搜尋代理人在控制網路內部的移動，來加速行動代理人之搜尋速度。

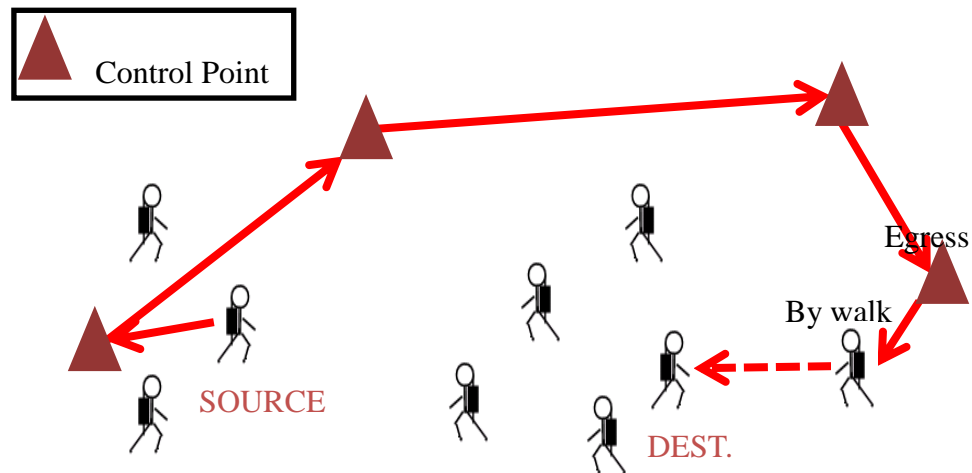


圖 2-4：使用控制網路來輔助行動代理人的搜尋

若此時搜尋代理人已距離目的端夠近，則直接由控制點傳送到目的端，否則，將經由其他與目的端反方向之行動節點的跳接，來逐漸靠近目的端。由於上述的行動節點指的就是由登山客所攜帶的裝置，故事實上是透過登山客的移動，來轉送搜尋代理人，因此，不但轉送速度非常慢，且用來轉送的行動節點必須與目的端的前進方向相反才行。

2.3.2 利用控制網路的搜尋策略

當有了控制網路後，可以修改現有的行動代理人之搜尋機制，來符合機會網路上的需要。

修改 Basic Binary Search (BBS) Algorithm 為例，運作流程如下。

1. 先利用現有的搜尋演算法，找到最接近行動代理人的控制點。例如：先將控

制點依照行動代理人的移動路徑方向排列，再利用 Basic Binary Search (BBS) Algorithm，從中間的控制點開始找，若發現行動代理人已在先前經過此控制點，則往後找，否則往前找。

2. 找到最接近行動代理人的控制點後，將用來搜尋用的行動代理人，送到行經此控制點的反向行動節點上，再由這些行動節點逐步接近欲搜尋的行動代理人。

可使用 Basic Binary Search (BBS)Algorithm 搜尋方式作為搜尋的策略，而在下一章本論文將針對搜尋演算法另提出一種利用機率預測的搜尋演算法，預測目標代理人所在之位置，並利用控制網路將搜尋代理人送至距離目標代理人最近即將走訪的控制節點上，以減少搜尋目標代理人所花費的時間。因一段健行路徑健行者所需的健行時間，可能是半小時到數小時不等，故若預測目標代理人所在位置準確，對於改善整體搜尋目標代理人所花費的時間大有助益，則在行動節點訊息交換或是緊急就難搜救上可提升其效益。

2.4 控制網路上控制節點選擇問題之相關研究

2.4.1 控制網路的建置

在[22]中，提出控制網路的建置模型，並沒有一定的標準，在不同的環境下，所適合的控制網路皆不盡相同，如要全盤考慮，其面向與範圍將非常龐大與複雜，故僅以「山文誌登山資訊系統」為例，來設計適用於山區環境與登山情境的控制網路，本文也依照相同假設，設計搜尋方法。

2.4.2 環境假設

以「山文誌登山資訊系統」為背景，可假設原本在山上就已存在網路，只是登山客在野外活動中不會、也不能去用到這個網路。因此，假設服務範圍內已存在有線或無線封包網路（例如：Internet），是相當合理的。

此外，基於此研究背景的登山應用，也作以下幾點環境假設。

1. 登山客遵循某些健行路徑在移動。
2. 人（行動節點）的移動速率差異不大。
3. 原則上候選控制點是位於健行路徑的交界處或端點，但若候選控制點之間的健行路徑仍太長，也可在路徑中間另加候選控制點。

2.4.3 設計考量與目標

為了建置成本的考量及避免自然景觀的破壞，在控制網路的建置上，必須有效地限制控制點的數量。以下為[22]所提出的控制點選擇問題（Control Point Selection Problem, CPSP），將考慮如何在有限的預算資源下，挑選適當的候選控制點，來組成最佳的控制網路。

這些模型的目標與限制條件都遵循以下幾個原則：

1. 限制：整體的建置花費不得超過可用資源之上限。
2. 目標：
 - 盡可能服務到最多的行動代理人或行動節點。
 - 能快速地找到任一個行動代理人或行動節點。

2.4.4 控制點選擇問題—最大總流量模型

假設每一條健行路徑的登山客流量為已知，此模型在控制網路的建置策略上，是在限制可用資源的前提下，以獲得最大的行動節點之總流量為目的，來選擇控制點的佈建地點，歸納如下。

- 限制：整體的建置花費不得超過可用資源之上限。
- 目標：最大化控制點所涵蓋的行動節點之總流量。

在此模型中，假設只在健行路徑的交界處或端點存在有候選控制點，且候選控制點之間的距離長度皆相等，故最大化控制點所涵蓋的行動節點之總流量，即能代表最大化地提升服務到行動節點的機會。但由於此模型的設計目標，可能導致控制點的佈建地點，可能只集中在少數幾條行動節點流量較多的路徑上。當控制網路的搜尋範圍，只能涵蓋到所有健行路徑的一小部分時，將使得部分行動節點很容易就遠離了控制網路所能提供服務的區域，而造成行動節點的搜尋成功率大為降低，或搜尋時間大幅增加。

此模型的控制網路建置設計，除了可能導致浪費可用資源外，更可能造成搜尋不到行動代理人的嚴重後果。

2.4.5 控制點選擇問題—最大涵蓋率模型

基於上述模型的缺點，另設計了最大涵蓋率模型，在此模型加入以下兩個設計目標。

1. 必須涵蓋所有健行路徑。因為行走在沒被涵蓋到的路徑上的人，可能永遠無法被找到。
2. 控制點建置位置分佈均勻。因為我們希望今天、明天、前幾天出發的人，都可以很快地被找到。

故此模型的設計如下：

- 1、每一條健行路徑上，至少需建置一個控制點。
- 2、盡量讓佈建的位置平均分散，以擴大控制點所涵蓋的區域。

但由於 Graph 由 node 和 edge 所組成，並沒有面積的概念，所以無法輕易地用較精確的數學模型，來表示此模型的設計概念，故將此模型的設計，轉化如下：

- 限制一：整體的建置花費不得超過可用資源之上限。
- 限制二：控制點的建置，需涵蓋到所有的健行路徑。
- 目標：最大化控制點所涵蓋的 edge 數。

此模型為一個控制點的涵蓋率問題，並隱含「每一條 edge 的長度都一樣長」的假設，故在此模型中並沒有考慮到候選控制點之間路段的長度，如此可能導致有些長路段（例如：2 小時路程），卻和短路段（例如：30 分鐘路程）所設的控制點數量相同。另外，也應考量此路段的重要性，例如：在容易發生山難的地方建置控制點有較高的優先度，讓此路段的控制點較為密集，才能提供更好的服務。

2.4.6 控制點選擇問題—最大加權涵蓋率模型

為了彌補最大涵蓋率模型的不足，提出一個考慮更周詳的最大加權涵蓋率模型，首先，將環境假設修改為「不只在健行路徑的交界處或端點存在有候選控制點，若候選控制點之間的健行路徑仍太長，也可在路徑中間另加候選控制點」，並加入路段重要性的考量，希望能將候選控制點之間的路段長度、重要性等狀況，也一併考慮進去，做為此路段的重要性權值（權重），主要設計如下。

- 限制一：整體的建置花費不得超過可用資源之上限。
- 限制二：控制點的建置，需涵蓋到所有的健行路徑。
- 目標：最大化控制點所涵蓋 edges 的總權值。

第3章

機會網路上行動代理人搜尋控制點選擇問題及解決方案

由於行動代理人搜尋機制的主要任務，為發送搜尋代理人來找特定的行動代理人，因此，控制點的可能位置，若能涵蓋、服務到最多行動節點的話，對於快速地找到行動代理人的機會，相信會是比較大的。若能加入此健行路段上的人數流量，會更加符合實際山區情況，為能達到此目標，本篇論文在控制點選擇問題上，另提出最大複合加權涵蓋率模型。

3.1 控制點選擇問題-最大複合加權涵蓋率模型

本文提出一個最大複合加權涵蓋率模型，除了將候選控制點之間的路段長度、重要性等狀況加入考量，另外在山區應考慮的重要條件，也就是健行路徑上的人數流量，也將一併做一綜合考慮，主要設計如下。

假設在每條健行路徑上的人數流量為已知，將候選控制點之間路段的重要性，同最大加權涵蓋率模型，設重要性權值（權重）。並將每一條健行路徑登山客的流量，設為流量權值（權重），此模型在追求其選擇的控制點所涵蓋的路段權重及經過此控制點的登山客總流量兩者乘積達到最高，考慮路段的重要性權重及各路徑登山客流量權重，對選擇控制點問題做一綜合考量。

- 限制一：整體的建置花費不得超過可用資源之上限。
- 限制二：控制點的建置，需涵蓋到所有的健行路徑。
- 目標：控制點所涵蓋的路段權重及經過此控制點登山客的總流量，兩者乘積最大化。

在將此模型轉化為數學表示式之前，先定義一個數學函式，如下：

$$P(M) = \{ \sum (z_{ij} \cdot f_k) \mid e_{ij} \in M', t_k \text{ passes } e_{ij} \}$$

其中， z_{ij} 表示已知路段 e_{ij} 之重要性權值(權重)， f_k 為經過此路段上的人數流量， M' 為被選中的節點的集合。 $z_{ij} \cdot f_k$ 數值愈大，代表重要性愈高，故函式 $P(M)$ 為在集合 M 中這些路段重要性權值之總和，亦即，選擇這些路段來建置控制點時可獲得的總收益 (profit)。

3.2 最大複合加權涵蓋率的數學模型

以下為控制點選擇問題—最大複合加權涵蓋率(CPSP-Utility-Flow) 的數學模型。

Given a graph $G=(V, E)$, where

- $V = \{v_1, \dots, v_n\}$ is the set of possible nodes for building control points,
- $E = \{e_{ij} \mid v_i, v_j \in V\}$ is the set of edges (trail segments) between v_i and v_j ,
- $W = \{w_i \mid v_i \in V\}$ is the set of the cost (required resources) of building a control point on v_i ,
- C is the available resources,
- $T = \{t_1, \dots, t_m\}$ is the set of hiking trails, each of which is a path, a sequence of nodes, used for hiking, and
- $Q = \{q_{ki} \mid v_i \in V, t_k \in T\}$ is a passing_trail matrix,

where $q_{k_i} = \begin{cases} 1 & \text{if trail } t_k \text{ passes node } v_i \\ 0 & \text{otherwise,} \end{cases}$

- $Z = \{z_{ij} | v_i, v_j \in V\}$ is the weight of the edge e_{ij} ,
- $F = \{f_k | t_k \in T\}$ is the set of hiker flows, which is the number of hikers per time unit passing a trail,

the *CPSP- Utility-Flow Problem* is to find $S \subseteq V$, to

maximize $P(S)$

subject to

$$\bigcup \{q_{k_j} \cdot x_i\} = T \quad \text{and} \quad \sum_{i \in V} w_i x_i \leq C, \quad \text{where } x_i = \begin{cases} 1 & \text{if } v_i \in S, \\ 0 & \text{otherwise.} \end{cases}$$

3.3 解決方案的演算法

控制點選擇問題—最大總流量模型、最大涵蓋率模型已由 [19]證明此兩個模型皆為 NP-complete 問題，最大複合加權涵蓋率模型也同樣為 NP-complete 問題，本研究省略其複雜度 NP-complete 之證明，著重在提出控制點選擇問題的啟發式演算法，分別對最大涵蓋率模型、最大加權涵蓋率模型，提出其改進之啟發式演算法，並針對本文另提出的最大複合加權涵蓋率，也提出一個啟發式演算法解決控制點選擇問題。

由於最大總流量 (CPSP-Flow)數學模型已有許多現成的 0-1 knapsack problem 之解決方案演算法可用，故本文僅提出最大涵蓋率 (CPSP-Coverage)、最大加權涵蓋率 (CPSP-Utility)及最大複合加權涵蓋率(CPSP-Utility-Flow)數學模型的解決方案改進演算法。

3.3.1 控制點選擇問題的評估指標

首先，我們定義幾個評估指標，做為 sub-optimal solution 與 optimal solution 比較的依據。在先前研究中，已經定義 MC distance 為「某一個行動節點到它最近的控制點之間的距離」。然而，由於在同一條 edge 上，移動方向相反的行動節點，分別與其最近的控制點之間的距離不一定相同，故於之後的實驗中，我們假設「在每條 edge 上，往返兩方向各有一個行動節點」，來與 MC distance 相關的評估指標進行計算。表 3-1 為控制點選擇問題 (CPSP) 的相關評估指標之定義。

表 3-1：控制點選擇問題 (CPSP) 的評估指標

Name	Definition
flow coverage	The summation of flow covered by all the nodes which were selected as control point.
edge coverage	The summation of edge covered by all the nodes which were selected as control point.
weighted edge coverage	The summation of weighted edge covered by all the nodes which were selected as control point.
populated coverage	The summation of weighted edge multiplying the flow of edge while the edge was covered by all the nodes which were selected as control point.

mean MC distance	The average distance of all mobile nodes reaching its nearest control point. The number of edge indicates the distance, calculated from the summation of all mobile nodes' MC distance over the total number of the mobile nodes.
max MC distance	The longest distance for any arbitrary mobile agent reaching its nearest control point, i.e., the maximization of all the mobile nodes' MC distance.
mean weighted MC distance	Same as the mean MC distance described above, only that the distance was indicated by the summation of weighted edge rather than the number of the edge. Assuming that the weights of all edge in the end of trails are all one.
max weighted MC distance	Same as the max MC distance described above, only that the distance was indicated by the summation of weighted edge rather than the number of the edge. Assuming that the weights of all edge in the end of trails are all one.
mean populated MC distance	Same as the mean MC distance described above, only that the distance was indicated by every edge weight multiplying the certain edge flow. Assuming that the weights of all edge in the end of trails are all one.

3.3.2 CPSP-Coverage 改進式解決方案- NPF2 演算法

為了找 CPSP-Coverage 的 sub-optimal solution，[18]提出一個啟發式演算法，命名為 Node Potential First (NPF) 演算法，本研究另提出此演算法之改進啟發式演算法，命名為 Node Potential First 2 (NPF2) 演算法，表 3-2 將列出在 NPF 及 NPF2 演算法中所運用到的參數及其意義。

表 3-2:Node Potential First (NPF)及 Node Potential First 2 (NPF2)

演算法參數表

Notation	Definition
edge degree	The edge degree number reached by the nodes.
node cost	The cost of the node selected as a control point.
selected edge degree	The summation of edge has been selected with the initial value being zero.
unselected edge degree	edge degree – selected edge degree
edge potential	The cover edges potential in the certain node. edge potential = (unselected edge degree/node cost)
trail hit count	The number of unmark trails gone through by the nodes.
trail potential	The cover trails potential in the certain node. trail potential = (unselected_trail degree/node cost)

Node Potential First (NPF) 演算法之運作流程，是優先選擇 edge potential、再挑選 trail hit count 的值最大，兩者相同再選擇 node cost 最小的候選控制點，在每次選擇一個候選控制點後，更新 edge potential、trail hit count 的值，以確保所選的下一個候選控制點能涵蓋到新的 edges 及 trails，避免重覆計算所涵蓋到的 edge 數量。

本文所提出的 NPF2 演算法的設計重點，如下：

- 所選的候選控制點之 node 的 cost 總和，不可超過 available resources。
- 優先選擇 α (edge potential) + β (trail potential) 值最大，若值相同再挑選 node cost 值最小的為候選控制點。

Node Potential First 2 (NPF2) 演算法之運作流程，優先選擇 α (edge potential) + β (trail potential) 值最大的節點為候選控制點，其中 α 、 β 由使用者自訂權重，若其值相同再挑選 node 之 cost 最小的候選控制點。我們也在每次選擇一個候選控制點後，更新 edge potential、trail potential 的值，以確保所選的下一個候選控制點能涵蓋到新的 edges 及 trails，避免重覆計算所涵蓋到的 edge 及 trail 數量。

此演算法之設計理念為考量不同目標函數給予不同權重做加權計算，避免當選擇了其候選節點為控制節點，但可能兩候選點 edge potential 的值差異不大，trail potential 的值差異卻很大時，優先誤選擇了此點，為避免此情況發生，此演算法設計精神為取兩者加權權重高的優先選擇選為控制節點。下圖為 Node Potential First 2 (NPF2)演算法之運作流程。


```

Candidate ListV ← all nodes

Iterate until Committed Cost >= Available Resources
{
    Deleting the node from the candidate ListV with node weight > Remain Resources.

    Selecting the node from the candidate ListV with the largest value of
 $\alpha$  ( edge potential ) +  $\beta$ ( trail potential ) and smallest value of
node cost in ascending order.

    Updating the value of edge potential and trail potential of every node by marking
covered trails.
}

```

圖 3-1: Node Potential First 2 (NPF2) 演算法之運作流程

表 3-3: NPF、NPF2 演算法比較

	NPF Algorithm	NPF2 Improved Algorithm
restriction	The summation of the cost by the nodes selected as the control point, can't exceed available resources	
selection order	<ol style="list-style-type: none"> 1. edge potential maximum 2. trail hit count maximum 3. node cost value minimum 	<ol style="list-style-type: none"> 1. α (edge potential) + β(trail potential) maximum 2. node cost value minimum

3.3.3 CPSP-Utility 改進式解決方案- NPF-U2 演算法

在此章節，我們將對[18]提出的 NPF-U 演算法作修改，提出其改良式演算法，命名為 NPF-U2 演算法，來解 CPSP-Utility 問題。NPF-U 與 NPF-U2 的差異在於優先挑選候選節點「 α (unselected edge weight /node cost)+ β (trail potential) 」。權值最高的候選節點做為控制節點。表 3-4 列出在 NPF-U 及 NPF-U2 演算法中所運用到的參數及其意義，其餘未列出的參數與 NPF 及 NPF2 參數相同。

表 3-4：Node Potential First U (NPF-U)及
Node Potential First U2 (NPF-U2)演算法參數表

Notation	Definition
selected edge weight	In this node, some edges weighted were selected with the initial value being zero.
unselected edge weight	The number of cover edges weighted increased in the result of selecting the certain node.

Node Potential First-U (NPF-U) 演算法之運作流程，優先選擇「unselected edge weight /node cost」、trail hit count 的值最大，而 node cost 最小的候選控制點，並在每次選擇出一個候選控制點後，更新 unselected edge weight /node weight、trail hit count 的值。

本文提出的 NPF-U2 演算法的設計重點，如下：

- 所選的候選控制點之 node weight 總和，不可超過 available resources。
- 優先選擇 α (「unselected edge weight /node weight」) + β (trail potential) 值最大，若值相同再挑選 node cost 值最小的為候選控制點。

下圖 3-2 為 Node Potential First-U2 (NPF-U2) 演算法之運作流程，此演算法之設計理念同 NPF2，為同時考量不同目標函數給予不同權重做加權計算。

```

Candidate ListV ← all nodes

Iterate until Committed Cost >= Available Resources
{
    Deleting the node from the candidate ListV with node weight > Remain Resources.

    Selecting the node from the candidate ListV with the largest value of
 $\alpha$  (unselected edge weight ) +  $\beta$ ( trail potential ) and smallest
value of node cost  in ascending order.

    Updating the value of edge potential and trail potential of every node by marking
covered trails.
}

```

圖 3-2：Node Potential First-U2 (NPF-U2) 演算法之運作流程

表 3-5:NPF-U、NPF-U2 演算法比較

	NPF-U algorithm	NPF-U2 improved algorithm
restriction	The summation of the cost by the nodes selected as the control point, can't exceed available resources	
selection order	<ol style="list-style-type: none"> 1. unselected edge weight maximum 2. trail hit count maximum 3. node cost minimum 	<ol style="list-style-type: none"> 1. α (unselected edge weight) + β(trail potential) maximum 2. node cost minimum

3.3.4 CPSP-Utility-Flow 解決方案- NPF-PF 演算法

此節我們對本文所提出的 CPSP-Utility-Flow 模型，提出一啟發式演算法，命名為 NPF-PF 演算法，來解 CPSP-Utility-Flow 問題。此演算法修改先前提出 NPF-U2 演算法，優先挑選

「 $(\alpha (\text{unselected edge weight / node cost}) + \beta (\text{trail potential})) \cdot \gamma (\text{flow_potential})$ 」值為最高的候選節點做為控制節點。

表 3-6:NPF-PF 演算法之參數表

Notation	Definition
flow coverage	The summation of flow covered by all the nodes which were selected as control point.
flow potential	The summation of flow covered by all the nodes which were unselected as control point / node cost

圖 3-3 為 Node Potential First-PF (NPF-PF) 演算法之運作流程

Candidate ListV \leftarrow all nodes

Iterate until Committed Cost \geq Available Resources

{

Deleting the node from the candidate ListV with node weight $>$ Remain Resources.

Selecting the node from the candidate ListV with the largest value of

$(\alpha (\text{unselected edge weight} / \text{node cost}) + \beta (\text{trail potential})) \cdot \gamma (\text{flow_potential})$

, and smallest value of node cost in ascending order.

Updating the value of 「unselected edge weight / node weight」

and trail potential and flow_potential of every node by marking covered trails.

}

圖 3-3 : Node Potential First-PF (NPF-PF) 演算法之運作流程

第4章

機會網路上行動代理人位置之預測方法

4.1 預測方法設計理念與目標

在機會網路上，要對一個行動節點下指令或回收任務資訊，首要的第一件事就是要先找到這個行動節點，方能對其執行控制動作。我們假設已經在所欲服務的區域事先佈建控制網路並導入行動代理人之機制，利用控制網路內部的快速轉送，來加速 search agent 的傳送，則此 search agent 即可透過控制網路找到適當的一個控制節點做為出口 (Egress) 跳出。但行動節點在健行路徑上的正確所在實際位置無法確切得知，除非該節點有正常的回報正確位置，否則只能預測其可能所在路段位置，再選取最靠近此行動節點前方的控制節點做為 Egress 跳出，接著附著在反向健行者身上，再逐步接近欲搜尋的目標。若所預測出目標節點的所在路段不準確，則浪費了此次搜尋時間，至少是一段健行路段的健行者步行走訪時間，搜尋失敗的代價很高。而在山區的環境底下，每一路段的距離可能需費時數小時之久，若能提高預測的準確度，將大幅縮短搜尋 mobile agent 的時間。

為了解決在機會網路環境上搜尋行動代理人的困難，我們提出一個預測方法，來輔助搜尋代理人的移動，以提高行動代理人的搜尋效率。

4.2 問題定義

4.2.1 控制網路下搜尋名詞定義

當有了控制網路後，除了可以使用先前所提出的一個簡單搜尋方式，Basic Binary Search (BBS) Algorithm [20]，這個現有的搜尋演算法來尋找目標節點，我們希望能提出更精準的另一預測搜尋演算法，表 4-1 是本方法所用到的名詞定義。

表 4-1:名詞定義

name	Definition
Search agent	To find the target agent under the control network. There could be several search agents which are managed and delivered by source agents.
Source	Source itself is also an agent which manages search agent, sends out search agent from Ingress, and records search time spending of search agent finding target agent.
Egress	An Egress is the closest control point right forward to target agent under control network, as an exit for search agent to jump out and head-on approach target agent.
Ingress	Ingress is the entrance for source agent to send out search agent under the control network, at the starting point of trail where target agent located under.
Messenger	Messenger is the mobile node which helps search agent to jump out form Egress to approach target agent.

