

淡江大學運輸管理學系運輸科學碩士班碩士論文

指導教授：邱顯明 博士

以類免疫演算法應用於船席指派問題之研究

A Study On The Application Of Artificial Immune
Algorithm On The Berth Allocation Problem

研究生：葉仁吉 撰

中華民國 95 年 6 月

論文名稱：以類免疫演算法應用於船席指派問題之研究

頁數：124

校系（所）組別：私立淡江大學 運輸管理學系運輸科學碩士班

畢業時間及提要別：94 學年度第 2 學期碩士學位論文提要

研究生：葉仁吉

指導教授：邱顯明博士

論文提要內容：

經營一個港埠，國家需投入大量的資金、設備與人力，而港埠設立後的各項設施、經營績效良窳，對國家經濟發展影響甚鉅。船舶停泊作業為海運過程中首要作業，效率的好壞直接影響到後續作業的運作，而國內目前在停泊作業上，航商、代理行與港口船席調配小組以會議的方式進行，此種方式固然可以解決問題，但是能做出最佳決策，值得商榷。

本研究在模式構建上主要是以參予船席指派決策的航商、港埠當局為主，在考量航商在港時間最小化、港埠當局營運成本最小化來構建多目標模式，並以車輛繞徑的角度解釋船席指派問題。問題的求解則是以 C 語言自行撰寫類免疫演算法配合模糊理論進行求解。

以自行設計的小型範例配合窮舉法來確認模式的正確性，並利用此結果來進行演算法的證確性測試；演算法參數的選擇的上，則是利用了 ANOVA 檢定找出較佳的參數。此外，設計了不同規模、不同船舶到達型態範例，以禁忌搜尋法、類免疫演算法進行求解，比較兩者的求解品質。最後利用基隆港民國 95 年 5 月 22 日至 5 月 28 日船席指派紀錄作為實証資料、敏感度分析。

本研究將「船席」視為車輛繞徑中的「車輛」，「船舶」視為「需求點」，且假設需求點的作業時間、節線成本會因為車輛、需求點的不同而有所差異，以此方式解釋船席指派問題並構建多目標模式。

窮舉法的結果確認模式的正確性(有解)，而禁忌搜尋法、類免疫演算法都可以找到與窮舉法相同的解，證實了演算法的正確性。不同類型的範例中，經過統計檢定兩種演算法的求解結果，發現類免疫演算法在大型範例中有較佳的求解品質；以基隆港過去船席指派紀錄與模式求解比較結果，本模式下兩個目標的隸屬函數值，均優於實務指派結果隸屬函數值；敏感度分析方面，經由多次的測試，發現船席服務的船舶數與服務時間、使用成本成反向變動，合乎一般的邏輯性，更可證明本模式正確性，具有應用價值，可以作為實務單位營運決策之參考。

關鍵字：模糊理論、多目標規劃、類免疫演算法、禁忌搜尋法、船席指派

Title of Thesis :

Total Page : 124

A Study On The Application Of Artificial Immune Algorithm On The Berth Allocation Problem

Key word : Berth Allocation Problem, Artificial Immune Algorithms, Tabu Algorithms, Multi-objective Programming

Name of Institute : Graduate Institute of Transportation Science, Tamkang University

Graduate date : June 2006

Degree Conferred : Master

Name of student : Jen-Chi Yeh

葉仁吉

Advisor : Dr. Hsien-Ming Chiu

邱顯明博士

Abstract :

Most of the import and export freights are transported by sea transportation in Taiwan. Berth Allocation Problem (BAP) is essential problem for the operation of the port authority. The time and location assignment of the berths to vessels is crucial to the efficiency of the port operation. In the current practice in Taiwan, this decision is made by port authority and shipowner in the daily berth allocation meeting, which may not be the best policy to be adopted. A series of academic researchs have devoted on the optimal BAP model for this problem. However, there is no model address this problem from the point of views of both port authority and shipowners, which is the focus of the model proposed in this study.