下圖 4-1 為控制網路在山區環境上的示意圖。

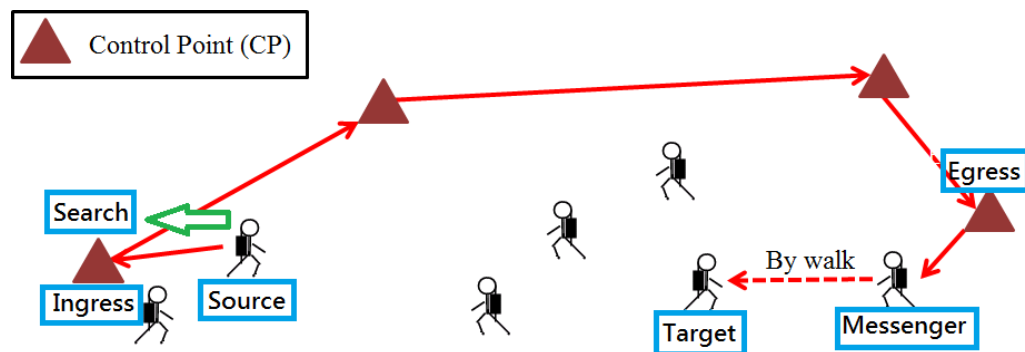


圖 4-1:控制網路底下行動代理人的搜尋及其名詞定義

假設在服務區域內已建置有控制網路，我們從 Source (其本身也是一個 mobile node) 將 search agent 從此條 trail 起始端的控制節點(稱為 Ingress)送上控制網路，此 search agent 可利用控制網路快速移動到我們所預測的目標位置前方控制節點上做為出口(稱為 Egress)，再跳至反向登山客 (行動節點) 上，此登山客稱為 Messenger，隨著反向登山客的行進，逐步接近迎面而來的 target agent 進行搜尋任務，找到後即可交換、傳遞訊息。

4.2.2 問題模型

- 問題模型一:

以「山文誌登山資訊系統」為背景，我們可假設原本在山上就已存在網路及控制網路，此外，基於此研究背景的登山應用，我們也作以下幾點環境假設。

1. 登山客在一條沒有叉路的 trail 上移動
2. 健行者 (行動節點) 人的移動速率差異不大
3. 山區已佈建控制網路，且 trail 兩端均建構控制節點，共有 n 個控制節點
4. 假設 target agent 沿著一條事先決定的 trail 行走，已知此 trail 上各路段所需步行時間的機率分佈為一穩定且已知的機率分佈，每個機率分佈當中的隨機變數均是 i.i.d.(independent and identically distributed)獨立同分佈。此機率分佈可假設為

一個靠右傾斜的 Gamma 分佈

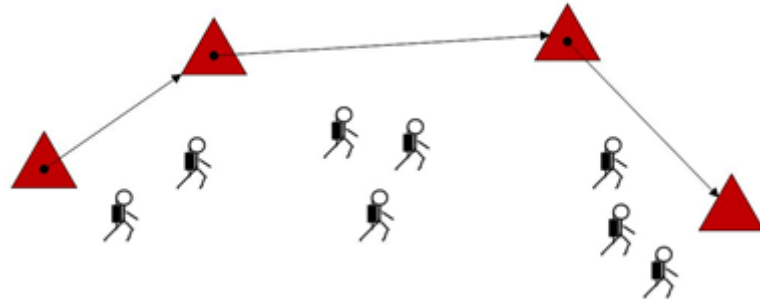


圖 4-2:問題模型一情境

目標:

找到 target agent 的搜尋次數最少

則問題模型一為求出一機率函數:

$$P_{e,t} : \text{probability of the target agent being located at edge } e \text{ on time } t \quad (1)$$

• 問題模型二

同樣以「山文誌登山資訊系統」為背景，假設原本在山上就已存在網路及控制網路做以下幾點環境假設。

1. 登山客在一條有叉路的 trail 上移動
2. 健行者 (行動節點) 人的移動速率差異不大
3. 山區已佈建控制網路，且 trail 兩端均建構控制節點，共有 n 個控制節點
4. 假設 target agent 沿著一條有叉路的 trail 行走，given 各路段叉路選擇方向的機率，每個叉路選擇機率總和為 1，並限制每段路段只走一次

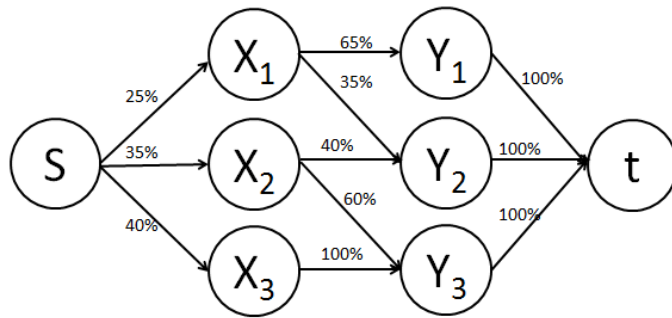


圖 4-3:問題模型二情境

目標:

找到 target agent 的搜尋次數最少

則問題模型二為當路徑允許叉路時，求出一機率函數:

$$q_{e,t} : \text{probability of the target agent being located at edge } e \text{ on time } t \quad (2)$$

4.3 機率預測函數

本研究選用假設健行者在每一路段上所花費的健行時間符合Gamma分配的機率。選用Gamma分配理由有二，Gamma分配的機率函數密度圖較符合實際在山區健行所花費時間曲線圖的特性，健行者走完一段健行路徑的時間，大多數人所花費時間peak普遍會集中在中間偏左，形成一個右偏的分佈，本研究所主要探討的行為是健行者在各路段上所花費的時間，屬於連續型變數，且必為正值。透過計算機率的方式預測其各路段機率，也可透過直接修改參數的方式，即可調整機率分配的設定。

在本文的研究上限制假設各路段所花健行時間均為兩兩互相獨立事件且假設服從Gamma分配，但可能與健行者實際上的統計時間分配有所出入。

4.3.1 Gamma 分配

Gamma 分配，其隨機變數 t 的機率密度函數為：

$$t \sim \text{Gamma}(\alpha, \beta) = \frac{1}{\Gamma(\alpha)\beta^\alpha} t^{\alpha-1} e^{-\frac{t}{\beta}} \quad (3)$$

其中 Gamma 分佈中的參數 α ，稱為形狀參數(shape parameter)， β 稱為尺度參數(scale parameter)。 α 影響 p.d.f.(機率密度函數)圖形之陡峭程度， β 影響散佈程度，期望值為 $E(t)=\alpha\beta$ ，變異數 $\text{Var}(t)=\alpha\beta^2$ 。機率密度函數 $f(t)$ 與隨機變數 t 的關係如圖 4-2。

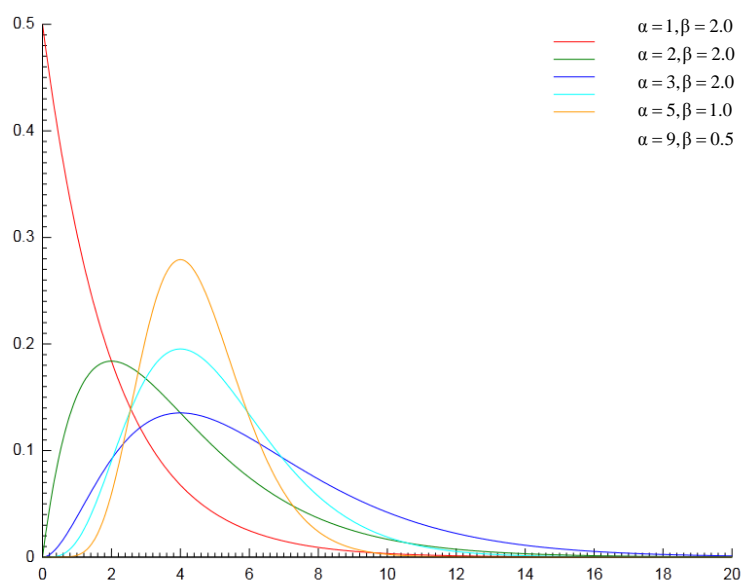


圖 4-4:Gamma 分配機率密度函數

其中 α 和 β 是伽瑪分佈的參數，其值均大於 0， $\Gamma(\alpha)$ 是伽瑪函數：

$$\Gamma(\alpha) = \int_0^{\infty} y^{\alpha-1} e^{-y} dy \quad (4)$$

且如果變量的下限不是 0，而是另外一個有限值 a ，那麼它僅影響分佈函數在 x 坐標中的位置而不影響函數的形狀，公式如(5):

$$t \sim \text{Gamma}(\alpha, \beta) = \frac{1}{\Gamma(\alpha)\beta^\alpha} (t-a)^{\alpha-1} e^{-\frac{(t-a)}{\beta}} \quad (5)$$

且 Gamma distribution 具有 Summation 特性:

當兩隨機變數服從 Gamma 分配，互相獨立，且單位時間內頻率相同時，Gamma 分佈具有加成性。若 X_i 是一個 $\Gamma(\alpha_i, \beta)$ 分佈 for $i = 1, 2, \dots, N$ (具有相同的尺度參數 β)，則具可加性。

$$\sum_{i=1}^N X_i \sim \text{Gamma}\left(\sum_{i=1}^N \alpha_i, \beta\right) \quad (6)$$

4.3.2 模型一使用計算機率預測法之搜尋程序

已知模型一為一條山區上的登山路徑，沒有叉路，路徑前後兩端均有控制節點，可假設共有 N 個控制節點，每段路段所花的健行時間均是一個機率分佈，且為 Gamma 分佈，則問題為求出此目標節點在任一時間點在各路段上的機率，接著就可從機率最高的路段做搜尋，此模型一問題可簡化為推導出 $P_{e,t}$ 之值。

為簡化問題的困難度我們假設各段時間的 Gamma 分佈其 scale parameter 相同。則依照 Gamma 分佈加成性的特性，當兩隨機變數服從 Gamma 分佈，兩兩獨立且尺度參數(scale parameter) 相同時，Gamma 分佈具有加成性，故可加總所有路段得到整條路徑的 Gamma 分佈。

● $P_{e,t}$ 之計算

$X_1, X_2, X_3, \dots, X_n$: A continuous random variable X_k , time spent on edge i , which has a Gamma distribution with parameter $\lambda > 0$, if for $k = 1, 2, \dots, n$

$P_{e,t}$: probability of the target agent being located at edge e on time t .

$P_{e_1,t} = P(X_1 > t)$

$P_{e_2,t} = P(X_1 < t, X_1 + X_2 > t)$

$P_{e_3,t} = P(X_1 + X_2 < t, X_1 + X_2 + X_3 > t)$

$P_{e_n,t} = P(X_1 + X_2 + X_3 + \dots + X_{n-1} < t, X_1 + X_2 + X_3 + \dots + X_n > t)$

$P_{e_{not_1\dots n},t} = P(X_1 + X_2 + X_3 + \dots + X_n < t)$

圖 4-5: $P_{e,t}$ 之推導

其中第二段至第 n 段為計算其聯合密度機率函數：

$$P_{e_n,t} = \begin{cases} \int_t^\infty \frac{x_n^{\alpha_n-1} e^{-\frac{x_n}{\beta}}}{\Gamma(\alpha_n)\beta^{\alpha_n}} dx_n & n = 1 \\ \int_0^t \frac{x_{n-1}^{\alpha_{n-1}-1} e^{-\frac{x_{n-1}}{\beta}}}{\Gamma(\alpha_{n-1})\beta^{\alpha_{n-1}}} dx_{n-1} - \\ \int_0^t \int_0^{t-x_{n-1}} \frac{x_{n-1}^{\alpha_{n-1}-1} e^{-\frac{x_{n-1}}{\beta}}}{\Gamma(\alpha_{n-1})\beta^{\alpha_{n-1}}} \frac{x_n^{\alpha_n-1} e^{-\frac{x_n}{\beta}}}{\Gamma(\alpha_n)\beta^{\alpha_n}} dx_n dx_{n-1} & n = 2 \dots n \end{cases} \quad (7)$$

當求出 $P_{e,t}$ ，我們即可使用 $P_{e,t}$ 提出機率預測搜尋演算法-單一路徑 (PPM-S)。

PPM-S Algorithm {

input: A continuous random variable X_k , time spent on edge i , which has a Gamma distribution with parameter $\lambda > 0$, if for $k = 1, 2, \dots, n$

output: $P_{e,t}$

Search BEGIN

Source sends search agent from an Ingress onto Control Network;

Take the C.P. nearest to the edge with highest probability $P_{e,t}$

as the Egress;

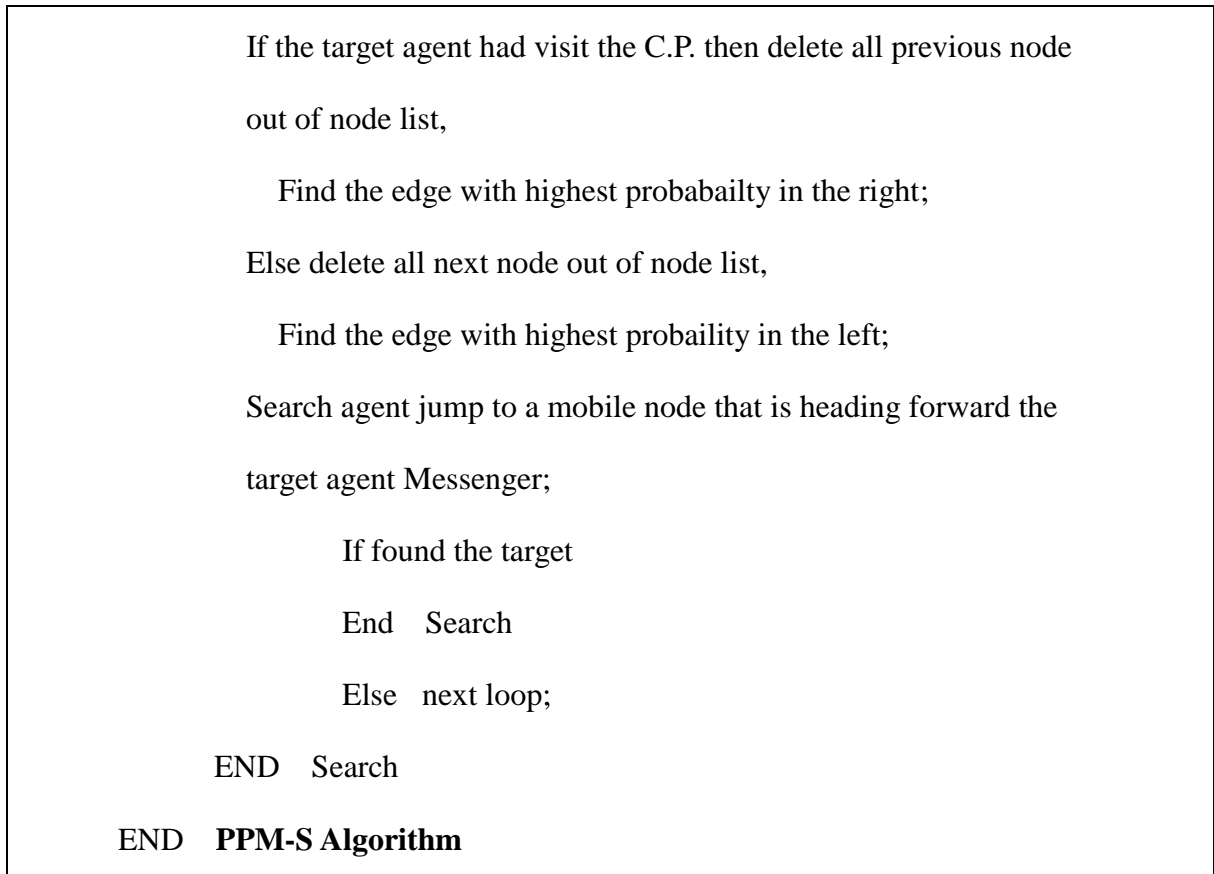


圖 4-6:機率預測搜尋演算法-單一路徑(PPM-S)

4.3.3 模型二使用計算機率預測法之搜尋程序

模型二適用於有叉路的登山路徑，且各端點均有控制節點，假設共有 N 個控制節點，與模型一相同每段路徑所花的時間皆為一個 Gamma 分佈的隨機變數，機率預測搜尋法先求出此目標節點在任一時間點在各路段上的機率 $q_{e,t}$ ，再從機率最高的路段開始搜尋目標 Agent，利用模型一所導出之 $P_{e,t}$ 及兩個數學函式(8)(9)，可計算出 $q_{e,t}$ 。定義數學函式，如下。

$$P_l: \text{ probability of target agent walking on path } l \quad (8)$$

$$q_{l,e,t}: \text{ probability of target agent on edge } e \text{ at time } t \text{ if it takes path } l \quad (9)$$

同模型一假設各路段所花時間為 Gamma 的機率分佈且其 scale parameter 相同，求 target agent 於任一時間點位於每一路段之機率。利用模型一推導出的 $P_{e,t}$ ，可計算出 $q_{l,e,t}$ 。

● $q_{e,t}$ 之計算

$X_1, X_2, X_3, \dots, X_n$: A continuous random variable X_k , time spent on edge i, which has a Gamma distribution with parameter $\lambda > 0$, if for $k = 1, 2, \dots, n$

$q_{e,t}$: probability of the target agent being located at edge e on time t

Step1:

$$P_{e_n,t} = \begin{cases} \int_0^\infty \frac{x_n^{\alpha_n-1} e^{-\frac{x_n}{\beta}}}{\Gamma(\alpha_n)\beta^{\alpha_n}} dx_n & n = 1 \\ \int_0^t \frac{x_{n-1}^{\alpha_{n-1}-1} e^{-\frac{x_{n-1}}{\beta}}}{\Gamma(\alpha_{n-1})\beta^{\alpha_{n-1}}} dx_{n-1} - \\ \int_0^t \int_0^{t-x_{n-1}} \frac{x_{n-1}^{\alpha_{n-1}-1} e^{-\frac{x_{n-1}}{\beta}}}{\Gamma(\alpha_{n-1})\beta^{\alpha_{n-1}}} \frac{x_n^{\alpha_n-1} e^{-\frac{x_n}{\beta}}}{\Gamma(\alpha_n)\beta^{\alpha_n}} dx_n dx_{n-1} & n = 2 \dots n \end{cases}$$

Step2:

$$q_{e,t} = \sum_{\forall l} p_l \cdot q_{l,e,t} \tag{10}$$

圖 4-7: $q_{e,t}$ 之推導

求出 $q_{e,t}$ ，我們即可使用 $q_{e,t}$ 計算機率的預測方式提出機率預測搜尋演算法-多重路徑 (PPM-M)。

PPM-M Algorithm {

input: A continuous random variable X_k , time spent on edge i , which has a Gamma distribution with parameter $\lambda > 0$, if for $k = 1, 2, \dots, n$, and the probability of each branch

output: $q_{e,t}$

Search BEGIN

Source sends Search agent from an Ingress onto Control Network;

Take the C.P. nearest to the edge with highest probability $q_{e,t}$ as the Egress;

Find the edge with highest probability;

Search agent jump to a mobile node that is heading forward the target agent Messenger;

If found the target

End Search

Else delete this edge;

Next loop;

END Search

END PPM-M Algorithm

圖 4-8: 機率預測搜尋演算法-多重路徑(PPM-M)

第5章

效能評估

本章將以模擬方式評估控制點選擇演算法的效能，代理人位置預測法中所求機率之正確度預測搜尋法之效能。5.1 節為實驗環境之描述;5.2 節為控制點選擇問題提出的幾個啟發式演算法效能評估;5.3 節對所提出之機率預測搜尋法進行效能評估，判斷代理人位置預測之計算值與實驗值的差距，並評估機率預測搜尋法之搜尋效率容錯能力。

5.1 實驗設計

實驗一:控制點選擇問題效能評估

使用 JAVA 在一個人電腦上實作所提出的幾個控制點選擇問題的啟發式演算法，並隨機產生 Graph 進行測試。在小型實驗中，比較各演算法與最佳解之差距，而大型實驗中則因計算量太大，僅比較各演算法之間的效能。軟硬體規格為：

- 處理器：Pentium(R) Dual-Core CPU E5300 @2.60GHz
- 記憶體：4.00 GB RAM
- 作業系統：32位元MS Win 7 SP1
- 開發環境：Eclipse

表 5-1：實驗一控制點選擇問題實驗描述

Experiment	Experiment Profile
1A	Small scale evaluation
1B	Large scale evaluation: node sensitivity test
1C	Large scale evaluation: trail sensitivity test

實驗二:機率預測搜尋法效能評估

使用 MATLAB 在一個人電腦上評估所提出的機率預測是否準確，並依各子實驗參數隨機產生 Gamma 及 Normal 亂數進行準確度測試，比較依計算機率求出的計算值與實驗值差距及錯誤容忍度的測試，並評估機率預測搜尋法與二分搜尋法兩者效能差異。

軟硬體規格為：

- 處理器：Pentium(R) Dual-Core CPU E5300 @2.60GHz
- 記憶體：4.00 GB RAM
- 作業系統：32位元MS Win 7 SP1
- 開發環境：MATLAB Simulink

表 5-2：實驗二機率預測搜尋法實驗描述

Experiment	Experiment Profile
2A	Single path evaluation
2A-1	Accuracy test of probabilistic calculation
2A-2	Performance evaluation against Binary search
2A-3	Error tolerance test (1) Against Normal distribution (2) Against inaccurate Gamma distribution

2B	Multiple path evaluation
2B-1	Accuracy test of probabilistic calculation
2B-2	Performance evaluation against single path
2B-3	Error tolerance test (1) Against Normal distribution (2) Against inaccurate Gamma distribution

5.2 控制點選擇問題效能評估

5.2.1 小型問題效能評估

由於 CPSP-Coverage、CPSP-Utility 及 CPSP-Utility-Flow 皆為 NP-complete 問題，因此我們將利用第三章所提出的 NPF2、NPF-U2 與 NPF-PF 啟發式演算法來求得次佳解，與 NPF、NPF-U 及最佳解做比較，以分析各演算法與最佳解之效能差異。因最佳解所耗費的計算資源龐大，故此種效能比較只適合小型問題。

5.2.2 實驗 1A 環境參數設定

我們將隨機產生 5 個 Graph，參數設定如下：

- Total number of candidate control points = 20。
- Total number of trails = 8。
- Available resources = 12。
- Deployment cost of candidate control points = 1~5。
- Edge weight = 1~5。
- Trail flow rate = 1~5。

5.2.3 實驗 1A 結果與分析

圖 5-1 至 5-3 為 CPSP-Coverage 的實驗結果，分別列出 NPF、NPF2 演算法與 optimal solution 所求得的 edge coverage、mean MCD、max MCD，三者比較。

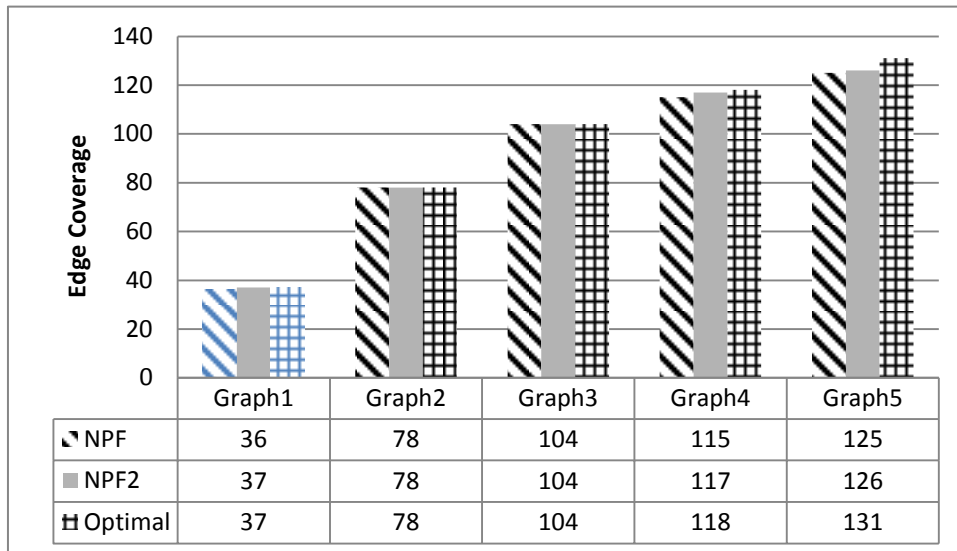


圖 5-1：CPSP-Coverage—NPF vs. NPF2 vs. Optimal solution (Edge Coverage)

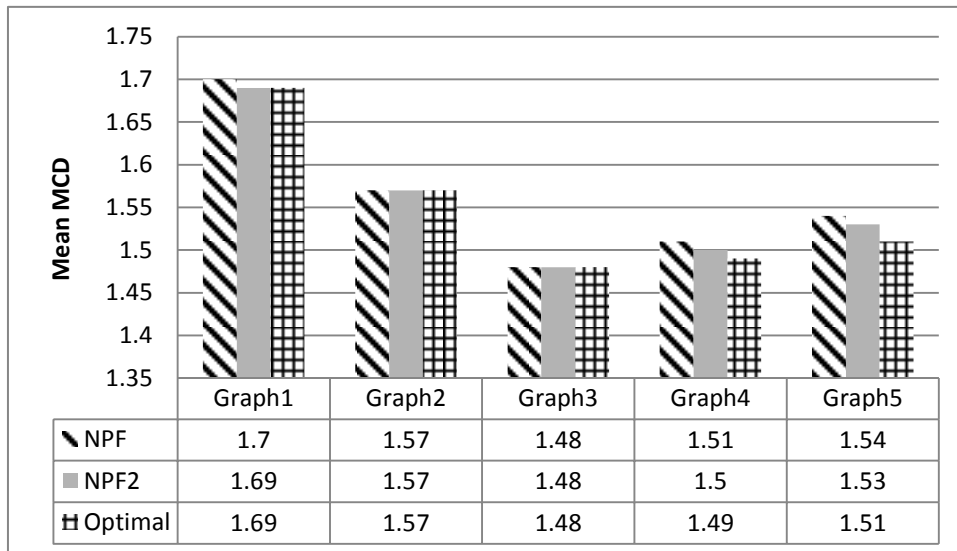


圖 5-2：CPSP-Coverage—NPF vs. NPF2 vs. Optimal solution (Mean MCD)

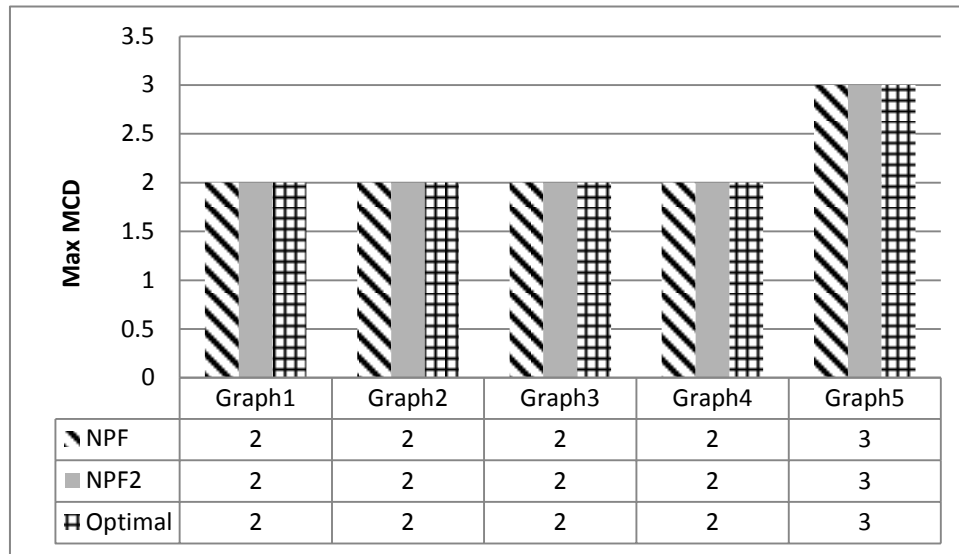


圖 5-3：CPSP-Coverage—NPF vs. NPF2 vs. Optimal solution
(Max MCD)

由圖 5-1 可知，在 Graph 1 中 NPF 演算法所求的 edge coverage 比 NPF2 及 optimal solution 均少一個 edge，而 NPF2 所求出的 edge 與 optimal solution 相同，在 Graph 4 中，NPF2 比 optimal solution 所求 edge 數少一個 edge，NPF 則又比 NPF2 少兩個 edge，在 Graph5 中，NPF2 比 optimal 少 5 個 edge，而 NPF 又比 NPF2 演算法所求的 edge 少一個 edge。可看出以 NPF2 選擇出的控制點在 edge coverage 表現較佳。且因為 NPF、NPF2 演算法會優先選擇 node potential 及 trail hit count 加權後較高的候選控制點，不能保證所得到的 profit 最高，因此，在某些情況下就會造成 NPF 及 NPF2 演算法的 edge coverage 效率低於 optimal solution。

由圖 5-2 可知，在 Graph 1 中 NPF 演算法所求的 mean MCD 比 optimal solution 高 0.01 個 edge，NPF2 演算法所求與 optimal solution 相同。在 Graph 4 中，NPF 比 optimal solution 高 0.02 個 edge，NPF2 比 optimal solution 高 0.01 個 edge。在 Graph 5 中，NPF 演算法所求的 mean MCD 比 optimal solution 高 0.03 個 edge，NPF2 演算法所求比 optimal solution 高 0.02 個 edge。在此 Graph 中，NPF2 所選的候選點比 NPF 所挑選的候選點

在所有 trail 上出現比例相對較高，故 NPF2 比 NPF 較容易找到距離行動節點較近的控制點，因此，在此情況下就會形成 NPF2 所求的 mean MCD 效率優於 NPF 演算法。而在 max MCD 三者數據均相同。

圖 5-4、圖 5-5 為 CPSP-Utility 的實驗結果，分別列出 NPF-U 演算法與 NPF-U2 演算法及 optimal solution 所求得的 weighted edge coverage、mean weighted MCD，而由於 max weighted MCD 所求相同，故不以圖表列出。

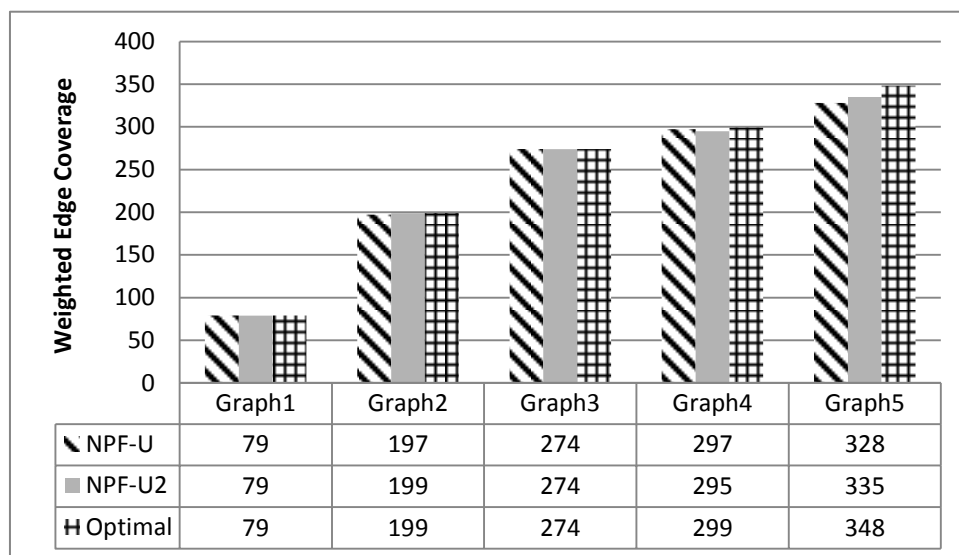


圖 5-4：CPSP-Utility—NPF-U vs. NPF-U2 vs. Optimal solution
(Weighted Edge Coverage)

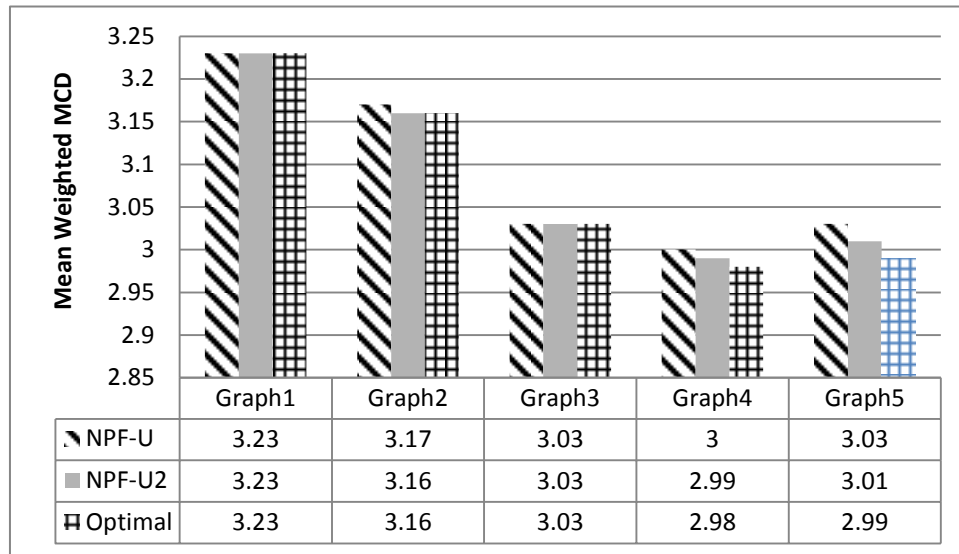


圖 5-5：CPSP-Utility—NPF-U vs. NPF-U2 vs. Optimal solution
(Mean Weighted MCD)

由圖 5-4 可知，在 Graph 2、Graph 4 及 Graph 5 中，NPF-U 演算法所求的 weighted edge coverage 分別比 optimal solution 少 2 個、2 個及 20 個 weighted edge，原因同上，也是因為 NPF-U 演算法的控制點選擇機制所導致。而在 Graph 4 及 Graph 5 中，NPF-U2 分別比 optimal solution 少 4 個及 13 個 weighted edge，使用 NPF-U2 演算法所挑選出來的點比 NPF-U 的 weighted edge 較多，這也是因為 NPF-U2 演算的設計原則挑選出來的候選點，在所有 trail 上出現的比例較 NPF-U 來的高。

而由圖 5-5 可知，在 Graph 2、4、5 中，NPF-U 演算法所求的 mean weighted MCD 比 optimal solution 多 0.01、0.02、0.04 個 weighted edge，這是由於在此 Graph 中，optimal solution 演算法所選的候選控制點在所有 trail 上出現的比例較 NPF-U 所選的來得高，故所求的 mean weighted MCD 效率低於 optimal solution。相同的，在 Graph 5 中 NPF-U2 演算法所求的 mean weighted MCD 比 optimal solution 多 0.01、0.02 個 mean weighted MCD，原因同 NPF。由圖 5-6 可看出，NPF-U 演算法的 mean weighted MCD 效能略低於 NPF-U2。

圖 5-6、圖 5-7 為 CPSP-Utility-Flow 的實驗結果，列出 NPF-PF 演算法與 optimal solution 所求得的 populated edge coverage、mean weighted MCD。

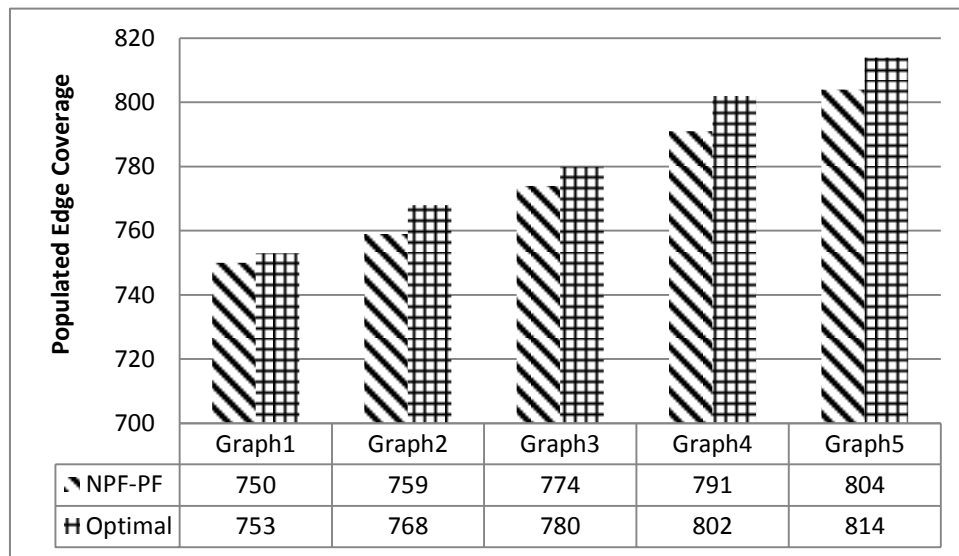


圖 5-6：CPSP-Utility-Flow—NPF-PF vs. Optimal solution (Populated Edge Coverage)

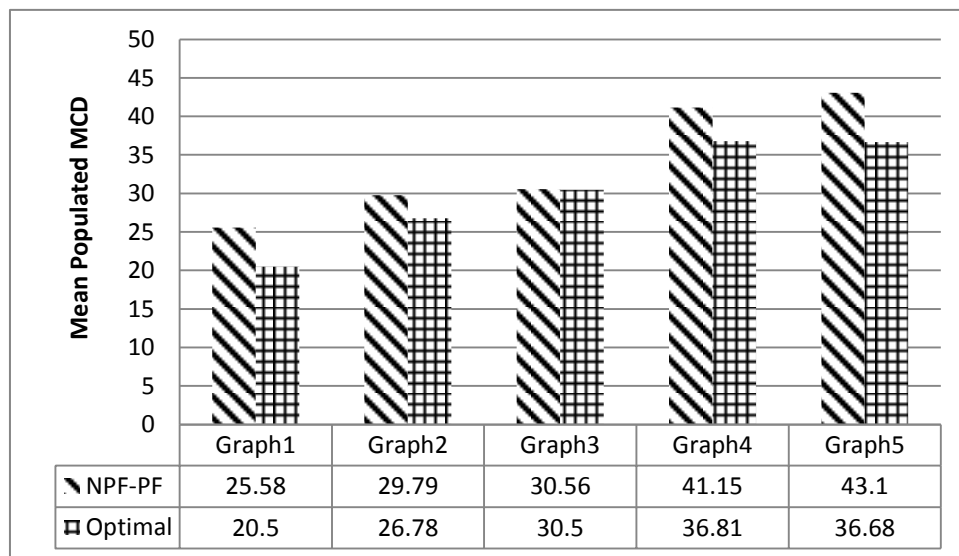


圖 5-7：CPSP-Utility-Flow—NPF-PF vs. Optimal solution (Mean Populated MCD)

由圖 5-6 可知，在 Graph 1、2、3、4 及 5 中 NPF-PF 演算法所求的 populated edge

coverage 比 optimal solution 各少 3、9、6、11 及 10 個 populated edge，因為 NPF-PF 演算法會優先選擇 unselected edge weight/node cost 加權 trail potential 再乘上 flow potential 較高的候選控制點，如同 0-1 knapsack problem，這樣做也不能保證所得到的 profit 最高，因此 NPF-PF 演算法的 populated edge coverage 效率低於 optimal solution。且每段路徑上的 flow 為 1~5，若遇到兩條 trail 相交則 flow 可能是 2~10，因此數據相差較其他評估指標較多。

由圖 5-7 可知，在 Graph 1、2、3、4 及 5 中，NPF-PF 演算法所求的 mean populated MCD 比 optimal solution 高 5.08、3.01、0.06、4.34 及 6.42 個 populated edge，且 mean populated MCD 需乘上在這個路段上的 flow，因此 mean populated MCD 相較於其他評估指標為高，下節將此演算法與其他演算法一同評估大型問題，可看出各啟發式演算法之效能比較。

5.2.4 大型問題效能評估

以下將模擬在不同 Graph 大小的情況下，NPF2、NPF-U2 及 NPF-PF 演算法之效率，並分析 node sensitivity、trail sensitivity 所造成的影響。由於 optimal solution 不易求得，故未與其比較。

我們所用的評估指標如下：

- Edge coverage
- Weighted edge coverage
- Populated edge coverage
- Mean MCD
- Mean weighted MCD
- Mean populated MCD

5.2.5 實驗 1B 環境參數設定

本實驗觀察節點數對演算法的影響，對 10~100 間大小不同的 Graph 各隨機產生 50 個 Graph，並以下列指標評估其效能：

- Total number of candidate control points = 10~100。
- Total number of trails = 8。
- Available resources = 12。
- Deployment cost of candidate control points = 1~5。
- Edge weight = 1~5。
- Trail flow rate = 1~10。

5.2.6 實驗 1B 結果與分析

由圖 5-8 可知，隨著候選控制點的增加，NPF2 演算法的 edge coverage 效率最高，NPF-U2 與 NPF-PF 兩者在這項評估上互有高低。由圖 5-9 得知，隨著候選控制點的增加，NPF-PF 演算法的 populated edge coverage 效率明顯高於其他兩者演算法許多，而 NPF2、NPF-U2 兩者表現差異不大。

由圖 5-10 可知，隨著點數的增加，NPF-U2 演算法的 mean weighted MCD 效率最好，NPF2 演算法次之，NPF-PF 的啟發式演算法最低。因 NPF-U2 演算法是設計挑選出最多 weighted edge coverage，也因此 mean weight MCD 的表現相較其他兩者為好。由圖 5-11 可知，隨著候選控制點的增加，原本在 edge coverage、mean weighted MCD 效率較低的 NPF-PF 的啟發式演算法，在 mean populated MCD 則明顯相較其他兩個演算法為佳，其他兩個演算法表現差異不大。

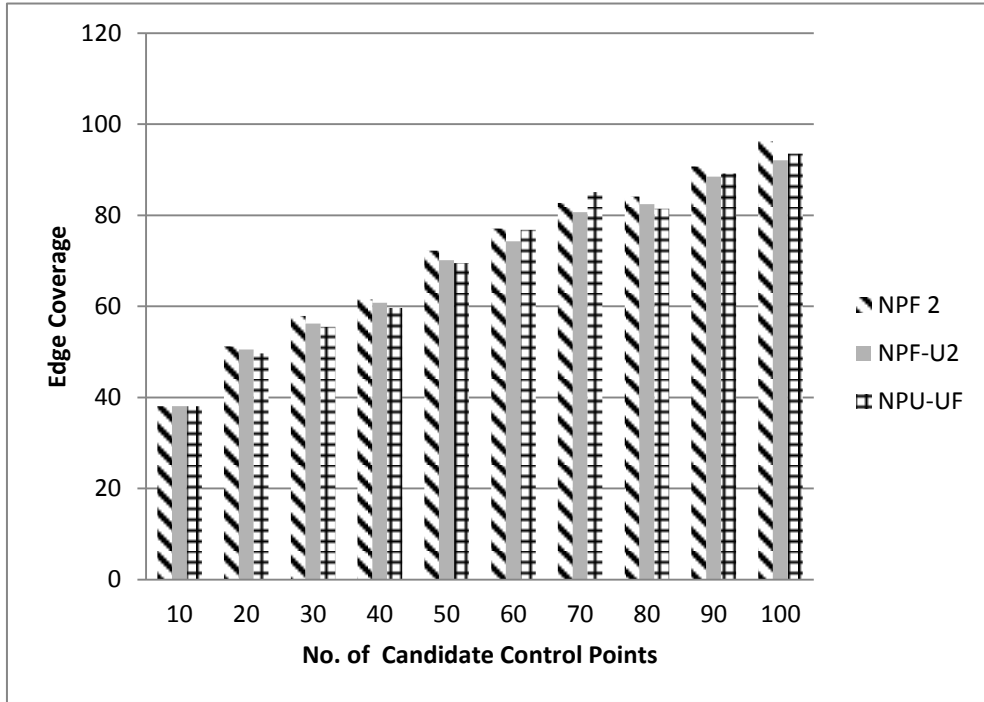


圖 5-8：各 NPF 演算法之效能比較 (Edge Coverage)
(Sensitivity of edge coverage on the number of nodes)

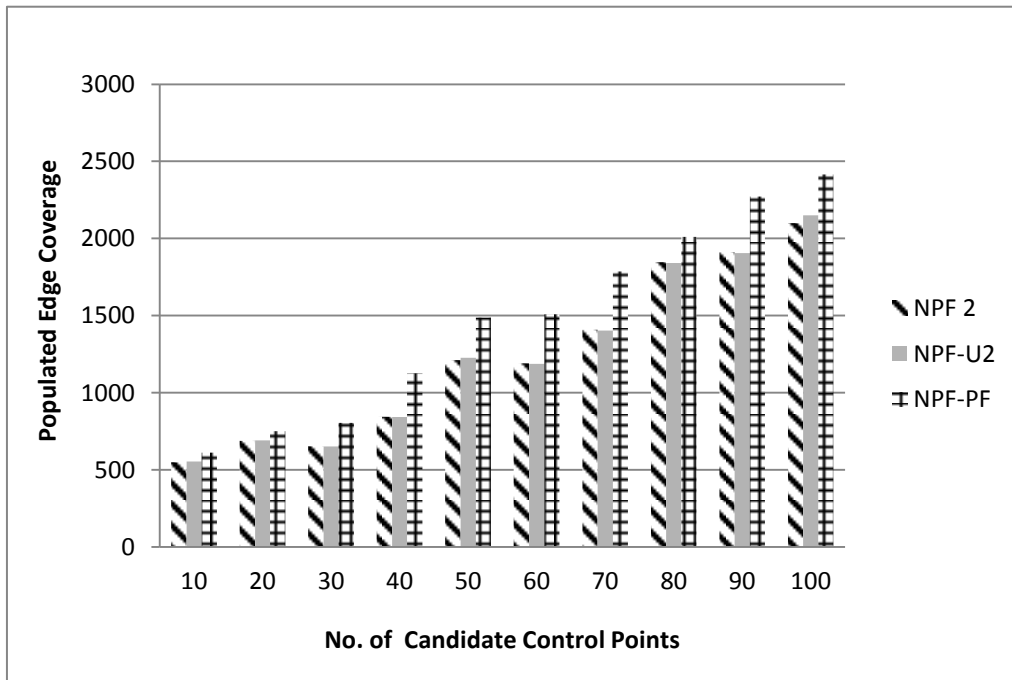


圖 5-9：各 NPF 演算法之效能比較 (Populated Edge Coverage)
(Sensitivity of populated edge coverage on the number of nodes)

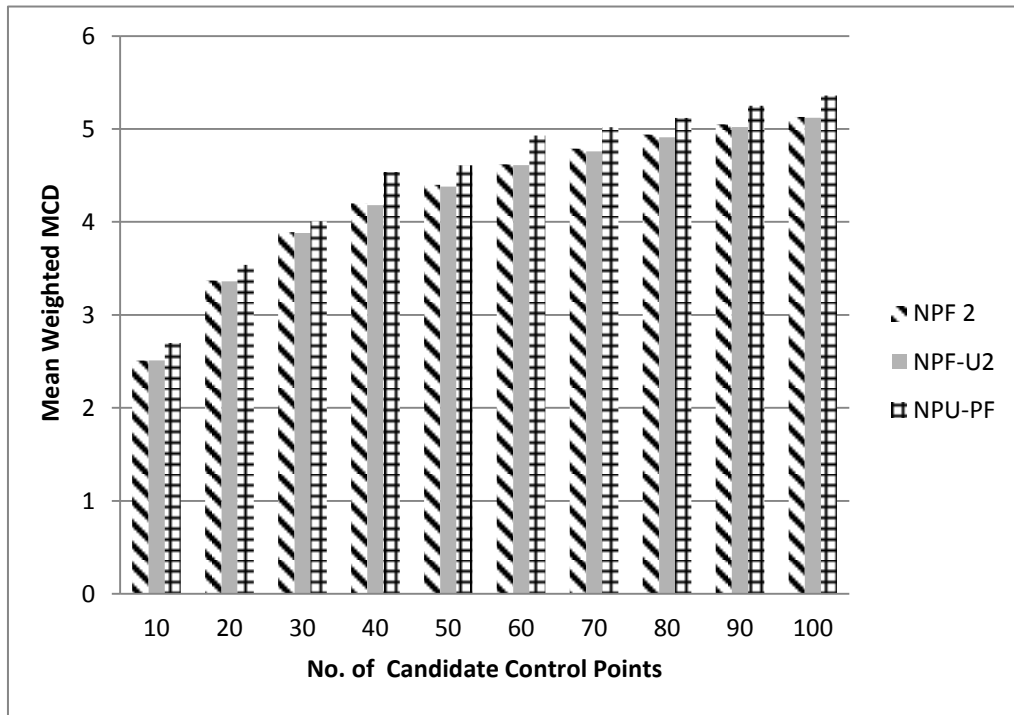


圖 5-10：各 NPF 演算法之效能比較 (Mean Weighted MCD)
(Sensitivity of mean weighted MCD on the number of nodes)