In this research, we attempt to explore the application of the VRP (vehicle routing problem) concept on the BAP. With the focus of the interests of the port authority and shipowner, a multi-objective programming model is formulated in this study. There are two objectives in this model, i.e., the minimization of the total time in port of the shipowners, and the minimization of the working cost of the port authority.

With the NP-Hard nature of the proposed model, it is harder to get the optimal solution when the size of problem increasing. Therefore, the artificial immune algorithm (IA) and Tabu Search algorithm (TA) are proposed as the main solution procedure of the study for its diversity search ability. In order to reduce subject judgements in the multiple-objective programming, a series of fuzzy functions for the objectives are developed for this study. Finally, to evaluate the proposed model and the algorithms developed in the study, a series of case studies include one data from Keelung harbor are tested in this study. The results of these numerical tests indicate

that the IA performs better in large size problem. The solution of the proposed procedure provides better result than the conclusion of the daily berth meeting approach.



誌謝

兩年的研究所生涯，即將隨論文定稿結束後而劃下句點。在淡江的生活其間，感謝恩師 邱顯明博士對於研究方向的確立、論文架構、觀念啟發乃至研究方法的指導，適時與耐心地給予教導及指點，讓學生能夠摸索出一條屬於自己的方向。追隨恩師期間，有幸參予相關重要之計畫案，使學生在實務上獲得相當多的經驗，如資料查詢、訪談、專家座談、書面撰寫等，此外，恩師對於學生在求學態度、做人處事上的提點，裨益無窮，其中之感謝，實非短短幾句話能表達。

所內論文口試期間，感謝各位老師適時提供意見，讓學生從不同的觀點思考，得到不同的啟發，使得論文方向、架構臻為充實。在論文所外口試期間，承蒙中央大學 顏上堯教授與元智大學 丁慶榮教授蒞臨指導，對於論文內容、架構給與相當多寶貴的意見與指點，使得本論文內容更為完整，並更正了許多語意、寫法上的謬誤，在此更致上最高的敬意與衷心的感謝。

研究所期間，感謝每位同學在課業上的幫助，特別感謝文龍同學，常常在我低落時跟我聊天及給予不同的想法；感謝6年的好友百里，在課業、球場、論文等方面，一起度過這兩年充實的日子；感謝秋如媽媽三不五時的叮嚀，在我有困境時給予媽媽般的關切與開導，讓我暢吐心事；感謝 pona、老闆在電腦及軟體的協助；感謝掐米在課業上幫忙及生活上的討論；特別感謝首源兩年來默默地為大家貢獻，無論是生日、班上大小事務連絡，真的是辛苦你了！更感謝文賢，常常在重要時刻，邀集大家聚餐、喝茶、聊天，使得我們在壓力緊繃後，有個像家庭般的聚會，聊天、抒發心情。感謝班上其他同學，能在這兩年間一起生活、一起吃飯、一起打球、一起度過，感謝你們，這兩年的時光將會永遠銘記於心，有你們這一群好朋友真好。

從籃球場上的大學生活至淡江運科所學術殿堂，對我不但是個極大的轉變更是一個挑戰，能夠順利畢業，要感謝的人真的很多，除了學校的生活圈，更要感謝爸爸、媽媽、姐姐的體諒與支持，時時刻刻給予我鼓勵，最後要感謝的，就是我的貼心女友靜宜，常在我低落到想放棄時，給予我走下去的力量，沒有妳，這兩年生活將不會如此順利、溫馨，除了表達心中的感謝外，並將這份得來不易的小小成果與妳共享。

最後要感謝的，是所上所有的老師，兩年期間，不但給予了我專業上的知識，更給予我們許多無形待人處世的方法、人生觀，這些無形的資產，將會是我生命裡無價的寶藏，在此，獻上至高由衷的感謝。