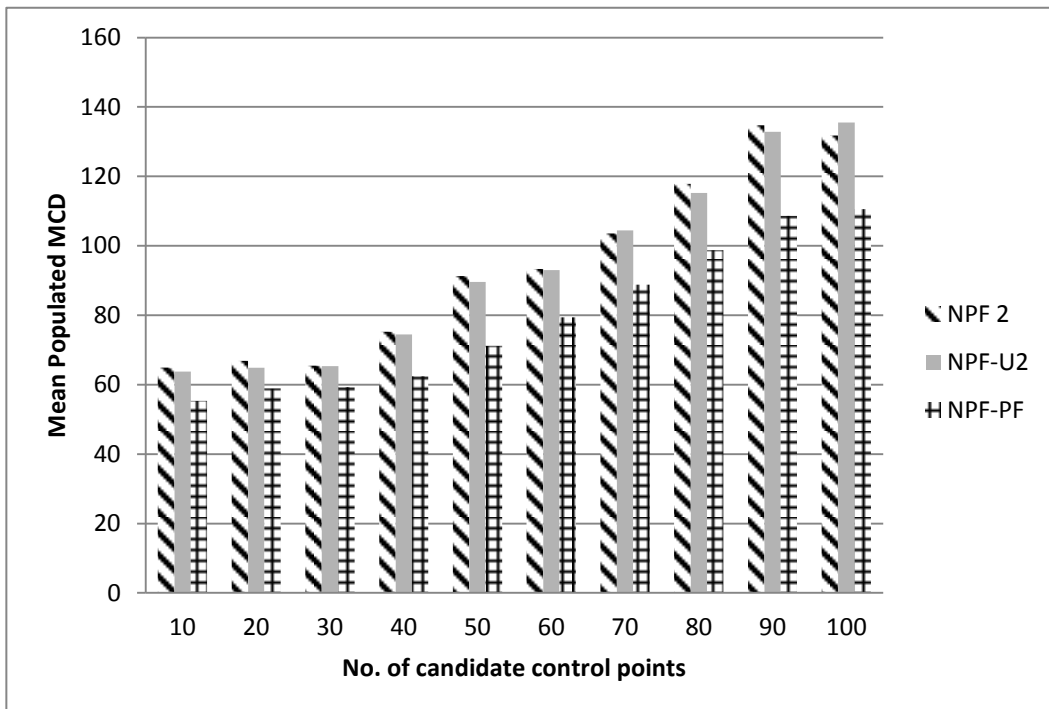


圖 5-11：各 NPF 演算法之效能比較 (Mean Populated MCD)
(Sensitivity of mean populated MCD on the number of nodes)

5.2.7 實驗 1C 環境參數設定

本實驗觀察登山路徑數對演算法的影響，對 4~40 間大小不同的 Graph 各隨機產生 50 個 Graph，並以下列指標評估其效能：

- Total number of candidate control points = 100。
- Total number of trails = 4~40。
- Available resources = 12。
- Deployment cost of candidate control points = 1~5。
- Edge weight = 1~5。
- Trail flow rate = 1~10。

5.2.8 實驗 1C 結果與分析

由圖 5-12 可知，隨著登山路徑數的增加，NPF-PF 啟發式演算法的 edge coverage 效率，會逐漸與 NPF2、NPF-U2 演算法接近，而 edge coverage 以 NPF-U2 效率最好，但與其他兩者差異不大。而由圖 5-13 可知，隨著登山路徑數的增加，NPF-PF 演算法的 populated edge coverage 效率最高，且效率明顯領先優於其他兩者演算法。

由圖 5-14 可知，隨著登山路徑數的增加，NPF-U2 演算法在 weighted MCD 效率最高，NPF-PF 演算法最低，但 NPF-PF 與 NPF2 兩者互有領先兩者差異不大。這三個啟發式演算法所求得的 mean weighted MCD 會愈來愈相近。在 trail sensitivity 下，推測可能是因為登山路徑數增加了，故交點多的地方，很容易就找到較好的候選控制點。在圖 5-15 可知，NPF-PF 其演算法在 mean populated MCD 效率高於其他兩者，而 NPF2 及 NPF-U2 兩者 mean populated MCD 效率差距不大。

根據以上描述可知，當必須考慮某些路段的長短及重要性時，NPF-U2 的涵蓋效率最高，故較建議使用 CPSP-Utility 模型來建立控制網路，以保證所有在路徑上行走

的登山客，都可以很快地接近控制點。而當必須考慮某些路段的長短、重要性及路段上的人數流量時，CPSP-Utility-Folw 模型則是較好的選擇，讓 agent 到最近控制點的平均距離最短，使得在路徑上行走的登山客，都可以很快地接近控制點。且我們針對此模型提出的啟發式演算法 NPF-PF，在 populated MCD 評估指標上明顯優於 NPF2 及 NPF-U 演算法，而在其他估評指標上，表現則互有高低。

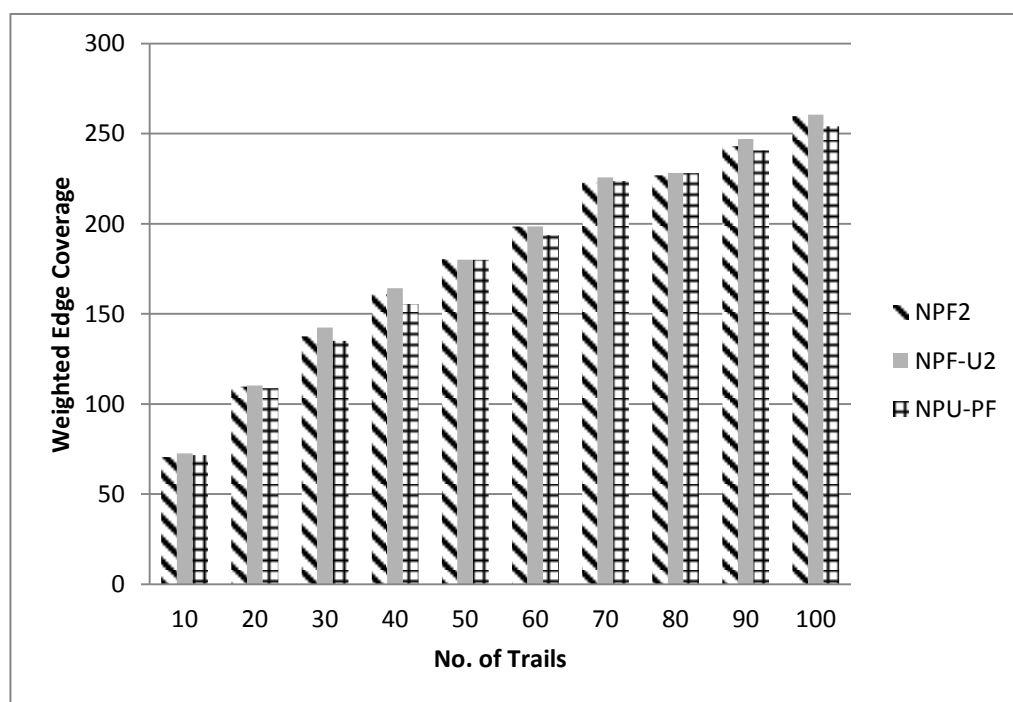


圖 5-12：各 NPF 演算法之效能比較 (Weighted Edge Coverage)
(Sensitivity of weighted edge coverage on the number of trails)

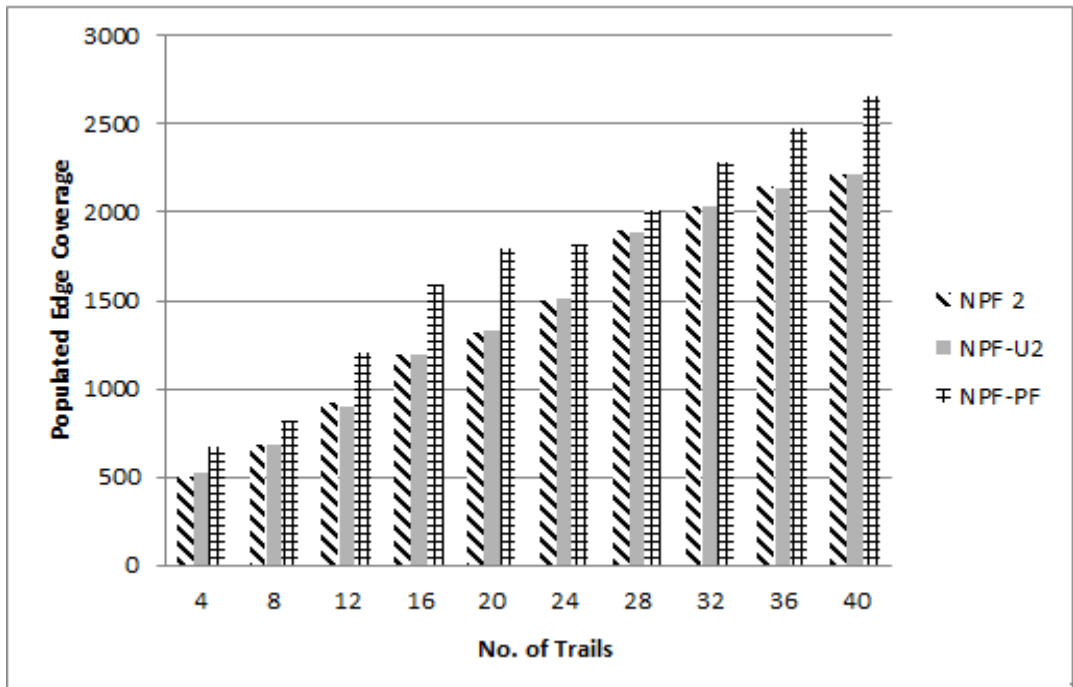


圖 5-13：各 NPF 演算法之效能比較 (Populated Edge Coverage)
(Sensitivity of populated edge coverage on the number of trails)

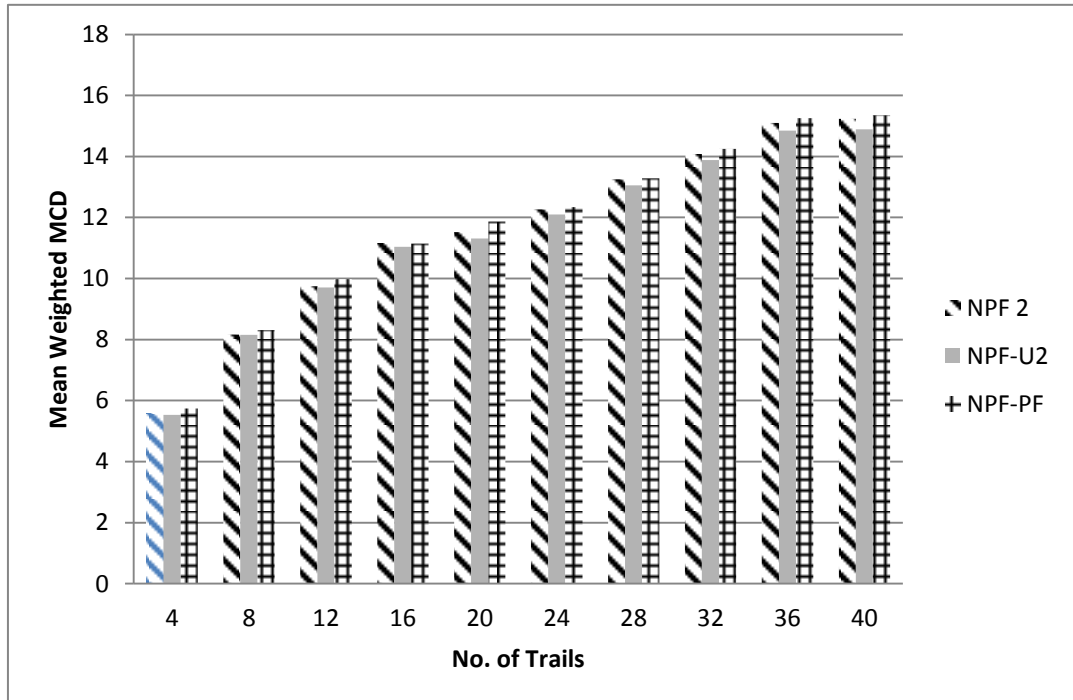


圖 5-14：各 NPF 演算法之效能比較 (Mean Weighted MCD)
(Sensitivity of mean weighted MCD on the number of trails)

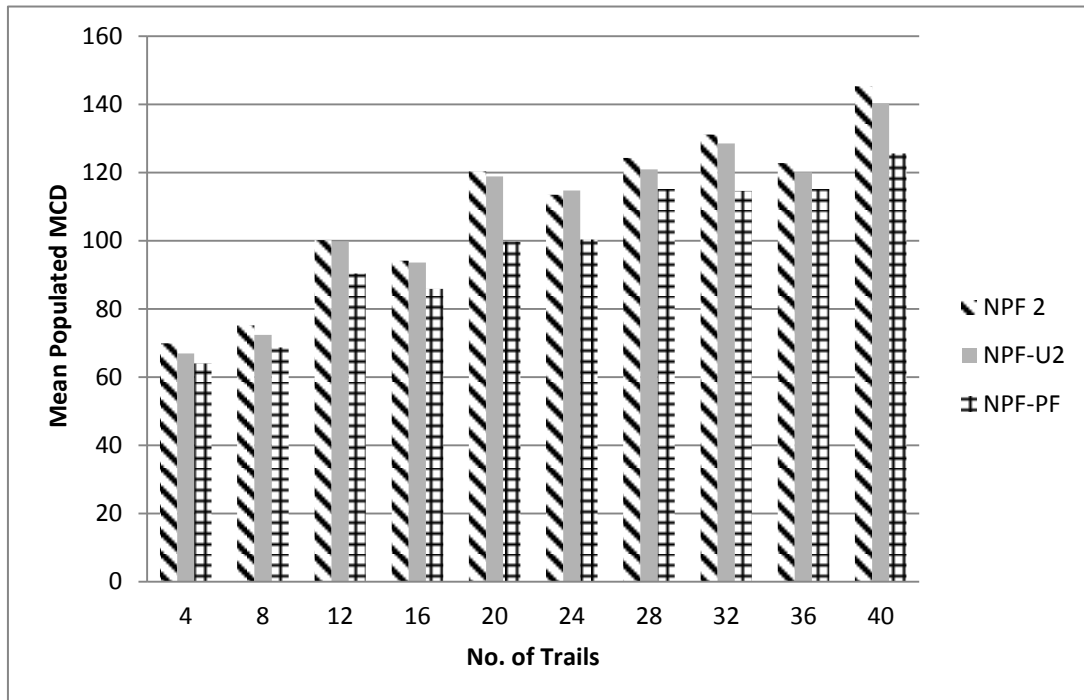


圖 5-15：各 NPF 演算法之效能比較 (Mean Populated MCD)
(Sensitivity of mean populated MCD on the number of trails)

5.3 機率預測搜尋法效能評估

此節開始，對第四章提出之機率預測搜尋演算法，隨機產生大量模擬亂數，評估預測準確度、搜尋法效能及當預測有誤時錯誤容忍度。為了分析實驗 2 機率計算值與實驗統計值的差異及機率預測搜尋法-單一路徑 (PPM-S)、機率預測搜尋法-多重路徑 (PPM-M) 與二分搜尋法演算法效能差異程度，定義以下評估指標：

- Probability Calculation Error (%) : (實驗值 - 計算值) / 實驗值 (For 公式(1),(2))。
- Number of Search Probes: 平均搜尋次數。
- Normalized Search performance : number of edges in the graph / 平均搜尋次數 (Number of Search Probes)。
- Performance Deviation (%): 兩個搜尋法搜尋次數之差距 / 二分搜尋法平均搜尋次數。

- Search Performance Ratio: 二分搜尋法平均搜尋次數 / PPM-S 平均搜尋次數 (Number of Search Probes) 。

實驗 2A 為 single path evaluation，子實驗 2A-1 目的為評估所提出的計算機率準確度;子實驗 2A-2 為機率預測搜尋法-單一路徑(PPM-S)與二分搜尋法兩者間效能評估;子實驗 2A-3 對所提出的計算機率做錯誤容忍度實驗。實驗 2B 為 multiple path evaluation，子實驗 2B-1 目的為評估在有叉路情況下，所提出的計算機率準確度;子實驗 2B-2 為機率預測搜尋法-多重路徑(PPM-M)效能評估;子實驗 2B-3 對其做錯誤容忍度實驗。

5.3.1 單一路徑機率預測準確度實驗

5.3.1.1 實驗 2A-1 環境參數設定

表 5-3：實驗 2A-1 參數

Experimental Parameters	Range of values
Number of edges on a path	100
Gamma distribution	shape parameter = 2、3、4、5 scale parameter =2

實驗 2A-1 目的為評估所提出的計算機率公式(1)的準確度，實驗依照實驗值參數隨機產生 100 條路段組成一條 path，每條路段花費時間的機率分佈假設為一 Gamma 分佈，其中 shape 參數設定為 2、3、4、5，scale 參數固定為 2，因此各路段花費時間的分佈形狀類似，將此參數代入本研究提出的計算機率預測求得計算值。並利用亂數產生 10000 個 agent 及其各路段所花時間，求得統計值，兩者互相比較。下圖 5-16 至 5-19 列出 Gamma 分佈 shape 參數為 2、3、4、5 及 scale 參數固定為 2 之 Probability Density Function 圖形，當 Gamma 分佈參數值落於此區間，更貼近實際健行者走訪山區路段所花費的時間分佈行為。

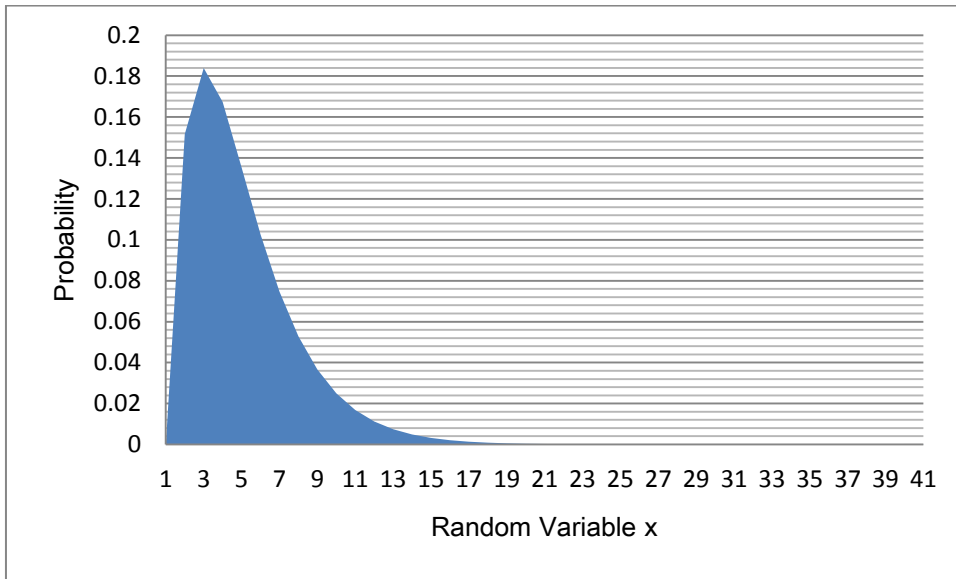


圖 5-16 : Gamma Distribution PDF (shape=2,scale=2)

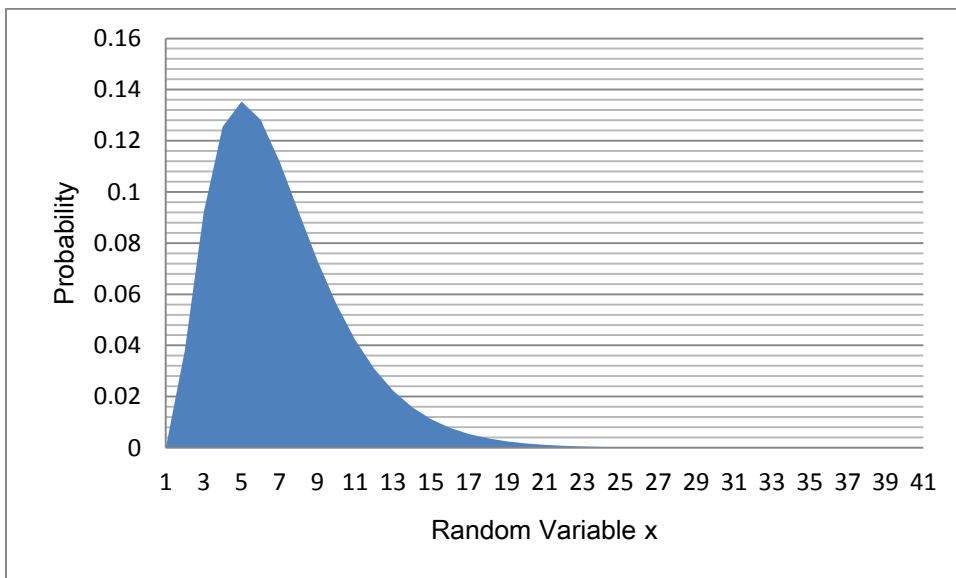


圖 5-17 : Gamma Distribution PDF (shape=3,scale=2)

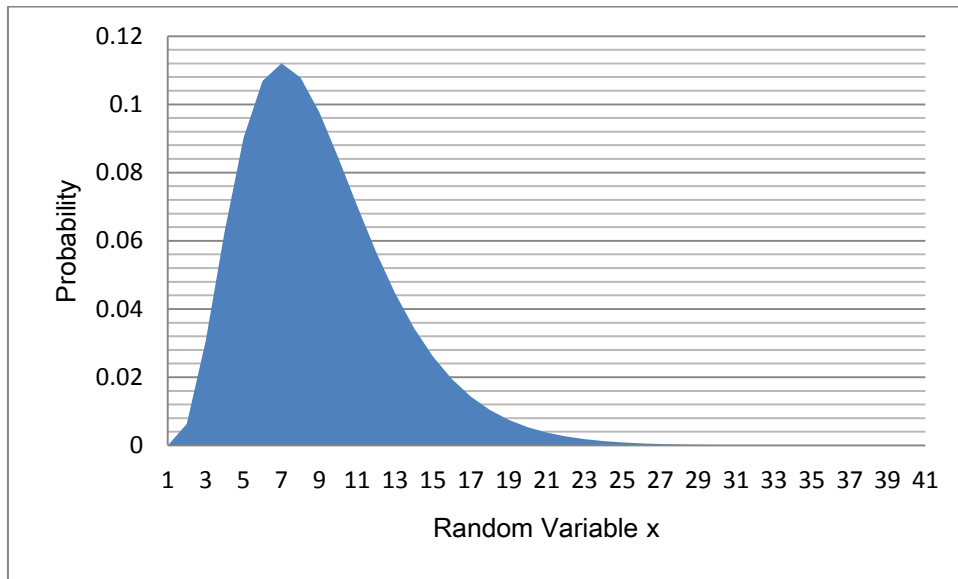


圖 5-18 : Gamma Distribution PDF(shape=4,scale=2)

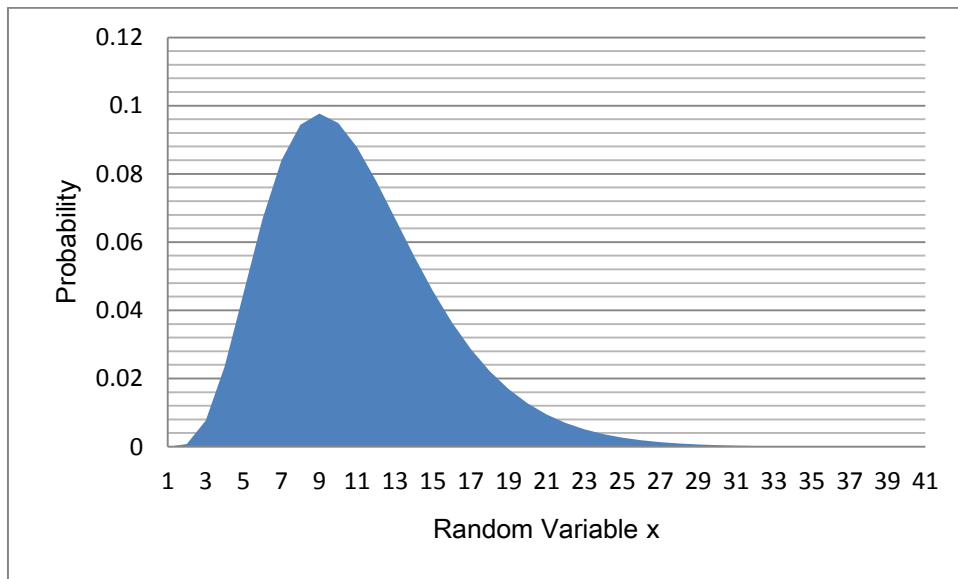


圖 5-19 : Gamma Distribution PDF (shape=5,scale=2)

5.3.1.2 實驗 2A-1 結果與分析

依照表 5-3 參數取出之亂數，整段 path 所花費時間平均加總為 389.46 分鐘，最大值為

434.19 分鐘，因此我們預測的時間點，取出發後時間 400 分鐘內，間隔為 10 分鐘做一次預測準確度評估。

在離出發時間較近的時候，mobile agent 大多落於前幾段，公式 (1) 之誤差較小。當時間增長，且各路段健行花費時間差異很大的時候，落於各段的分佈較廣，誤差逐漸增加，這是合理的。當 mobile agent 隨著時間前進不斷往前方路段移動，某一健行者在時間 t 時，位於各路段的機率，由公式(1)所計算的機率與本實驗所得到的統計值之間的差距，由圖 5-20 至 5-23 呈現，計算值與實驗值在出發後 400 分鐘內，兩者平均偏差最小為 2.45% 最大為 4.84%。為簡化敘述，圖中以 Error 表示 Probability Calculation Error。