葉仁吉 謹誌

2006.07.09

目錄

中文摘要

英文摘要

誌謝

目錄..... I

圖目錄 IV

表目錄 V

第一章 序論 1

1.1 研究背景與動機 1

1.2 研究目的 2

1.3 研究範疇 2

1.4 研究方法與流程 3

第二章 問題描述與相關文獻回顧 5

2.1 船席指派 5

2.1.1 船舶進出港流程與船席指派作業..... 6

2.1.2 港埠作業相關成本..... 6

2.1.3 船席指派相關文獻..... 8

2.1.4 小結..... 14

2.2 機門指派 18

2.2.1 機門指派因素..... 18

2.2.2 機門指派相關文獻..... 19

2.2.3 小結..... 22

2.3 模糊及多目標規劃..... 23

2.3.1 簡介..... 23

2.3.2 多目標規劃..... 23

2.3.3 模糊多目標規劃..... 26

2.4 車輛繞徑 28

2.4.1 船席指派與車輛繞徑..... 28

2.4.2 車輛繞徑問題..... 29

2.4.3 小結..... 30

2.5 類免疫演算法概述 30

2.5.1 生物免疫系統介紹..... 30

2.5.2 類免疫演算法..... 31

2.5.3 類免疫演算法相關研究..... 35

2.6 禁忌演算法概述 37

2.6.1 禁忌搜尋法簡介.....	37
2.6.2 禁忌演算法.....	37
2.6.3 禁忌演算法相關應用文獻.....	39
第三章 問題定義與模式構建	40
3.1 問題定義	40
3.1.1 問題特性與描述.....	40
3.1.2 問題定義.....	43
3.1.3 問題的轉換.....	43
3.2 前提與假設	45
3.3 問題架構	46
3.4 模式構建	47
3.5 模式修正與求解	49
3.6 小結	50
第四章 求解策略與方法	51
4.1 求解策略概念	51
4.2 問題輸入與輸出	53
4.3 資料結構	53
4.4 求解方法(一)類免疫演算法	56
4.4.1 類免疫演算法應用步驟與流程.....	56
4.4.2 類免疫演算法於船席指派問題求解.....	58
4.5 求解方法(二)禁忌搜尋法	61
4.5.1 禁忌搜尋法應用步驟與流程：.....	61
4.5.2 禁忌搜尋法於船席指派問題求解.....	63
4.6 小結	66
第五章 範例測試與參數設計比較	67
5.1 範例測試	67
5.1.1 窮舉法.....	67
5.1.2 小型範例測試求解結果.....	69
5.1.3 小結.....	71
5.2 啟發式演算法	72
5.2.1 類免疫演算法.....	72
5.2.2 禁忌搜尋法.....	73
5.2.3 小結.....	74
5.3 參數設計比較	76
5.3.1 參數設計內容與比較方式.....	76
5.3.2 演算法較佳參數.....	78

5.3.3 小結.....	78
5.4 演算法求解品質比較.....	79
5.4.1 比較的方式.....	79
5.4.2 演算法比較的結果.....	80
5.4.3 不同範例規模的比較.....	81
5.5 多目標船席指派求解流程.....	82
5.5.1 案例內容.....	82
5.5.2 求解過程.....	82
5.5.3 求解結果.....	85
5.6 實證研究與參數敏感度分析.....	87
5.6.1 資料收集與比較方式.....	87
5.6.2 實證比較的結果.....	88
5.6.3 參數敏感度分析.....	88
5.7 小結.....	90
第六章 結論與建議.....	91
6.1 結論.....	91
6.2 建議.....	93
參考文獻.....	94
附錄一、小型範例.....	98
附錄二、大型例題(5,50)相關資訊.....	102
附錄三、類免疫演算法求解求解在港時間參數實驗設計.....	105
附錄四、禁忌搜尋法求解求解在港時間參數實驗設計.....	109
附錄五、建議船席指派位置、順序與多目標隸屬函數值.....	111
附錄六、船舶進出港流程與基隆港船席指派作業.....	116
附錄七、生物免疫系統介紹.....	120