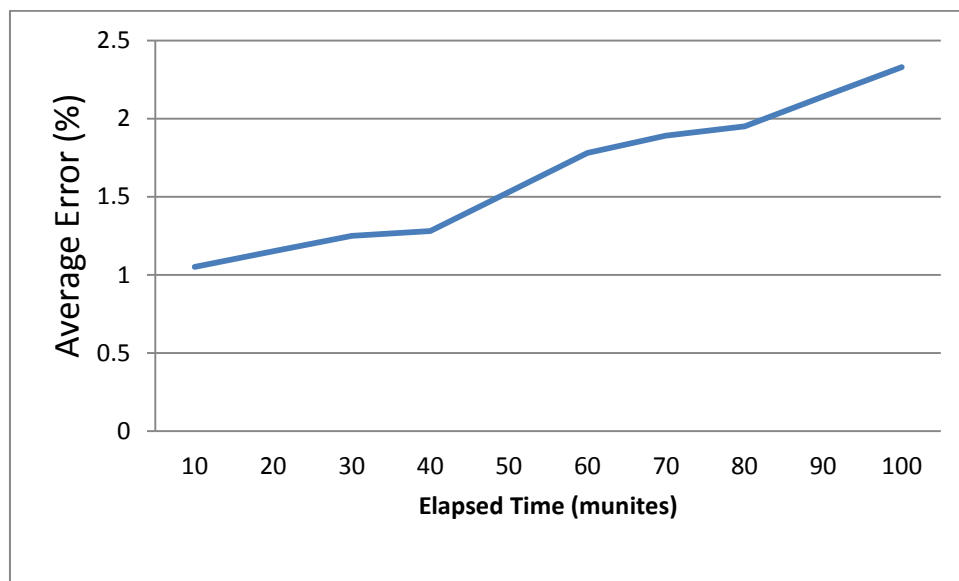


圖 5-20:實驗 2A-1 出發後 10-100 分鐘
計算值與實驗值平均偏差(預測準確度實驗)

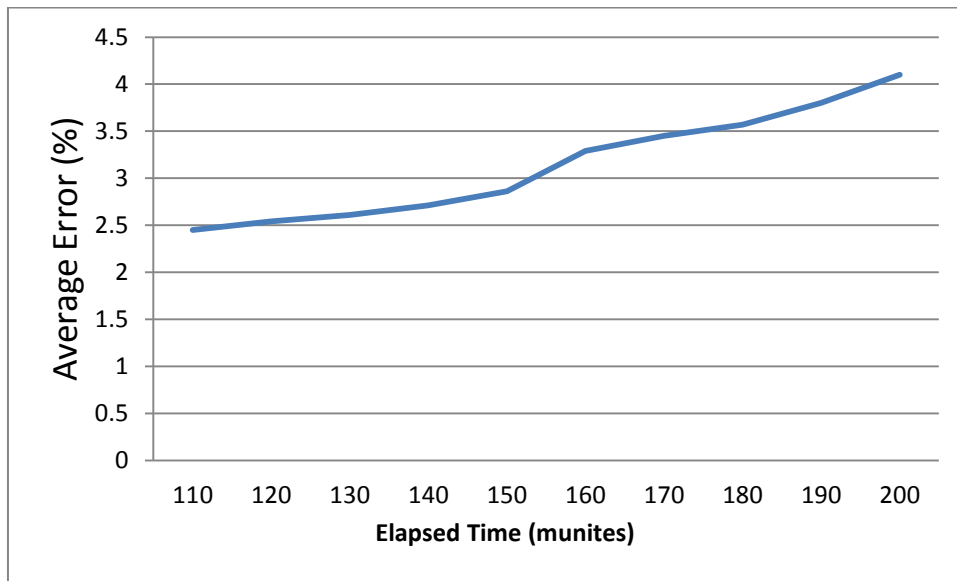


圖 5-21: 實驗 2A-1，出發後 110-200 分鐘
計算值與實驗值平均偏差(預測準確度實驗)

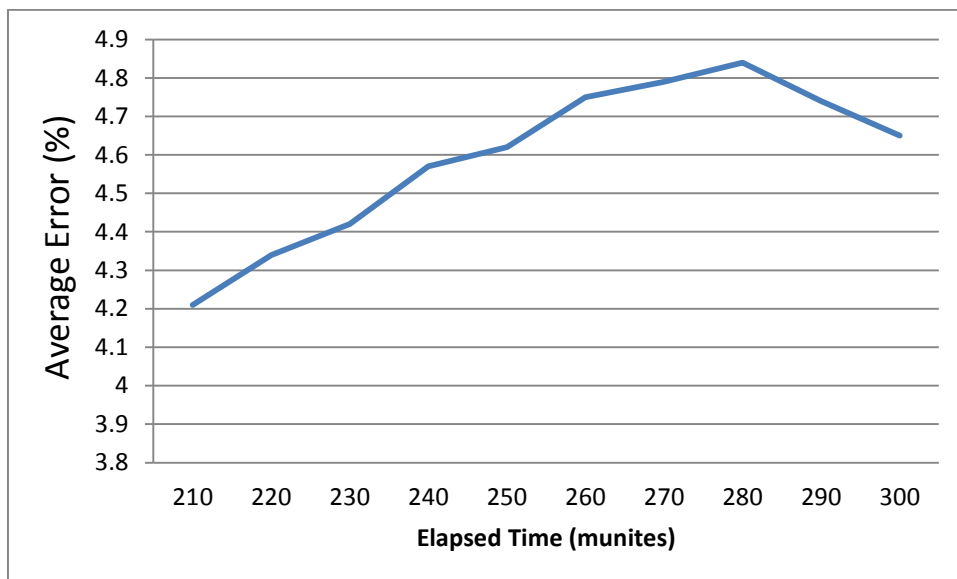


圖 5-22: 實驗 2A-1，出發後 210-300 分鐘
計算值與實驗值平均偏差(預測準確度實驗)

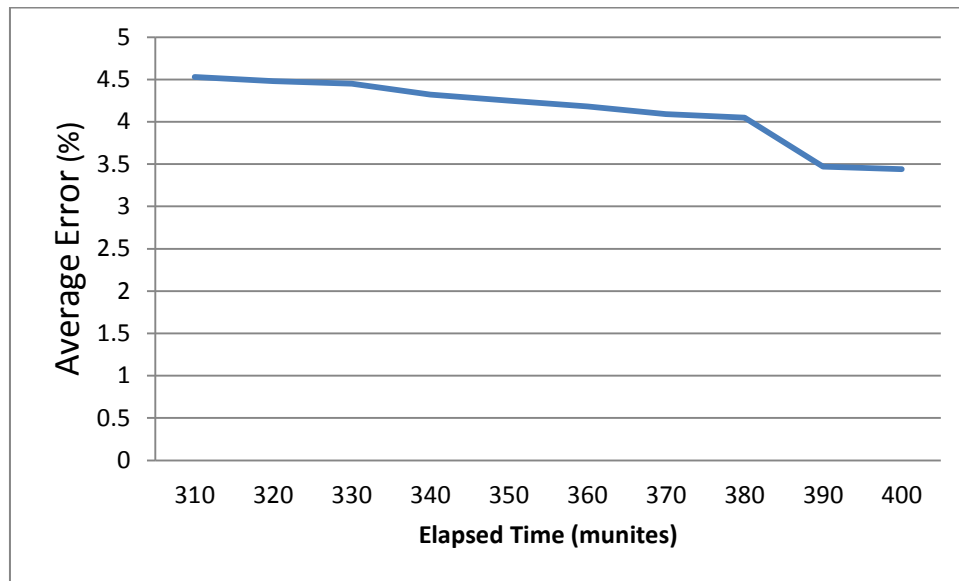


圖 5-23: 實驗 2A-1，出發後 310-400 分鐘
計算值與實驗值平均偏差(預測準確度實驗)

表 5-4:實驗 2A-1 預測準確度實驗

Elapsed Time(Minutes)	10	20	30	40	50	60	70	80	90	100
Probability Calculation Error (%)	2.45	2.54	2.61	2.71	2.86	3.29	3.45	3.57	3.8	4.1
Elapsed Time(Minutes)	110	120	130	140	150	160	170	180	190	200
Probability Calculation Error (%)	4.21	4.34	4.42	4.57	4.62	4.75	4.79	4.84	4.74	4.65
Elapsed Time(Minutes)	210	220	230	240	250	260	270	280	290	300
Probability Calculation Error (%)	4.53	4.48	4.45	4.32	4.25	4.18	4.09	4.05	3.47	3.44
Elapsed Time(Minutes)	310	320	330	340	350	360	370	380	390	400
Probability Calculation Error (%)	4.53	4.48	4.45	4.32	4.25	4.18	4.09	4.05	3.47	3.44

5.3.2 單一路徑機率預測搜尋法效能評估

5.3.2.1 實驗 2A-2 環境參數設定

表 5-5：實驗 2A-2 參數

Parameters	Range of values
Number of edge on a path	100
Gamma distribution	shape parameter = 2、3、4、5 scale parameter =2

實驗 2A-2 使用表 5-5 參數產生測試案例，對同一組測試案例進行實驗，比較使用機率預測搜尋法-單一路徑 (PPM-S) 及二分搜尋法的平均搜尋次數 (number of search probes)，自出發後 10 分鐘起，每 10 分鐘搜尋 (search) 一次，直到 400 分鐘為止，實驗共執行 10000 回合。

5.3.2.2 實驗 2A-2 結果與分析

由下圖 5-24 至圖 5-28 呈現，出發後 10 至 400 分鐘內，機率預測搜尋法-單一路徑 (PPM-S) 的平均搜尋次數 (number of search probes) 介於 1.95 至 2.84 間，平均為 2.25 次。二分搜尋法的平均搜尋次數 (number of search probes) 介於 6.09 至 6.87 間，平均為 6.52 次。因機率預測搜尋法-單一路徑 (PPM-S)，是從公式 (1) 計算出機率較高的路段開始搜尋，當 target agent 不在此路段上，若此節點已通過此路段，去掉已走訪過路段，若此節點尚未走訪過此路段，則將此路段前方路段去掉，再從目前機率最高的路段再次搜尋。相較於二分搜尋法每回合均從路徑中間路段開始搜尋 (search)，機率預測搜尋法-單一路徑 (PPM-S) 可較快逼近 mobile agent 所在路段，因此在各時間點上平均搜

尋次數 (number of search probes) 均比二分搜尋法少。機率預測搜尋法-單一路徑 (PPM-S) 與二分搜尋法 Performance Deviation 最小為 55.86% 最大為 81.61%。由實驗 2A-2 可得知，使用機率預測搜尋法-單一路徑 (PPM-S) 的搜尋效率較二分搜尋法效能較高。機率預測搜尋法-單一路徑 (PPM-S) 的 Normalized Search performance 為 44.44。

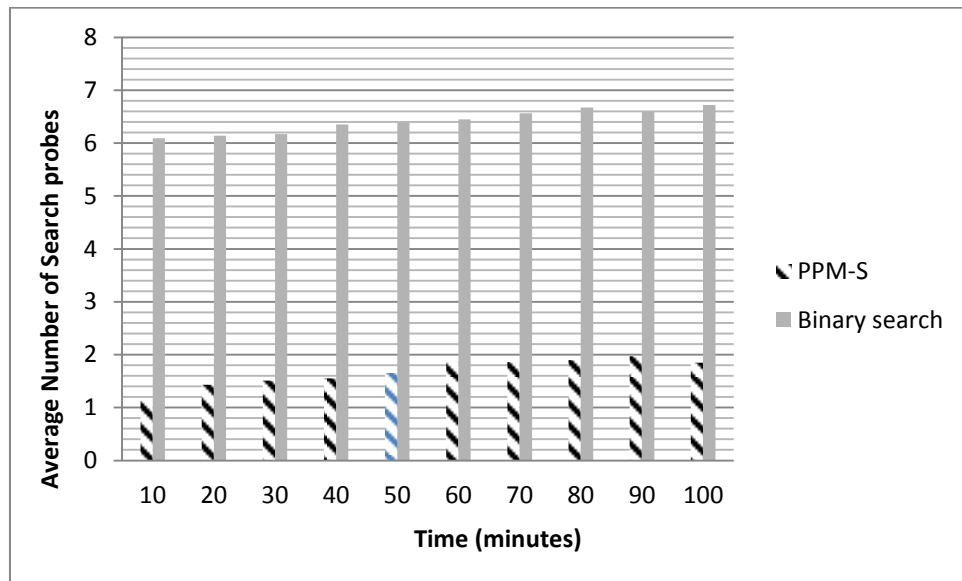


圖 5-24: 實驗 2A-2，出發後 10-100 分鐘 (搜尋法效能評估實驗)

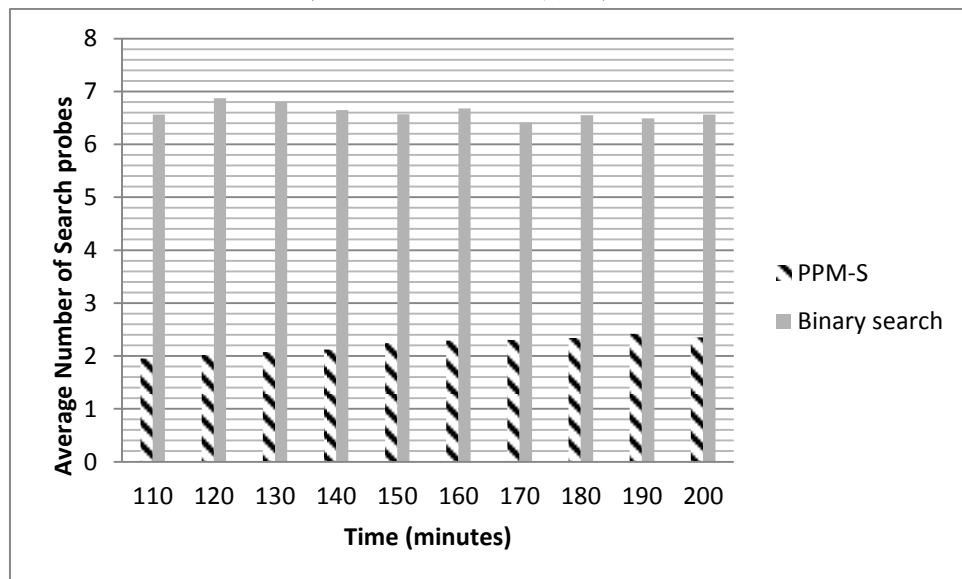


圖 5-25: 實驗 2A-2，出發後 110-200 分鐘 (搜尋法效能評估實驗)

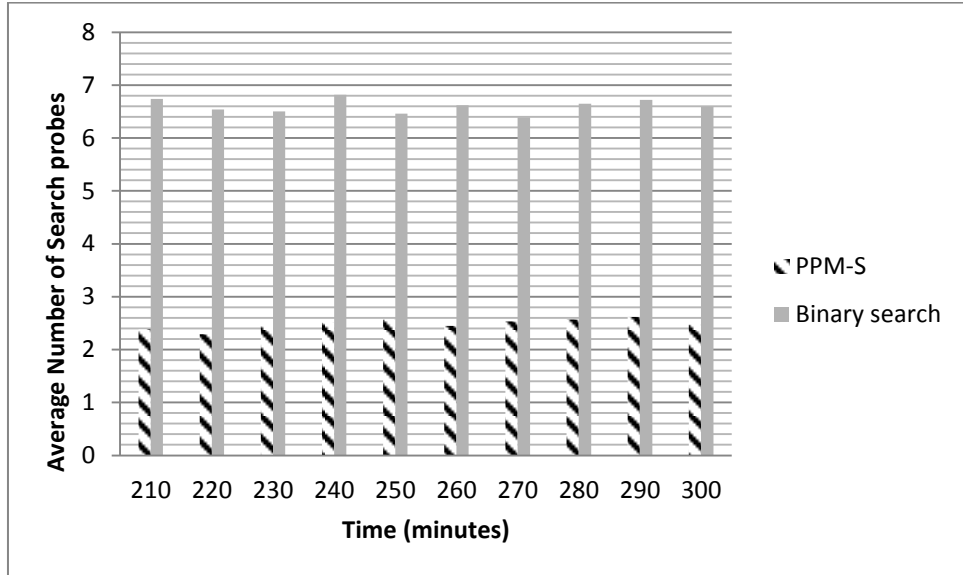


圖 5-26: 實驗 2A-2，出發後 210-300 分鐘
(搜尋法效能評估實驗)

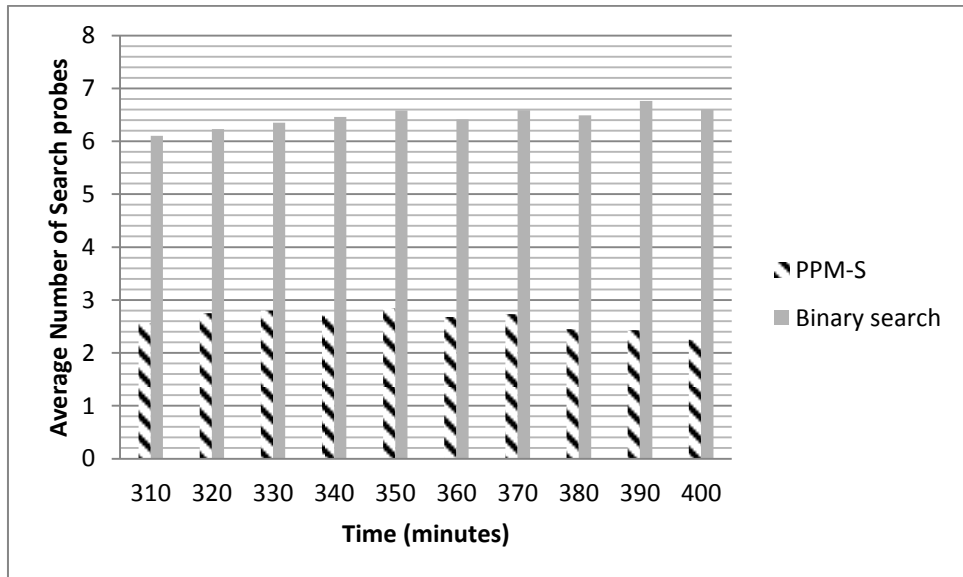


圖 5-27: 實驗 2A-2，出發後 310-400 分鐘
(搜尋法效能評估實驗)

表 5-6 實驗 2A-2 搜尋法效能評估實驗

Elapsed Time (Minutes)	10	20	30	40	50	60	70	80	90	100
PPM-S	1.12	1.43	1.51	1.55	1.64	1.87	1.86	1.9	1.97	1.85
Binary Search	6.09	6.14	6.17	6.35	6.39	6.45	6.56	6.67	6.59	6.72
Performance Deviation(%)	81.61	76.71	75.53	75.59	74.33	71.01	71.65	71.51	70.11	72.47
Elapsed Time (Minutes)	110	120	130	140	150	160	170	180	190	200
PPM-S	1.95	2.02	2.07	2.12	2.24	2.29	2.3	2.34	2.42	2.35
Binary Search	6.56	6.87	6.81	6.65	6.57	6.68	6.41	6.55	6.49	6.56
Performance Deviation(%)	70.27	70.60	69.60	68.12	65.91	65.72	64.12	64.27	62.71	64.18
Elapsed Time (Minutes)	210	220	230	240	250	260	270	280	290	300
PPM-S	2.38	2.29	2.44	2.51	2.57	2.45	2.53	2.57	2.62	2.47
Binary Search	6.74	6.54	6.5	6.82	6.46	6.62	6.39	6.65	6.72	6.59
Performance Deviation(%)	64.69	64.98	62.46	63.20	60.22	62.99	60.41	61.35	61.01	62.52
Elapsed Time (Minutes)	310	320	330	340	350	360	370	380	390	400
PPM-S	2.56	2.75	2.8	2.71	2.84	2.68	2.73	2.45	2.43	2.25
Binary Search	6.1	6.23	6.35	6.46	6.58	6.41	6.59	6.49	6.76	6.6
Performance Deviation(%)	58.03	55.86	55.91	58.05	56.84	58.19	58.57	62.25	64.05	65.91

5.3.3 單一路徑錯誤容忍度實驗

5.3.3.1 實驗 2A-3 環境參數設定

表 5-7：實驗 2A-3-1 參數

Parameters	Range of values
Number of edge on path	100
Gamma distribution (by Calculation)	shape parameter = 2、3、4、5 scale parameter =2
Normal distribution (by Experiment)	location parameter = 2、3、4、5 scale parameter =2

表 5-8：實驗 2A-3-2 參數

Parameters	Range of values
Number of edge on path	100
Gamma distribution (by Calculation)	shape parameter = 2、3、4、5 scale parameter =2
Gamma distribution (by Calculation)	shape parameter = 3、4、5、6 scale parameter =2

實驗 2A-3 目的為評估所提出的計算機率公式 (1) 對不同機率分佈及參數做錯誤容忍準確度實驗。每回合實驗依照實驗值參數隨機產生 100 條路段組成一條 path，計算值將每條路段花費時間的機率分佈 Gamma 分佈參數 shape 設定在 2、3、4、5，scale 參數固定為 2，將此參數代入本研究提出的計算機率預測求得計算值。實驗 2A-3-1，測

試案例參數設定各路段花費時間的機率分佈為 Normal 分佈，location 參數為 2、3、4、5，scale 參數固定為 2；實驗 2A-3-2，測試案例參數設定各路段花費時間的機率分佈為 Gamma 分佈，shape 參數設定在 3、4、5、6，scale 參數固定為 2。實驗 2A-3-1 及 2A-3-2 均依各實驗參數亂數產生 10000 個 agent 及其各路段所花時間，求得統計值，與計算值互相比較。接著進行錯誤容忍度實驗，使用表 5-7、5-8 實驗值參數各自產生測試案例，對同一組測試案例進行實驗，比較使用機率預測搜尋法-單一路徑 (PPM-S) 及二分搜尋法的平均搜尋次數 (number of search probes)，自出發後 10 分鐘起，每 10 分鐘搜尋 (search) 一次，直到 400 分鐘為止，實驗共執行 10000 回合。

5.3.3.2 實驗 2A-3 結果與分析

實驗 2A-3 兩組實驗使用同一組計算值預測 mobile agent 在出發後 400 分鐘內，每 10 分鐘間隔 target agent 在各路段上的機率，做錯誤容忍度實驗。實驗 2A-3-1，依照 Normal 參數 location= 2、3、4、5，scale = 2 亂數產生的實驗值會相較參數相同的 Gamma 亂數變異較小，因此在離出發時間較近的時候，平均偏差較少，也就是大部分的 mobile agent 大多落在前幾段路段上，公式(1)之誤差較小。當時間增長，各路段健行時間差異逐漸增大的時候，落於各段的分佈較廣，誤差逐漸增加。當 mobile agent 隨著時間前進不斷往前方路段移動，某一健行者在時間 t 時，位於各路段的機率，由公式 (1) 所計算的機率與本實驗所得到的統計值之間的差距，由表 5-9 呈現，兩者平均 Error 最小為 6.16%最大為 15.87%。實驗 2A-3-2，依照 Gamma 參數 shape= 3、4、5、6，scale = 2 亂數產生，評估公式(1)在 Gamma 分佈其 shape 參數相差 1 時，預測的損害程度。由表 5-10 得知，兩者平均 Error 最小為 2.51%最大為 9.57%。將 2A-1、2A-3-1 與 2A-3-2 實驗之平均 Error 列於圖 5-40 至 5-43，可得知利用機率預測公式 (1) 於各實驗之預測準確度結果，以實驗 2A-1 較準確，平均 Error 在 4.84%之內，實驗 2A-3-2 次之，平均 Error 在 9.57%之內，實驗 2A-3-1 準確度最差，平均 Error 在 15.87%之內。