圖目錄

圖 1-1 研究流程	4
圖 2-1 連續型船席指派示意圖	10
圖 2-2 船席指派後的示意圖	12
圖 2-3 時空概念網路流動圖與指派結果展示圖	13
圖 2-4 四種航站概念型式	19
圖 2-5 單日機門指派作業網路圖	21
圖 2-6 類免疫演算流程圖	33
圖 2-7 兩兩互換之移步方式	37
圖 3-1 單一船席指派示意圖	40
圖 3-2 問題描述示意圖	42
圖 3-3 節點作業時間說明圖	45
圖 3-4 節線成本說明圖	46
圖 4-1 求解策略示意圖(一)	52
圖 4-2 求解策略示意圖(二)	52
圖 4-3 船舶屬性結構體	54
圖 4-4 船舶編號二維陣列結構體	54
圖 4-5 本研究運用類免疫演算法流程圖	57
圖 4-6 株落選擇說明圖	58
圖 4-7 輕鏈重組示意圖	59
圖 4-8 重鏈重組示意圖	59
圖 4-9 船席指派問題求解流程圖	60
圖 4-10 本研究應用禁忌搜尋法流程圖	62
圖 4-11 最鄰近法流程圖	64
圖 4-12 節點(船舶編號)交換示意圖	65
圖 4-13 更新禁忌名單示意圖	66
圖 5-1 船舶編號二維陣列結構體	67
圖 5-2 不同規模範例下演算法求解品質比較	81
圖 5-3 第一目標式隸屬函數圖	83
圖 5-4 第二目標式隸屬函數圖	84
圖 5-5 多目標運算迭代圖	85
圖 5-6 實證流程圖	87

表目錄

表 2-1 港埠成本分類表	7
表 2-2 Lai and Shih (1992)四種指派策略	9
表 2-3 Nishimura 等待船席的時間	10
表 2-4 指泊船舶順序與選定船席方案說明	11
表 2-5 船席被同時佔據的長度	12
表 2-6 船席指派問題相關文獻整理表	15
表 2-7 船席指派與機門指派之比較	22
表 2-8 多目標規劃應用文獻整理	24
表 2-9 報酬表計算內容	26
表 2-10 「船席指派問題」轉換為「車輛繞徑問題」之構想	28
表 2-11 生物免疫系統與類免疫演算法特性對照表	32
表 2-12 類免疫系統與生物免疫系統之對照表	35
表 2-13 類免疫應用相關文獻	36
表 3-1 「船席指派問題」轉換為「車輛繞徑問題」之構想	44
表 5-1 範例一窮舉求解結果	69
表 5-2 範例二窮舉求解結果	70
表 5-3 範例三窮舉求解結果	70
表 5-4 範例四窮舉求解結果	71
表 5-5 類免疫演算法求解小型範例結果	72
表 5-6 禁忌搜尋法求解小型範例結果	73
表 5-7 禁忌搜尋法、類免疫演算法求解小型範例結果比較	75
表 5-8 類免疫演算法參數設計相關內容	76
表 5-9 禁忌搜尋法參數設計相關內容	76
表 5-10 抗體數目實驗設計運算結果	77
表 5-11 演算法較佳參數	78
表 5-12 禁忌演算法與類免疫演算法求解結果比較	80
表 5-13 不同規模範例下演算法求解品質比較	81
表 5-14 第一目標式測試求解值	83
表 5-15 第二目標式測試求解值	84
表 5-16 不同船舶到達分配下多目標求解結果	86
表 5-17 船席會議指派與模式指派結果比較表	88
表 5-18 船席指派參數敏感度分析	89

第一章 序論

1.1 研究背景與動機

根據 World Bank(2003)的報告顯示，全球貨物透過海運方式的運送比例每年增加 4~5%之間，且目前全球約有 80%以上的貨物運輸必須透過海運，而貨櫃化之標準運輸型態更是海運的主流；台灣四面環海，屬於海島經濟型態的環境，由於天然資源稀少，進口與出口的產品須依靠海洋運輸的方式，由交通部統計資訊網站的資料顯示(<http://www.motc.gov.tw/service/year-c/ycmain.htm>)，台灣進出口(不含轉運)的國際貨物運量中，海運占了 99.5%，對台灣而言，海運扮演了一極為重要之角色。