在 Normal distribution 下，搜尋效能損害評估，由下表 5-11 得知，出發後 10 至

400 分鐘內，機率預測搜尋法-單一路徑(PPM-S)的平均搜尋次數 (number of search probes) 介於 2.35 至 3.73 間，平均為 3.08 次，二分搜尋法的平均搜尋次數 (number of search probes) 介於 6.04 至 6.84 間，平均為 6.47 次。在各時間點上 PPM-S 的平均搜尋次數 (number of search probes) 均比二分搜尋法來的少。機率預測搜尋法-單一路徑(PPM-S)與二分搜尋法 Performance Deviation 最小為 40.83%最大為 62.67%。機率預測搜尋法-單一路徑(PPM-S)的 Normalized Search Performance 為 32.47。Search Performance Ratio (original) - Search Performance Ratio (Vs. Normal) = 0.8。在錯誤的 Gamma distribution 下之效能損害評估，由下表 5-12 得知，出發後 10 至 400 分鐘內，機率預測搜尋法-單一路徑(PPM-S)的平均搜尋次數 (number of search probes) 介於 1.52 至 3.25 間，平均為 2.66 次，二分搜尋法的平均搜尋次數 (number of search probes) 介於 6.14 至 6.94 間，平均為 6.38 次。機率預測搜尋法-單一路徑(PPM-S)與二分搜尋法 Performance Deviation 最小為 49.85%最大為 75.24%。機率預測搜尋法-單一路徑(PPM-S)的 Normalized Search Performance 為 37.59，Search Performance Ratio (original) - Search Performance Ratio (Vs. Gamma) = 0.5。將 2A-1、2A-3-1 與 2A-3-2 實驗之 Search Performance Ratio 列於圖 5-40 至 5-43，可得知利用機率預測搜尋法相較於二分搜尋法，以實驗 2A-1 較準確，實驗 2A-3-2 次之，實驗 2A-3-1 最差，使用 PPM-S 對測試案例進行錯誤容忍度實驗，即使各路段花費時間的機率分佈的假設有誤時，搜尋效能仍高於二分搜尋法約 2 倍。

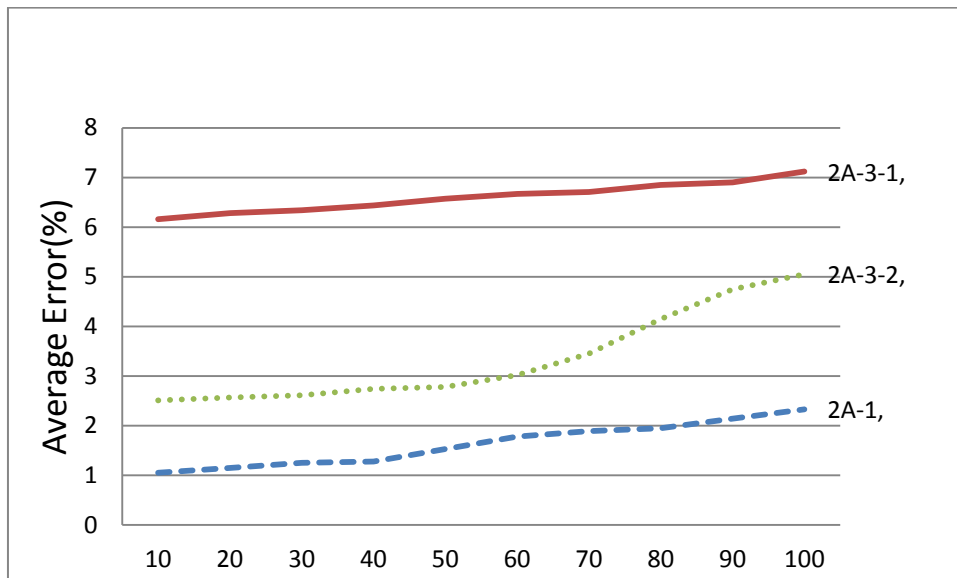


圖 5-28: 2A 各實驗之平均偏差比較
出發後 10-100 分鐘

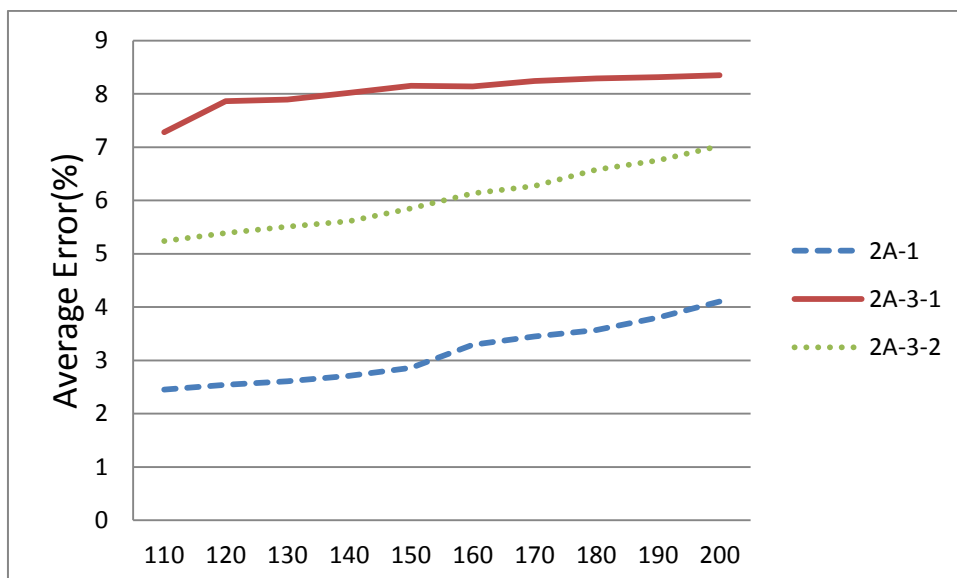


圖 5-29: 2A 各實驗之平均偏差比較
出發後 110-200 分鐘

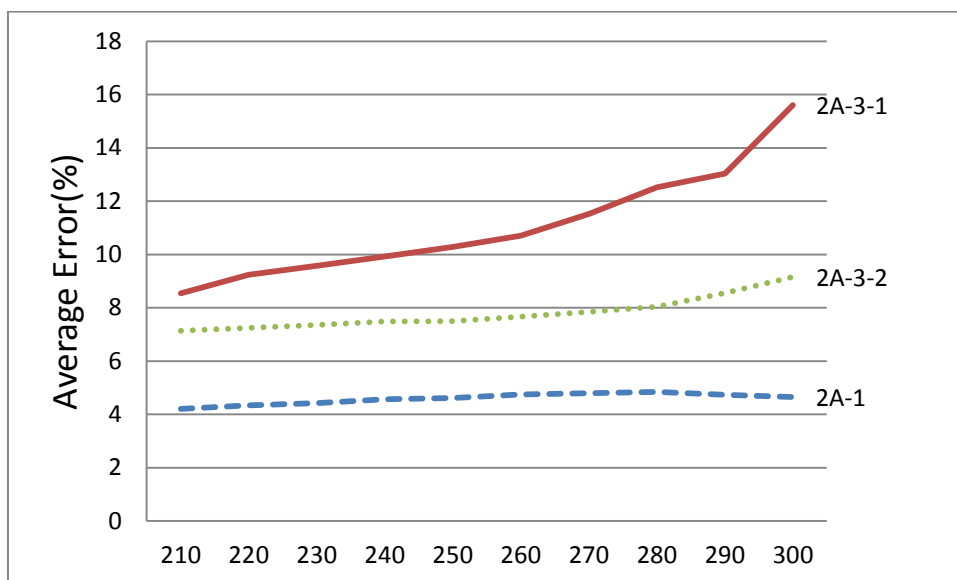


圖 5-30: 2A 各實驗之平均偏差比較
出發後 210-300 分鐘

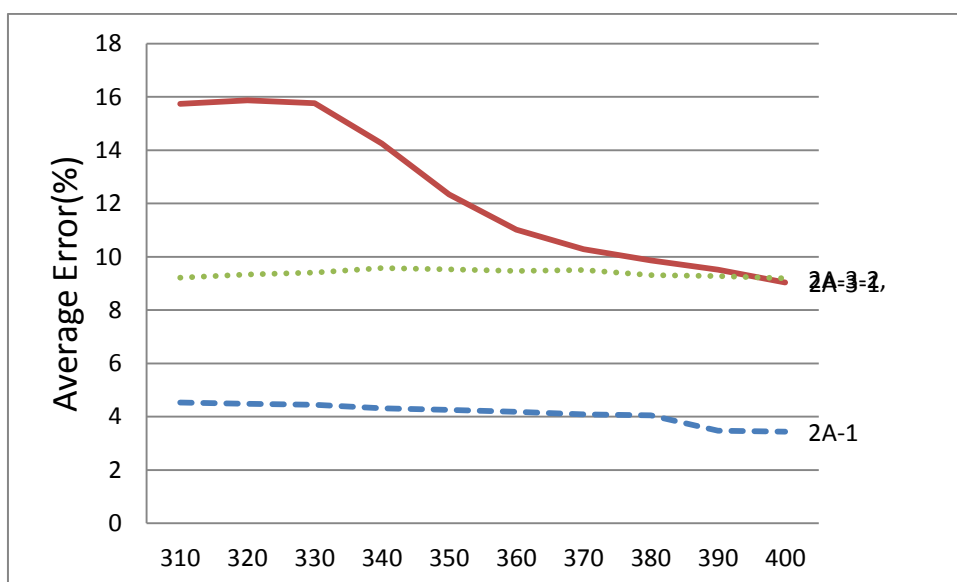


圖 5-31: 2A 各實驗之平均偏差比較
出發後 310-400 分鐘

表 5-9: 實驗 2A-3-1 預測準確度實驗

Elapsed Time(Minutes)	10	20	30	40	50	60	70	80	90	100
Probability Calculation Error (%)	6.16	6.28	6.34	6.44	6.57	6.67	6.71	6.85	6.9	7.12
Elapsed Time(Minutes)	110	120	130	140	150	160	170	180	190	200
Probability Calculation Error (%)	7.28	7.86	7.89	8.02	8.15	8.14	8.24	8.29	8.31	8.35
Elapsed Time(Minutes)	210	220	230	240	250	260	270	280	290	300
Probability Calculation Error (%)	8.54	9.24	9.58	9.92	10.28	10.7	11.52	12.52	13.04	15.61
Elapsed Time(Minutes)	310	320	330	340	350	360	370	380	390	400
Probability Calculation Error (%)	15.74	15.87	15.76	14.25	12.34	11.02	10.28	9.86	9.51	9.04

表 5-10:實驗 2A-3-2 預測準確度實驗

Elapsed Time(Minutes)	10	20	30	40	50	60	70	80	90	100
Probability Calculation Error (%)	2.51	2.57	2.61	2.74	2.78	3.02	3.45	4.15	4.75	5.05
Elapsed Time(Minutes)	110	120	130	140	150	160	170	180	190	200
Probability Calculation Error (%)	5.24	5.39	5.51	5.61	5.85	6.13	6.27	6.58	6.75	7.02
Elapsed Time(Minutes)	210	220	230	240	250	260	270	280	290	300
Probability Calculation Error (%)	7.14	7.25	7.35	7.49	7.5	7.67	7.84	8.04	8.56	9.15
Elapsed Time(Minutes)	310	320	330	340	350	360	370	380	390	400
Probability Calculation Error (%)	9.21	9.34	9.41	9.57	9.53	9.47	9.5	9.31	9.28	9.19

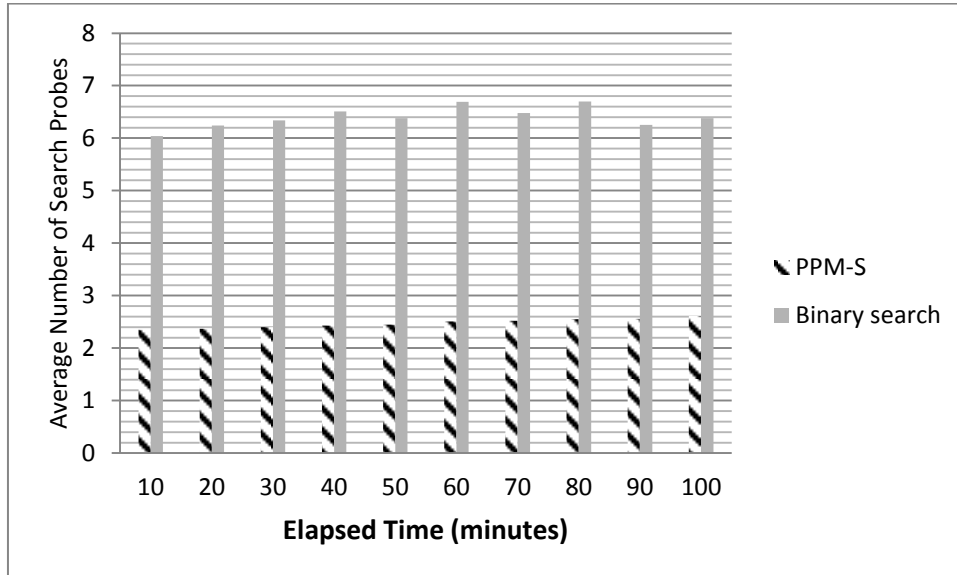


圖 5-32: 實驗 2A-3-1，出發後 10-100 分鐘
(錯誤容忍度實驗，Against Normal distribution)

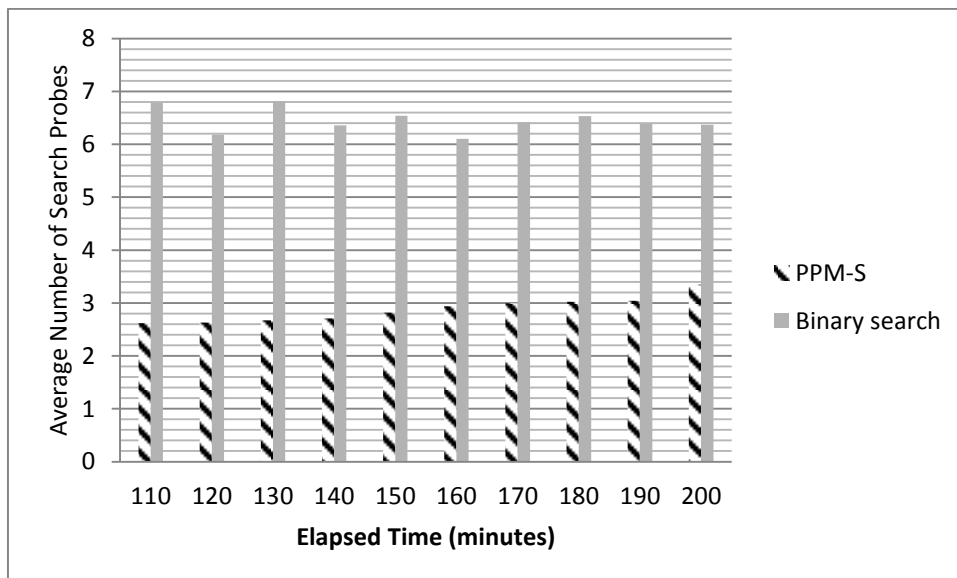


圖 5-33: 實驗 2A-3-1，出發後 110-200 分鐘
(錯誤容忍度實驗，Against Normal distribution)

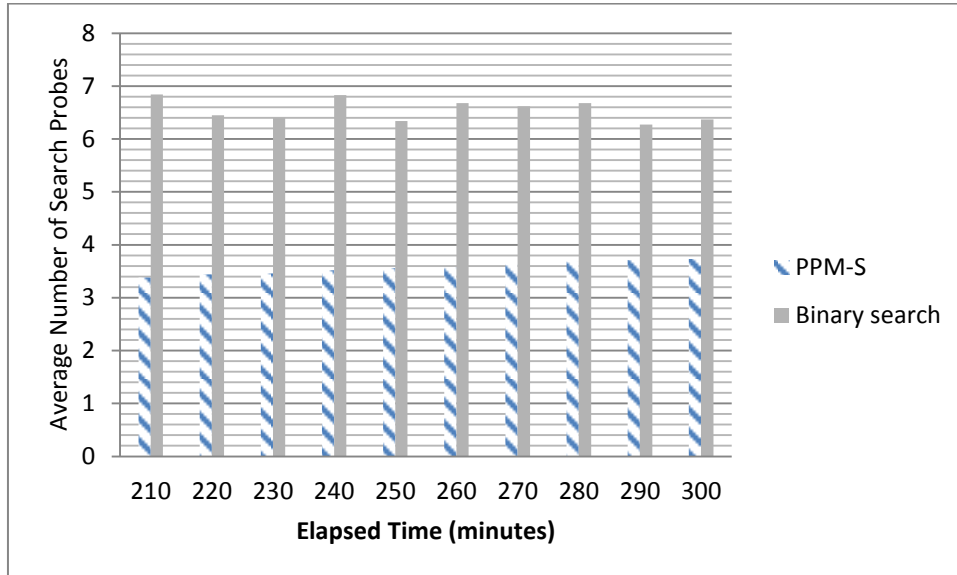


圖 5-34: 實驗 2A-3-1, 出發後 210-300 分鐘
(錯誤容忍度實驗, Against Normal distribution)

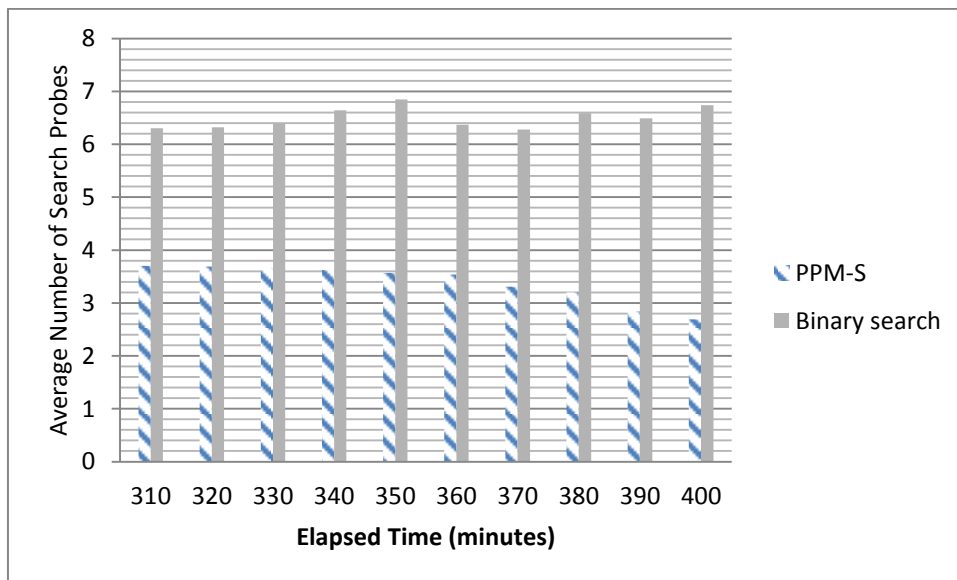


圖 5-35: 實驗 2A-3-1, 出發後 310-400 分鐘
(錯誤容忍度實驗, Against Normal distribution)

表 5-11: PPM-S 錯誤容忍度實驗 (Against Normal distribution)

Elapsed Time (Minutes)	10	20	30	40	50	60	70	80	90	100
PPM-S	2.35	2.37	2.4	2.43	2.45	2.51	2.52	2.55	2.56	2.61
Binary Search	6.04	6.24	6.34	6.51	6.38	6.69	6.48	6.7	6.25	6.38
Performance Deviation(%)	61.09	62.02	62.15	62.67	61.60	62.48	61.11	61.94	59.04	59.09
Elapsed Time (Minutes)	110	120	130	140	150	160	170	180	190	200
PPM-S	2.62	2.63	2.67	2.71	2.82	2.94	2.99	3.02	3.04	3.35
Binary Search	6.78	6.18	6.82	6.36	6.54	6.1	6.42	6.53	6.39	6.37
Performance Deviation(%)	61.36	57.44	60.85	57.39	56.88	51.80	53.43	53.75	52.43	47.41
Elapsed Time (Minutes)	210	220	230	240	250	260	270	280	290	300
PPM-S	3.38	3.44	3.46	3.52	3.56	3.58	3.63	3.68	3.71	3.73
Binary Search	6.84	6.45	6.4	6.83	6.34	6.68	6.62	6.68	6.27	6.37
Performance Deviation(%)	50.58	46.67	45.94	48.46	43.85	46.41	45.17	44.91	40.83	41.44
Elapsed Time (Minutes)	310	320	330	340	350	360	370	380	390	400
PPM-S	3.7	3.69	3.65	3.62	3.57	3.54	3.31	3.2	2.84	2.69
Binary Search	6.3	6.32	6.39	6.64	6.85	6.37	6.28	6.59	6.49	6.74
Performance Deviation(%)	41.27	41.61	42.88	45.48	47.88	44.43	47.29	51.44	56.24	60.09

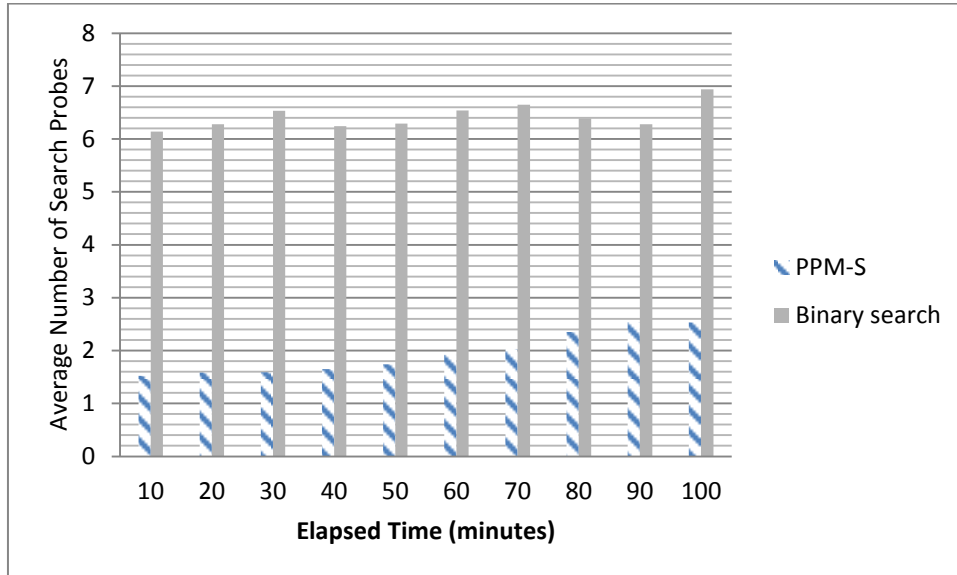


圖 5-36: 實驗 2A-3-2，出發後 10-100 分鐘
(錯誤容忍度實驗，Against inaccurate Gamma distribution)

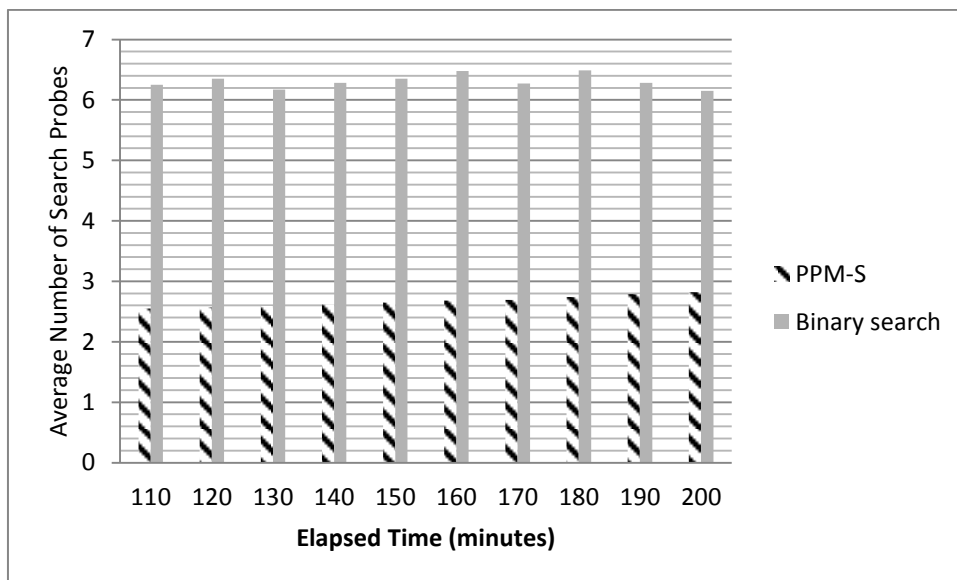


圖 5-37: 實驗 2A-3-2，出發後 110-200 分鐘
(錯誤容忍度實驗，Against inaccurate Gamma distribution)

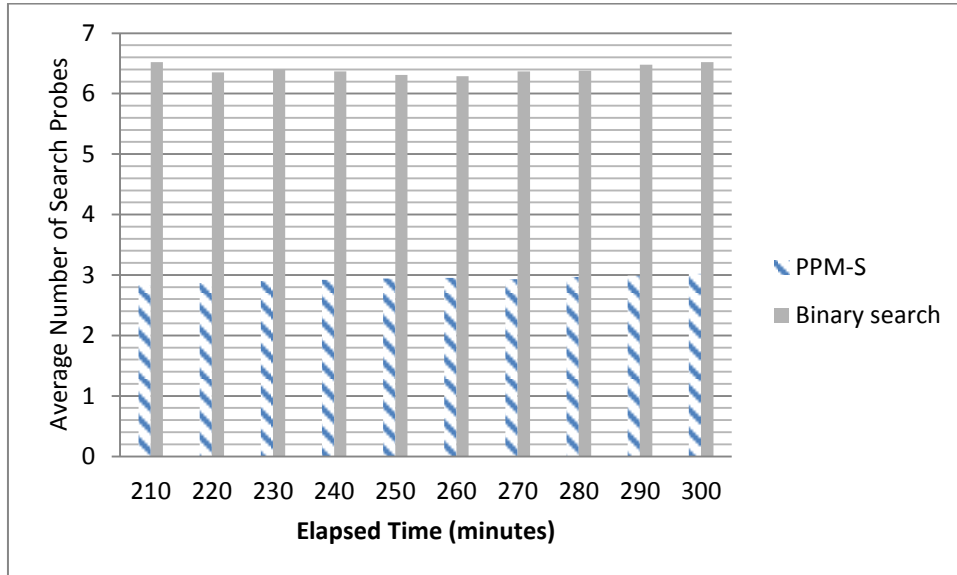


圖 5-38: 實驗 2A-3-2, 出發後 210-300 分鐘
(錯誤容忍度實驗, Against inaccurate Gamma distribution)