港埠(Port)與港灣(Harbor)詞雖類似，但是意義不同：

1. 「港灣」係指位於沿海、沿江(河)或內湖之水灣，有寧靜之水面，足夠之水深，可供船舶駛入及錨泊之場所，一般而言，僅指天然形成之水域。
2. 「港埠」除具有港灣性質外，且須具備旅客上下及貨物裝卸、儲存之設備、船舶繫泊設備、船舶安全導航之設備、船岸通訊之設備(如信號臺、雷達站、無線電話及電報等)、船舶修理設備、油煤水電之供應設備，及通達腹地之鐵路公路等運輸工具，並配備足夠之作業勞力與有效之管理組織，以提供旅客及貨物運轉之需要。

經營一個港埠，國家需投入大量的資金、設備與人力，而港埠設立後的各項設施、經營績效良窳，對國家經濟發展影響甚鉅。船舶停泊作業為海運過程中首要作業，效率的好壞直接影響到後續作業的運作，而國內目前在停泊作業上，航商、代理行與港口船席調配小組以會議的方式進行，在避免擁擠、充分運用船席、增進港埠營運效能等目標下，指派船舶到適當的船席停靠；長期規劃為配合原有港口營運之增長或航運方法之改變所做的擴建、改善或新港的闢建規劃。

港埠的作業中，船席指派的運作對於航商而言，主要會影響其在港的時間(包含等待時間、作業時間)、產生相關的成本及對後續船舶作業效率產生影響；對於港埠管理單位而言，會因應不同的航商、船舶停泊位置、貨物裝卸量而產生不同的資源使用，如相對應的機具使用、貨物移動距離、人員等產生相對應的成本。

從船舶等候時間指標來看，若船舶等候時間過長，會使得船、貨主負擔鉅額的等待成本，而船、貨主等待成本是一種社會成本，因為它會轉嫁給消費者，增加社會大眾的負擔。

因此，一套有效率的船席指派作業可以縮短船舶停留在港口的時間，對於企業而言，可以盡速地將貨物送達到世界各地顧客的手中；對於港埠管理單位而言，適當的指派可以使港口作業效率提高、有效率運用資源，亦可以提高港口本身的競爭力或吸引力。

1.2 研究目的

船席指派在實務作業上，是以決策者(港埠當局與船商)依經驗來進行，此種方式缺乏數理分析基礎，決策方式固然可以解決問題，但是否為最佳及最有效率的方式，有待商榷。過去已有相關文獻中對於船席指派問題的探討，以最低成本、最有效率或是配合指派原則來分配船席，其方式大都是以「航商」的角度為觀點，使得船舶的總服務時間(等待時間加上裝卸貨時間)最小。

港埠屬公營事業之一環，其經營目標應為最有效使用各種資源，使全民福祉達最大，如何使港埠之各項資源組合達到總成本最小化經營亦如其他企業體一般，不僅可維持與他港競爭力，並能持續生存發展。

本研究將會從「航商」與「港埠管理單位」的角度出發，結合多目標規劃的概念，構建出船席指派的模式，求出船席指派順序；與過往文獻不同的地方在於本研究假設港埠當局會因應不同船舶、貨物裝卸量、停泊位置而所產生不同的成本。本研究之目的整理如下：

1. 在兼顧船舶在港時間(航商角度)與船席的使用成本(港埠管理單位角度)最小的情況下，建立船席指派的模式。
2. 在船舶的到達時間 ETA(Estimated time of arrival)、裝卸貨作業時間等相關資訊已知的情況下，求出平日船舶的指派順序。
3. 將多目標規劃方法結合類免疫演算法、禁忌搜尋法撰寫求解程式，可供未來類似問題的求解；並比較兩種演算法在求解上品質與速度上的差異。
4. 以基隆港過去的船舶指派紀錄作為實証資料，比較模式與實務指派兩者之差異。

1.3 研究範疇

研究所探討之問題乃針對，船舶抵達外港後，等候安排船席、靠泊碼頭、裝卸貨櫃完成、解纜離開碼頭及駛離港口之一系列作業流程。而有關貨櫃場站作業及內陸運輸問題，則不加以探討。

本研究為一日常作業問題之探討，故以一日為限，即本研究所求出的指派順序是港埠每日之指派作業。

1.4 研究方法與流程

在研究方法上，本研究主要的核心是在於「船席指派」的問題上，因此主要課題為如何指派船席，以找出船舶等待時間、港埠當局成本最小化，本研究將此一問題視為一「組合最佳化」問題，根據此一最佳化指派構思此一問題。

由文獻得知，在探討船席指派時，大都是以「航商」的角度為出發點，以最小化船舶的等待時間為目標。基於實務上做法(航商、港埠當局兩者以會議的方式所做出的決策)，本研究認為船席的指派，不應只侷限於「航商」的角度，也應該將「港埠管理單位」的觀點列入考量，故在目標式的分析上，以單一目標來求解可能會對實際狀況有所遺漏，故本研究將會嘗試以多目標決策概念來構建目標模式，以充分反應所涉及的各個層面。

船席指派問題是屬於組合最佳化的問題，因此本身具有 NP-HARD 的特性，即會有「規模增加時求解不易」的現象，過往的文獻在探討船席指派問題時的求解方法，大都是以啟發式解法為主。由於「基因演算法」本身具有平行的演算能力，適合處理群組化資料，已被廣泛應用且證實為求解此等問題最佳方法之一。

由相關文獻得知，類免疫演算法在求解最佳化問題上，有不錯的效果，其演算機制不但與基因演算法類似，且其求解品質比基因演算法好，因此，本研究將會嘗試以類免疫演算法進行此問題之求解。

最後，本研究以多目標規劃的模式配合擬議之演算法，找出在符合各目標函數之最滿意解，作為本研究輸出的「最佳指派順序」。並利用基隆港過去的船席指派記錄作為實證資料並與模式求解結果進行比較，研究流程如圖 1-1 所示。