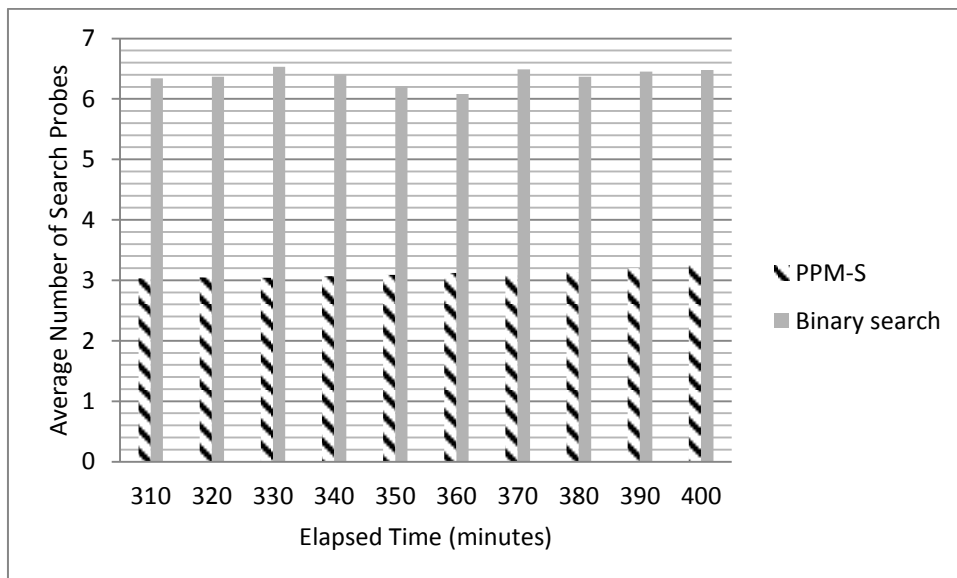


圖 5-39: 實驗 2A-3-2, 出發後 310-400 分鐘
(錯誤容忍度實驗, Against inaccurate Gamma distribution)

表 5-12: PPM-S 錯誤容忍度實驗 (Against inaccurate Gamma distribution)

Elapsed Time (Minutes)	10	20	30	40	50	60	70	80	90	100
PPM-S	1.52	1.58	1.59	1.65	1.74	1.92	2.03	2.35	2.54	2.53
Binary Search	6.14	6.28	6.53	6.24	6.29	6.54	6.65	6.39	6.28	6.94
Performance Deviation(%)	75.24	74.84	75.65	73.56	72.34	70.64	69.47	63.22	59.55	63.54
Elapsed Time (Minutes)	110	120	130	140	150	160	170	180	190	200
PPM-S	2.55	2.57	2.59	2.62	2.65	2.68	2.69	2.74	2.79	2.82
Binary Search	6.25	6.35	6.17	6.28	6.35	6.48	6.27	6.49	6.28	6.15
Performance Deviation(%)	59.20	59.53	58.02	58.28	58.27	58.64	57.10	57.78	55.57	54.15
Elapsed Time (Minutes)	210	220	230	240	250	260	270	280	290	300
PPM-S	2.83	2.87	2.9	2.92	2.94	2.95	2.93	2.97	2.99	3.02
Binary Search	6.52	6.35	6.4	6.37	6.31	6.29	6.37	6.38	6.48	6.52
Performance Deviation(%)	56.60	54.80	54.69	54.16	53.41	53.10	54.00	53.45	53.86	53.68
Elapsed Time(Minutes)	310	320	330	340	350	360	370	380	390	400
PPM-S	3.03	3.05	3.04	3.07	3.09	3.12	3.08	3.14	3.2	3.25
Binary Search	6.34	6.37	6.53	6.4	6.21	6.08	6.49	6.37	6.45	6.48
Performance Deviation(%)	52.21	52.12	53.45	52.03	50.24	48.68	52.54	50.71	50.39	49.85

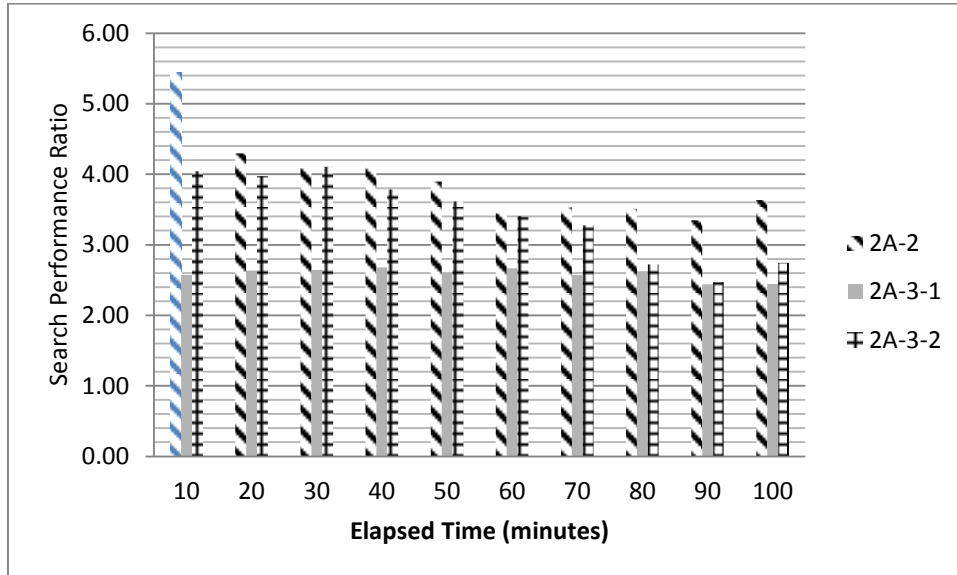


圖 5-40: 實驗 2A，出發後 10-100 分鐘
(錯誤容忍度實驗，Search Ratio)

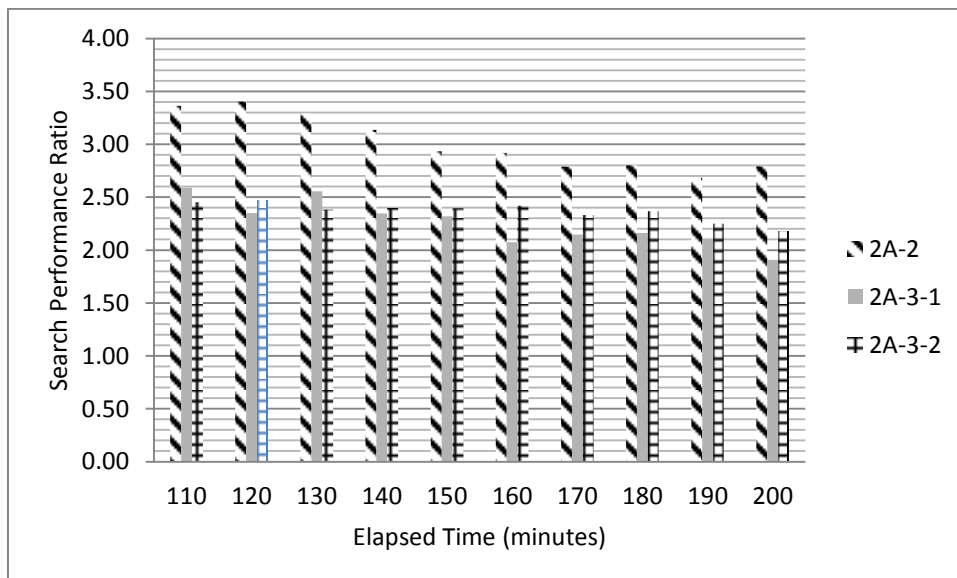


圖 5-41: 實驗 2A，出發後 110-200 分鐘
(錯誤容忍度實驗，Search Ratio)

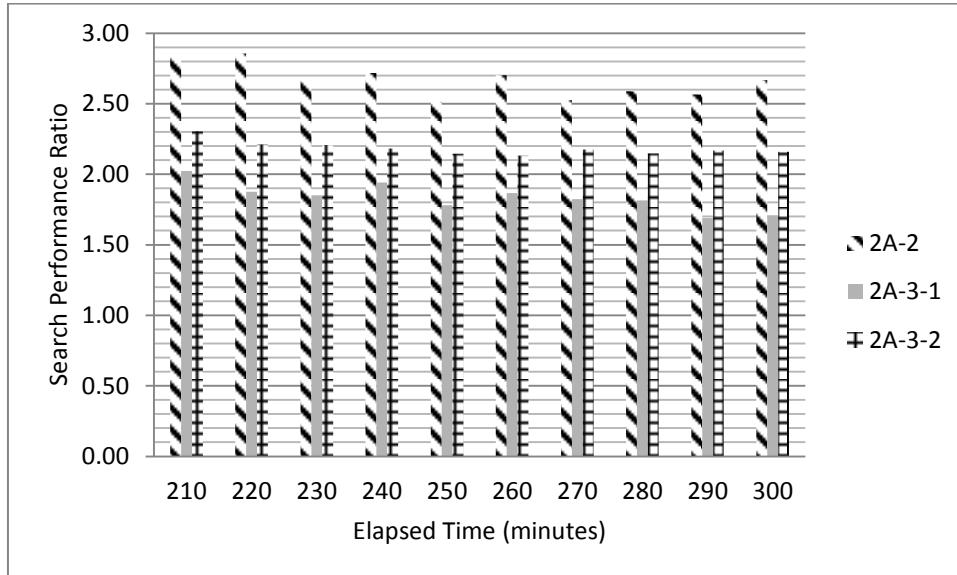


圖 5-42: 實驗 2A，出發後 210-300 分鐘
(錯誤容忍度實驗，Search Ratio)

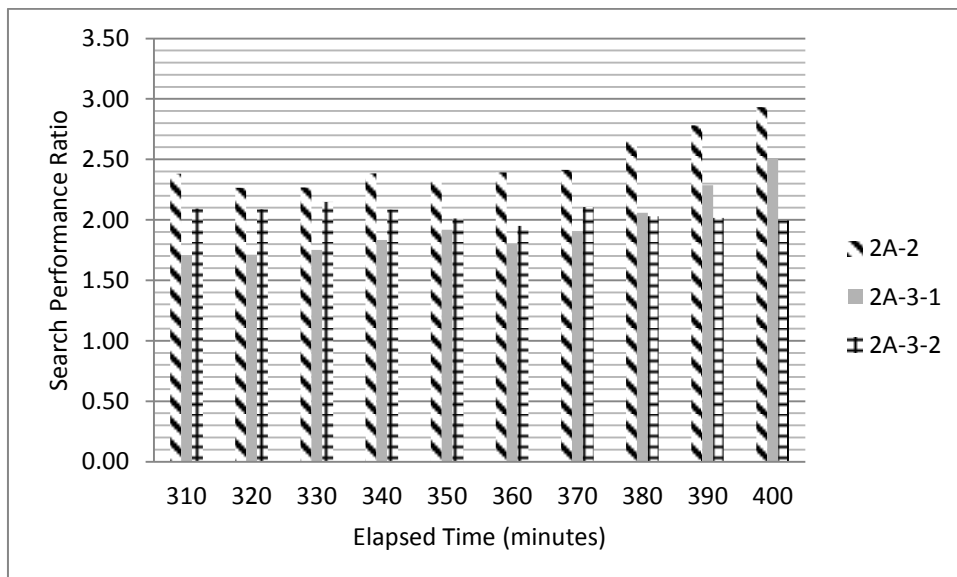


圖 5-43: 實驗 2A，出發後 310-400 分鐘
(錯誤容忍度實驗，Search Ratio)

表 5-13: PPM-S 錯誤容忍度實驗 (Search Ratio)

Elapsed Time (Minutes)	10	20	30	40	50	60	70	80	90	100
2A-2	5.44	4.29	4.09	4.10	3.90	3.45	3.53	3.51	3.35	3.63
2A-3-1	2.57	2.63	2.64	2.68	2.60	2.67	2.57	2.63	2.44	2.44
2A-3-2	4.04	3.97	4.11	3.78	3.61	3.41	3.28	2.72	2.47	2.74
Elapsed Time (Minutes)	110	120	130	140	150	160	170	180	190	200
2A-2	3.36	3.40	3.29	3.14	2.93	2.92	2.79	2.80	2.68	2.79
2A-3-1	2.59	2.35	2.55	2.35	2.32	2.07	2.15	2.16	2.10	1.90
2A-3-2	2.45	2.47	2.38	2.40	2.40	2.42	2.33	2.37	2.25	2.18
Elapsed Time (Minutes)	210	220	230	240	250	260	270	280	290	300
2A-2	2.83	2.86	2.66	2.72	2.51	2.70	2.53	2.59	2.56	2.67
2A-3-1	2.02	1.88	1.85	1.94	1.78	1.87	1.82	1.82	1.69	1.71
2A-3-2	2.30	2.21	2.21	2.18	2.15	2.13	2.17	2.15	2.17	2.16
Elapsed Time (Minutes)	310	320	330	340	350	360	370	380	390	400
2A-2	2.38	2.27	2.27	2.38	2.32	2.39	2.41	2.65	2.78	2.93
2A-3-1	1.70	1.71	1.75	1.83	1.92	1.80	1.90	2.06	2.29	2.51
2A-3-2	2.09	2.09	2.15	2.08	2.01	1.95	2.11	2.03	2.02	1.99

5.3.4 多重路徑機率預測準確度實驗

5.3.4.1 實驗 2B-1 環境參數設定

表 5-14:實驗 2B-1 參數

Parameters	Range of values
Number of paths	70
Number of edge a path	40
Probability of a branch	Uniform (0~1)
Gamma distribution	shape parameter = 2、3、4、5 scale parameter =2

實驗 2B-1 目的為評估所提出的計算機率公式 (2) 的準確度，實驗依照實驗值參數隨機產生 40 條路段組成一條 path，共有 70 條不同 path，每條路段花費時間的機率分佈假設為一 Gamma 分佈，其中 shape 參數設定為 2、3、4、5，scale 參數固定為 2，叉路選擇機率為 0~1，將此參數代入本研究提出的計算機率預測求得計算值，並利用亂數產生 10000 個 agent 極其各路段所花時間，求得統計值，兩者互相比較。

5.3.4.2 實驗 2B-1 結果與分析

依照表 5-13 參數取出之亂數，整段 path 所花費時間平均加總為 285.18 分鐘，最大值為 344.62 分鐘，因此我們預測的時間點，取在時間範圍 300 分鐘內，間隔為 10 分鐘，做一次預測準確度評估。

由下圖 5-44 至 5-46 呈現，出發後 10 至 100 分鐘內，平均 Error 介於 2.8% 至 4.1% 之間，公式 (2) 誤差較小，隨著時間增加誤差逐漸提高，原因在離出發時間較近的時候，mobile agent 大多落於前幾段。當時間越長，且每段健行時間差異很大且在有叉路

的時候，落於各段的分佈較廣，誤差逐漸增加。出發後時間 210~300 分鐘的平均偏差介於 6.78%至 7.59%之間，此時平均次數隨著時間增加反而減少，原因為此時間區間落於整體健行時間的後半段時間內，大部分的 mobile agent 普遍落在後幾段的健行路段即將走訪完完整路徑，誤差較小。當 mobile agent 隨著時間前進不斷往前方路段移動且允許叉路時，某一健行者在時間 t 時，位於各路段的機率，由公式 (2) 所計算的機率與本實驗所得到的統計值，之間的差距，兩者平均 Error 最小為 2.8%最大為 7.59%。

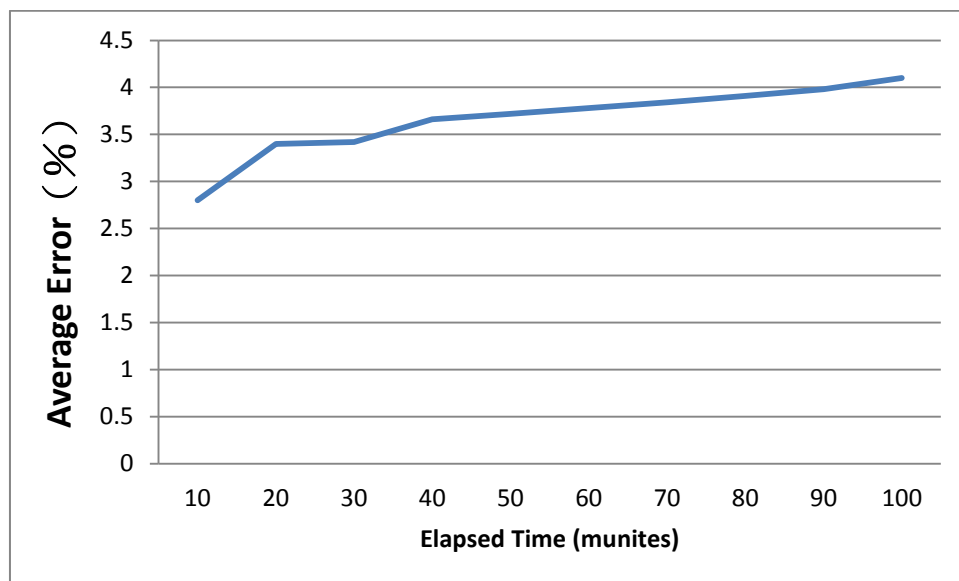


圖 5-44: 實驗 2B-1，出發後 10-100 分鐘
(預測準確度實驗)

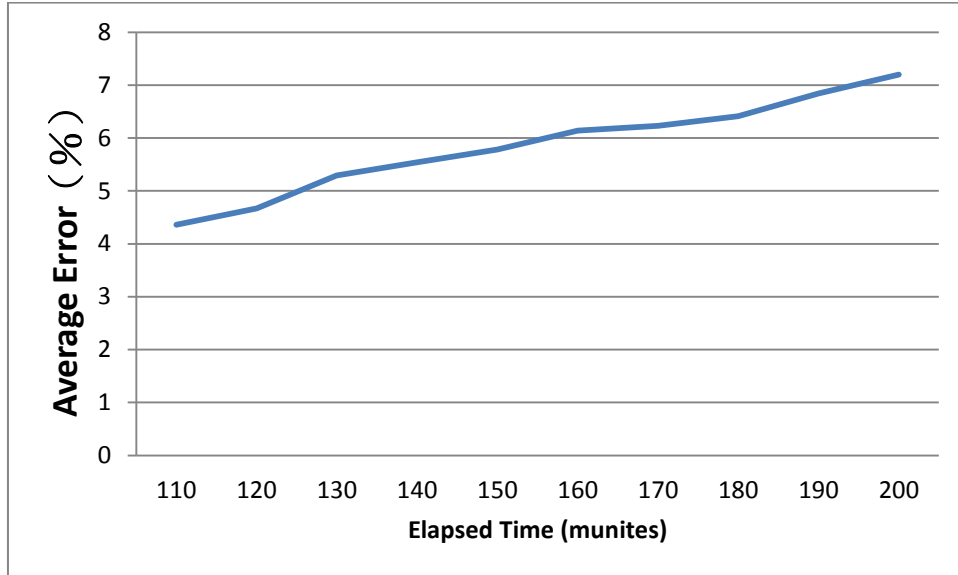


圖 5-45: 實驗 2B-1，出發後 110-200 分鐘
(預測準確度實驗)

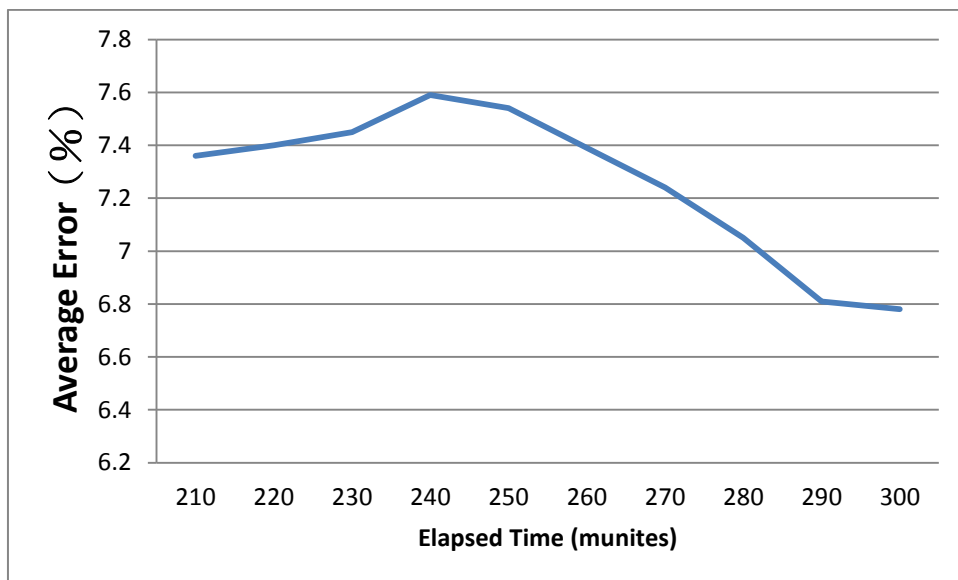


圖 5-46: 實驗 2B-1，出發後 210-300 分鐘
(預測準確度實驗)

表 5-15:實驗 2B-1 預測準確度實驗

Elapsed Time (Minutes)	10	20	30	40	50	60	70	80	90	100
Probability Calculation Error (%)	2.8	3.4	3.42	3.66	3.72	3.78	3.84	3.91	3.98	4.1
Elapsed Time (Minutes)	110	120	130	140	150	160	170	180	190	200
Probability Calculation Error (%)	4.36	4.67	5.29	5.54	5.78	6.14	6.23	6.41	6.84	7.2
Elapsed Time (Minutes)	210	220	230	240	250	260	270	280	290	300
Probability Calculation Error (%)	7.36	7.4	7.45	7.59	7.54	7.39	7.24	7.05	6.81	6.78

5.3.5 多重路徑機率預測搜尋法效能評估

5.3.5.1 實驗 2B-2 環境參數設定

表 5-16：實驗 2B-2 參數

Parameters	Range of values
Number of paths	70
Number of edge a path	40
Probability of a branch	Uniform (0~1)
Gamma distribution	shape parameter = 2、3、4、5 scale parameter =2

實驗 2B-2 使用表 5-15 參數產生測試案例，對同一組測試案例進行實驗，估評使用機率預測搜尋法-多重路徑(PPM-M)的平均搜尋次數 (number of search probes)，自出發後 10 分鐘起，每 10 分鐘搜尋 (search) 一次，直到 300 分鐘為止，實驗共執行 10000 回合。評估本研究所提出的機率預測搜尋法-多重路徑(PPM-M)與機率預測搜尋法-單一路徑(PPM-S)其 Normalized Search performance。

5.3.5.2 實驗 2B-2 結果與分析

由表 5-17 得知，機率預測搜尋法-多重路徑(PPM-M)的平均搜尋次數(number of search probes) 介於 1.51 至 3.95 間。機率預測搜尋法-多重路徑 (PPM-M) 是由公式 (2) 計算出機率較高的路段開始搜尋，當目標節點不在此路段上，再依照機率次高的路段依次搜尋(search)，機率預測搜尋法-多重路徑 (PPM-M) 的平均搜尋次數(number of search probes)為 2.88 次， Normalized Search performance 為 13.89，而機率預測搜尋法-單一路徑 PPM-S 的平均搜尋次數 (number of search probes) 為 2.25 次，Normalized Search performance 為 44.44， PPM-M 較 PPM-S 高出 30.55。

表 5-17 實驗 2B-2 搜尋法效能評估實驗

Elapsed Time (Minutes)	10	20	30	40	50	60	70	80	90	100
PPM-M	1.51	1.54	1.68	1.87	1.89	1.92	1.96	2.02	2.05	2.34
Elapsed Time (Minutes)	110	120	130	140	150	160	170	180	190	200
PPM-M	2.56	2.62	2.7	2.89	3.03	3.24	3.35	3.58	3.71	3.88
Elapsed Time (Minutes)	210	220	230	240	250	260	270	280	290	300
PPM-M	3.91	3.95	3.84	3.82	3.65	3.59	3.47	3.42	3.28	3.16

5.3.6 多重路徑錯誤容忍度實驗

5.3.6.1 實驗 2B-3 環境參數設定

表 5-18:實驗 2B-3-1 參數

Parameters	Range of values
Number of path	70
Number of edge on path	40
Probability of each branch	Uniform (0~1)
Gamma distribution (by Calculation)	shape parameter = 2、3、4、5 scale parameter =2
Normal distribution (by Experiment)	location parameter = 2、3、4、5 scale parameter =2

表 5-19:實驗 2B-3-2 參數

Parameters	Range of values
Number of Path	70
Number of edge on path	40
Probability of each branch	Uniform (0~1)
Gamma distribution (by Calculation)	shape parameter = 2、3、4、5 scale parameter =2
Gamma distribution (by Calculation)	shape parameter = 3、4、5、6 scale parameter =2

實驗 2B-3 目的為評估所提出的計算機率公式 (2) 對不同機率分佈及參數做錯誤容忍準確度實驗。實驗依照參數隨機產生 40 條路段組成一條 path，共有 70 條不同 path，叉路選擇機率為 0~1，計算值將每條路段花費時間的機率分佈 Gamma 分佈參數 shape 設定在 2、3、4、5，scale 參數固定為 2，將參數代入本研究提出的計算機率預測求得計算值。實驗 2B-3-1 及 2B-3-2 均依各實驗參數亂數產生 10000 個 agent 及其各路段所花時間，求得統計值，與計算值互相比較。接著進行錯誤容忍度實驗，使用表 5-18、5-19 實驗值參數各自產生測試案例，對同一組測試案例進行實驗，實驗機率預測搜尋法-多重路徑(PPM-M)的平均搜尋次數 (number of search probes)，自出發後 10 分鐘起，每 10 分鐘搜尋 (search) 一次，直到 300 分鐘為止，實驗共執行 10000 回合。

5.3.6.2 實驗 2B-3 結果與分析

實驗 2B-3 兩組實驗使用同一組計算值預測 mobile agent 在出發後 300 分鐘內，每 10 分鐘間隔 target agent 在各路段上的機率，做錯誤容忍度實驗。實驗 2B-3-1，依照 Normal 參數 location= 2、3、4、5，scale = 2 亂數產生實驗數據。在離出發時間較近的時候，mobile agent 大部分集中落於前方路段，平均 Error 較少，公式(2)誤差較小。當 mobile agent 隨著時間前進不斷往前方路段移動且允許叉路時，某一健行者在時間 t 時，位於各路段的機率，由公式(2) 所計算的機率與本實驗所得到的統計值，之間的差距，由表 5-20 呈現，平均 Error 最佳為 6.61%最差為 19.3%。實驗 2B-3-2，依照 Gamma 參數 shape= 3、4、5、6，scale = 2 亂數產生，評估本文所提出的機率預測方法在 Gamma 分佈其 shape 參數相差 1，預測的損害程度，兩者平均 Error 最小為 4.56%最大為 15.24%。將 2B-1、2B-3-1 與 2B-3-2 實驗之平均 Error 列於圖 5-56 至 5-58，可得知利用機率預測公式(2)在各實驗之預測準確度結果，以實驗 2B-1 較準確，平均 Error 在 7.59%之內，實驗 2B-3-2 次之，平均 Error 在 15.24%之內，實驗 2B-3-1 準確度最差，平均 Error 在 19.3%之內。

在 Normal distribution 下，搜尋效能損害評估，由下表 5-22 呈現，出發後 10 至

300 分鐘內，機率預測搜尋法-多重路徑(PPM-M)的平均搜尋次數 (number of search probes) 介於 2.28 至 4.75 間，平均為 3.98 次。機率預測搜尋法-多重路徑 (PPM-M)的 Normalized Search performance 為 10.05，相較於機率預測搜尋法-單一路徑 (PPM-S)的 Normalized Search performance 為 32.47，減少 22.42;在錯誤的 Gamma distribution 下之效能損害評估，由下表 5-23 呈現，出發後 10 至 300 分鐘內，機率預測搜尋法-多重路徑 (PPM-M)的平均搜尋次數 (number of search probes) 介於 2.52 至 4.05 間。平均為 3.45 次。機率預測搜尋法-多重路徑 (PPM-M)的 Normalized Search performance 為 11.6，相較於機率預測搜尋法-單一路徑 (PPM-S)的 Normalized Search performance 為 37.59，減少 25.99。下圖 5-50 至圖 5-52 列出 2B-2、2B-3-1、2B-3-2 的平均搜尋次數 (Number of Search Probes)。

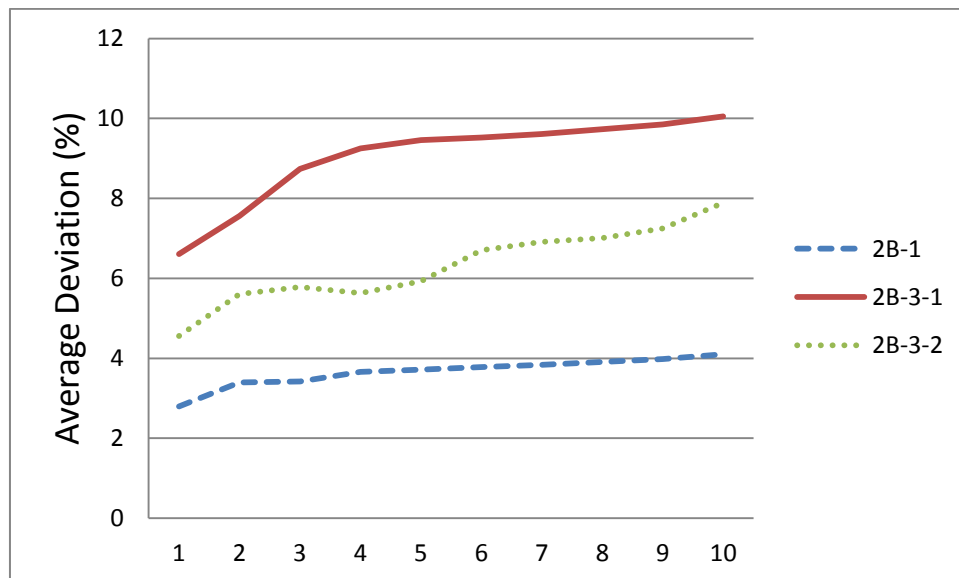


圖 5-47: 2B 各實驗之平均偏差比較
出發後 10-100 分鐘

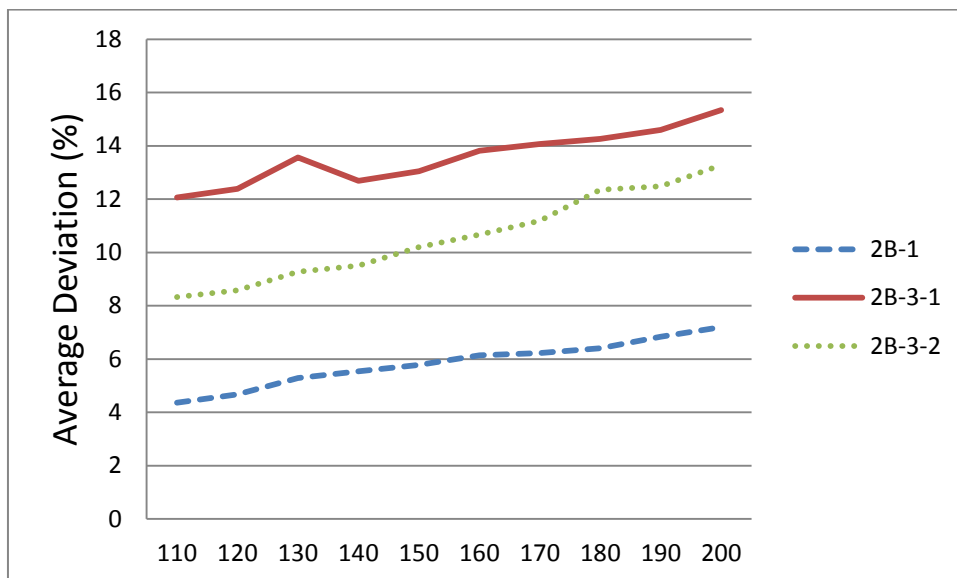


圖 5-48: 2B 各實驗之平均偏差比較
出發後 110-200 分鐘

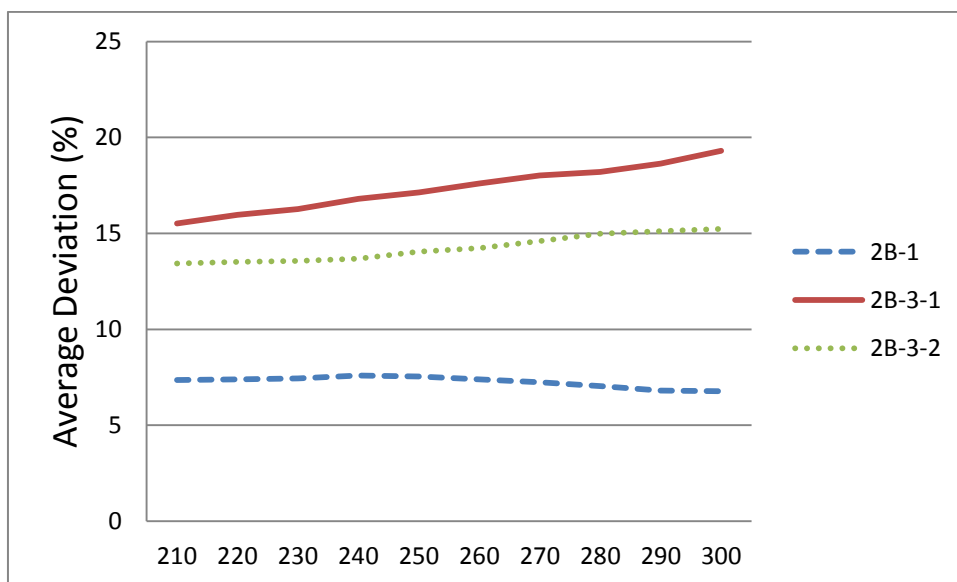


圖 5-49: 2B 各實驗之平均偏差比較
出發後 210-300 分鐘

表 5-20:實驗 2B-3-1 預測準確度實驗

Elapsed Time (Minutes)	10	20	30	40	50	60	70	80	90	100
Probability Calculation Error (%)	6.61	7.56	8.74	9.25	9.46	9.52	9.61	9.73	9.85	10.05
Elapsed Time (Minutes)	110	120	130	140	150	160	170	180	190	200
Probability Calculation Error (%)	12.06	12.38	13.56	12.69	13.05	13.81	14.07	14.26	14.59	15.34
Elapsed Time (Minutes)	210	220	230	240	250	260	270	280	290	300
Probability Calculation Error (%)	15.51	15.96	16.27	16.8	17.14	17.61	18.02	18.21	18.63	19.3

表 5-21:實驗 2B-3-2 預測準確度實驗

Elapsed Time (Minutes)	10	20	30	40	50	60	70	80	90	100
Probability Calculation Error (%)	4.56	5.61	5.78	5.63	5.93	6.7	6.91	7.01	7.25	7.89
Elapsed Time (Minutes)	110	120	130	140	150	160	170	180	190	200
Probability Calculation Error (%)	8.32	8.58	9.27	9.5	10.2	10.67	11.18	12.35	12.48	13.28
Elapsed Time (Minutes)	210	220	230	240	250	260	270	280	290	300
Probability Calculation Error (%)	13.44	13.51	13.57	13.68	14.05	14.24	14.6	14.99	15.12	15.24

表 5-22: PPM-M 錯誤容忍度實驗(Against Normal distribution)

Elapsed Time (Minutes)	10	20	30	40	50	60	70	80	90	100
PPM-M	2.28	2.37	2.63	2.78	2.97	2.99	3.06	3.65	3.92	4.15
Elapsed Time (Minutes)	110	120	130	140	150	160	170	180	190	200
PPM-M	4.4	4.58	4.67	4.71	4.75	4.74	4.73	4.69	4.6	4.43
Elapsed Time (Minutes)	210	220	230	240	250	260	270	280	290	300
PPM-M	4.4	4.36	4.35	4.37	4.28	4.26	4.31	4.08	4.01	3.92

表 5-23: PPM-M 錯誤容忍度實驗(Against inaccurate Gamma distribution)

Elapsed Time (Minutes)	10	20	30	40	50	60	70	80	90	100
PPM-M	2.52	2.64	2.74	2.81	2.92	2.98	2.96	2.87	2.88	3.01
Elapsed Time (Minutes)	110	120	130	140	150	160	170	180	190	200
PPM-M	2.98	2.99	3.1	3.15	3.35	3.68	3.89	3.9	3.95	3.99
Elapsed Time (Minutes)	210	220	230	240	250	260	270	280	290	300
PPM-M	4.02	3.99	3.98	3.97	3.99	4.01	4.05	4.03	4.05	4.01

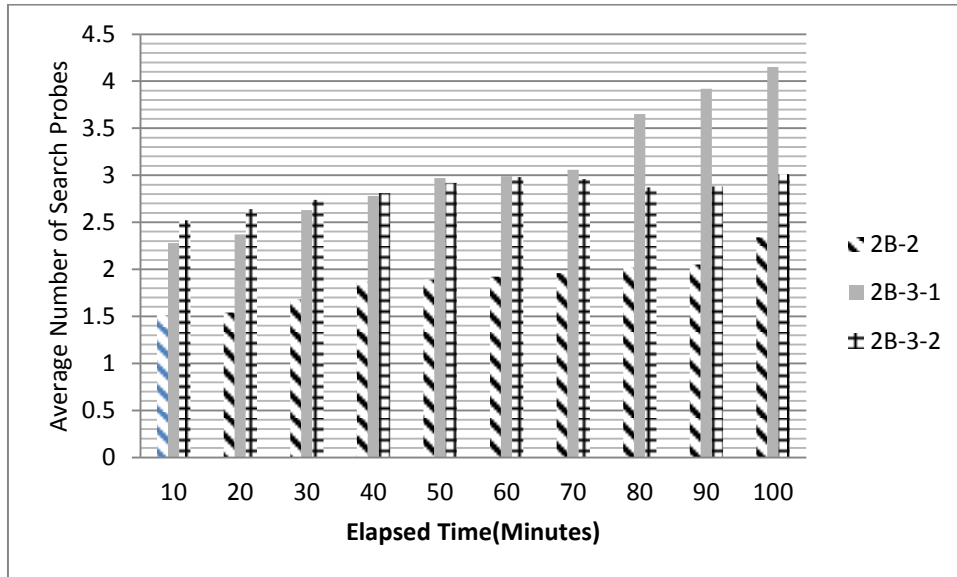


圖 5-50: 實驗 2B，出發後 10-100 分鐘
(錯誤容忍度實驗)

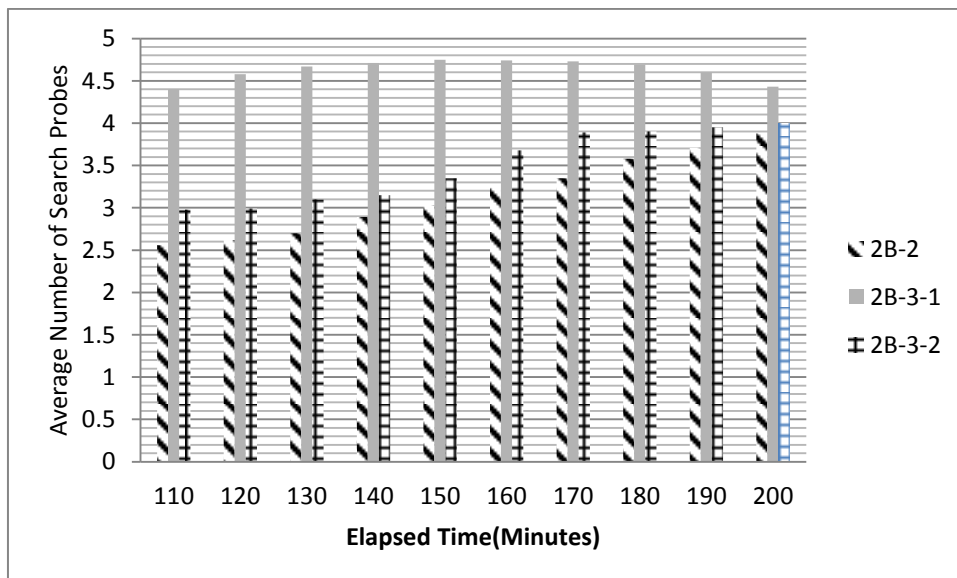


圖 5-51: 實驗 2B，出發後 110-200 分鐘
(錯誤容忍度實驗)

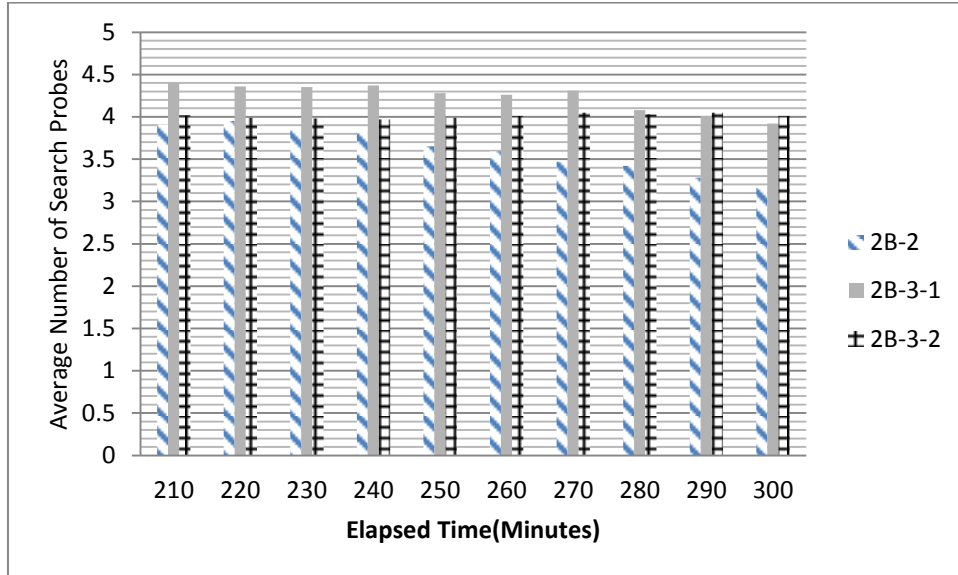


圖 5-52: 實驗 2B，出發後 210-300 分鐘
(錯誤容忍度實驗)

第6章

結論與未來研究

本篇論文主要為在控制網路底下，提出一個行動代理人之搜尋演算法，演算法利用計算機率的方式，預測在各個時間點行動節點落在每個控制節點之間路段的機率，依照計算出的路段機率高低順序來搜尋，以提高搜尋效率。機率預測演算法讓搜尋代理人能利用控制網路，優先搜尋到目標代理人所在路段，使得行動代理人技術的應用及優點，能移植到機會網路上，可增加「山文誌登山資訊系統」的應用功能。除了山難救援為其中的一個應用，其他還有變更集合地點、互動通訊...等，留待更多的應用被發掘。

依照服務區域內不同路徑的情況，本研究提出兩個模型，並使用機率預測搜尋法，預測行動節點可能所在位置優先搜尋此路段來降低整體搜尋時間。為驗證其預測的準確度，本文透過一連串的實驗，評估本法之搜尋效率以及當各路段花費時間的機率分佈假設有誤時，搜尋效率的受損程度。在我們的實驗中，機率模型之準確度極高，誤差不超過 7.59%，即使機率分佈錯誤，搜尋效能仍高於二分搜尋法約 2 倍。

本研究提出的模型仍需進一步改進，本文假設在其行動節點於各段路徑所花費的時間符合 Gamma 分佈，且為兩兩獨立事件。但在實際山區路段上，兩段健行路徑若設定為相依事件而非獨立事件，較為符合實際情況，但複雜度也相對地提升許多，未來研究若能考慮此點則更加貼近實際狀況。另本研究與 Gamma、Normal 兩機率分佈

比較，評估其錯誤容忍及預測準確度，倘若能與多個不同的機率分佈做一綜合比較，則能多方評估本研究的機率預測結果準確度以及是否適用於其他不同的機率分佈。

參考文獻

[1] S. A. Tamhane, M. Kumar, A. Passarella, and M. Conti, "Service Composition in

- Opportunistic Networks," in *Proc. of the 2012 IEEE International Conference on Green Computing and Communications*, 2012, pp.285-292
- [2] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," *IEEE Communications Magazine*, vol. 44, no. 11, pp. 134-141, Nov. 2006.
- [3] C.-M. Huang, K.-c. Lan, and C.-Z. Tsai, "A survey of opportunistic networks," in *Proc. of the 22nd International Conference on Advanced Information Networking and Applications*, 2008, pp. 1672-1677.
- [4] J. H. Huang, S. Amjad, and S. Mishra, "Cenwits: a sensor-based loosely coupled search and rescue system using witnesses," in *Proc. of the 3rd International Conference on Embedded Networked Sensor Systems*, 2005, pp. 180-191.
- [5] Y. T. Huang, Y. C. Chen, J. H. Huang, L. J. Chen, and P. Huang, "YushanNet: A delay-tolerant wireless sensor network for hiker tracking in Yushan national park," in *Proc. of the 10th International Conference on Mobile Data Management: Systems, Services and Middleware*, 2009, pp. 379-380.
- [6] Y. T. Huang, Y. C. Chen, J. H. Huang, L. J. Chen, and P. Huang, "YushanNet: A delay-tolerant wireless sensor network for hiker tracking in Yushan national park," in *Proc. of the 10th International Conference on Mobile Data Management: Systems, Services and Middleware*, 2009, pp. 379-380.
- [7] 陳伶志, 黃寶儀, and 黃致豪. "山文誌登山資訊系統." Internet: http://nrl.iis.sinica.edu.tw/YushanNet/tw_project.php, Retrieved on May. 1, 2013.
- [8] A. Fujihara, "ZebraNet and its theoretical analysis on distribution functions of data gathering times" in *Proc. of the Second International Workshop on Mobile Opportunistic Networking*, 2010, pp. 205-206
- [9] M. Martonosi, "ZebraNet and beyond: applications and systems support for mobile,

- dynamic networks”, in *Proc. of the 2008 International Conference on Compilers, architectures and synthesis for embedded systems*, 2008, pp.21-21
- [10] S. Farrell, *et al.* "InterPlanetary Internet." Internet: <http://www.ipnsig.org/>, Retrieved on May. 1, 2013.
- [11] Y. Chen, and H. Wu “Communications and Architectures for Interplanetary Internet”, in *Proc. of 2011 First International Conference on Instrumentation, Measurement, Computer, Communication and Control*, 2011, pp. 895-899
- [12] E. Brewer, *et al.* "Technology and Infrastructure for Emerging Regions (TIER)." Internet: <http://tier.cs.berkeley.edu/drupal/about>, Retrieved on May. 1, 2013.
- [13] Z. Lu and J. Fan, "Delay/Disruption tolerant network and its application in military communications," in *Proc. of the 2010 International Conference on Computer Design and Applications (ICCD)*, 2010, pp. 231-234.
- [14] Y. Sasaki and Y. Shibata, "Distributed disaster information system in DTN based mobile communication environment," in *Proc. of the 2010 International Conference on Broadband, Wireless Computing, Communication and Applications*, 2010, pp. 274-277.
- [15] 陳禹銘, 祝鈞毅, 李雅萍, and 周子勤, "多工協調技術之應用與展望," *機械工業*, no. 285, pp. 93-104, Dec. 2006.
- [16] D. S. Milojevic, F. Douglass, and R. Wheeler. *Mobility: processes, computers, and agents*. New York, NY: ACM Press/Addison-Wesley Publishing Co., 1999.
- [17] D. B. Lange and M. Oshima, "Seven good reasons for mobile agents," *Communications of the ACM*, vol. 42, no. 3, pp. 88-89, Mar. 1999.
- [18] L. T. Yang. *Mobile Intelligence*. Hoboken, NJ: John Wiley & Sons, Inc., 2010.
- [19] 楊柏華譯, "<<資訊新知>>:行動代理人技術(上)," *中央研究院計算中心通訊*, vol. 18, no. 23, pp. 184-188, Nov. 2002.

[20] 陳伶志, and 黃有德, "登山客的黑盒子," *中央研究院週報*, no. 1376, pp. 4-6, June 2012."

[21] J. H. Huang, S. Amjad, and S. Mishra. "Cenwits Search and Rescue System." Internet: <https://cse1.cs.colorado.edu>, Retrieved on May.1, 2013.

[22] Y.N. Lien, Y.S. Lin, "Placement of control network for mobile agents over opportunistic networks," in *Proc. of the 2012 IEEE International Conference on Pervasive Computing and Communications Workshops*, 2012, pp.631-636