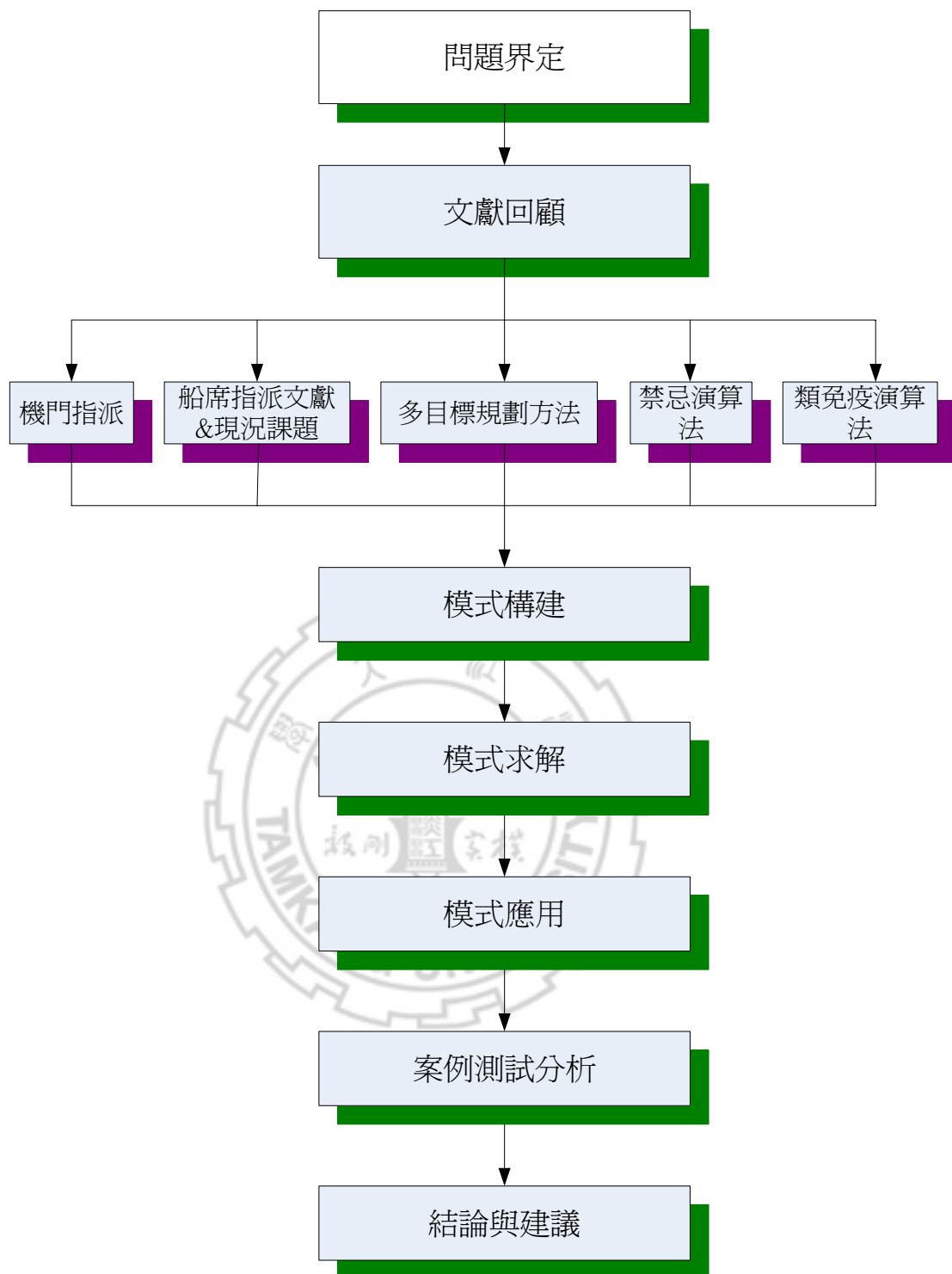


圖 1-1 研究流程

第二章 問題描述與相關文獻回顧

本研究所探討的是問題為在考量「航商」與「港埠管理當局」兩者的立場下，求出港口平日的「船舶指派順序」，此一問題屬於一種決策過程，過去的文獻將此種問題稱之為船席指派問題(Berth Allocation Problem, BAP)。

因此，在本章會先探討實務上對於船席指派問題處理的情形及考慮的因素、條件，使我們對於研究的主體有確切的認識與瞭解。並回顧學術上所做之相關研究，從過去的一些相關文獻中瞭解在處理此課題時所做的相關假設條件、參數設定、考慮因素或是有不足的地方，以協助構建模式。

此外，機門指派問題(Gate Allocation Problem, GAP)與船席指派概念類似，除了所探討的對象、環境限制與目標函式不同外，這兩方面的最終目的相同，也就是要找出最佳化指派模式。

若是將整個港口視為一個虛擬的場站，船席視為車輛，船舶視為需求點，不同的車輛在不同的需求點間巡迴，船席指派問題可以視為車輛繞徑，因此，亦會回顧相關車輛繞徑模式的相關文獻，以助後續的模式構建。

在演算法方面，基因演算法在求解多目標規劃上，有相當多的應用(Jones et al., 2002)；由文獻得知，類免疫演算法的演算機制與基因演算法類似，且求解的品質似乎比基因演算法更佳，另外，將會再回顧禁忌搜尋法的相關文獻，以比較此兩種演算法的求解差異。

因此，本章將劃分為六個小節，分別為船席指派相關文獻、機門指派相關文獻、多目標規劃相關文獻、車輛繞徑相關文獻、類免疫演算法相關文獻及禁忌搜尋法相關文獻。

2.1 船席指派

此小節會先說明船舶進出港流程、船席指派會議進行的方式，以瞭解船席指派是位於港埠作業中的哪個環節，之後回顧學術上船席指派相關文獻，使對於研究主題有確切的認識與瞭解。

2.1.1 船舶進出港流程與船席指派作業

根據王丘明君(2003)所著「港埠管理」，摘要如下，附錄六為船舶進出港部分內容。

船舶進出港流程大致分為「船舶進、出港預報」、「船舶進出港檢查」、「船席調派」、「靠泊碼頭」、「裝卸作業」。

以基隆港現行的船席調派作業為例，其主要目的為充分運用船席，使船席的空閒時間減至最少，並使船舶於船席之船舶能迅速完成裝卸工作，而不發生中斷之狀況；一般考量船席調派時考量的因素有船舶長度、船舶吃水與碼頭深度之配合、內陸運輸、倉儲能量、裝卸設備與天候因素等。

現行參予船席指派會議的人員有航商、代理公司與船席調派小組(港埠單位相關人員參加)，每日舉行一次會議，遇有特殊狀況時會再加開一次會議，目的是在於決定翌日(或以後)之船席分配。

2.1.2 港埠作業相關成本

港埠經營對於國家而言，扮演相當重要的一環，相關的作業對於航商、港埠當局都會產生相對的成本與資源使用，對於港埠當局與航商而言，適時的瞭解相關作業成本可以作為其訂定費率的基礎或是其他作用，因此，本研究將港埠作業所產生的相關成本做一整理。

在港埠作業成本方面，徐玉琦(1998)依基隆港務局之損益表，將港埠之成本依企業之營業性質劃分為三大項，說明如下，本研究將其整理如表 2-1 所示：

1. 營業成本：係指營業所發生之各項成本而言，其依業務性質分為港埠成本及營業成本，分別敘述如下。

(1)港埠成本：

①港灣費用：係為辦理港灣業務如供船舶進出、船席調度、碼頭管理、船舶繫泊、帶結纜使用拖船協助船舶靠離碼頭之拖船調度費用等。

②棧埠費用：辦理碼頭船上裝卸業務及貨物搬運所發生費用、通棧倉儲業務所發生如整理費用、機械設備費、損害賠償費用、設備租金等。

③維持費用：港埠疏浚、維護所發生成本。

(2)其他營業成本：船舶修造廠辦理局外工程船舶機具修造所發生成本

2. 營業費用：包含業務費用、管理費用及其他營業費用，分別敘述如下。

(1) 業務費用：是指港埠推廣業務，辦理有關營運事務所發生之各項費用如業務組掌管 港埠業務規劃、營運、管理之事項；資訊室掌管資訊業務系統分析設計、程式撰寫、電腦操作等事項；員工訓練所辦理所屬員工、碼頭工人訓練進修等。

(2) 管理費用：係指綜理港埠所發生之管理及總務費用，總務室掌理文書、印信、庶務、出納材料採購、管理通訊、員工及眷屬保健等事；秘書室掌理項其他營業費用係指研究發展費用。

(3)其他營業費用：秘書室掌理項其他營業費用係指研究發展費用。

3. 營業外費用：正常業務營運範圍以外之各項費用或損失，如財務費用如利息費用兌換損失及警政航政民防費用等。

表 2-1 港埠成本分類表

營業成本	係指營業所發生之各項成本而言，其依業務性質分為		
	港埠成本	港灣費用	港灣業務如供船舶進出、船席調度、碼頭管理、船舶繫泊、
		棧埠費用	辦理碼頭船上裝卸業務及貨物搬運所發生費用
		維持費用	
	其他營業成本	船舶修造廠辦理局外工程船舶機具修造所發生成本	
營業費用	業務費用	港埠推廣業務，辦理有關營運事務所發生之各項費用	
	管理費用	係指綜理港埠所發生之管理及總務費用	
	其他營業費用	秘書室掌理項其他營業費用係指研究發展費用。	
營業外費用	正常業務營運範圍以外之各項費用或損失，如財務費用如利息費用兌換損失及警政航政民防費用等。		

本研究的研究範圍從船舶進港、等候船席、裝卸貨物，有關內陸運輸的部份不加以探討；而船舶係指貨櫃船舶，因此針對貨櫃船提供設施服務時產生的成本加以整理，說明如下。

1. 港灣費用部分：包含貨櫃輪碼頭碇泊業務、曳船業務及垃圾清理業務，分別敘述如下。

(1) 貨櫃輪碼頭碇泊業務為維持碼頭營運之直接成本，有貨櫃碼頭之折舊、保養維修、航道濬深。

(2) 曳船業務係指協助船舶進、出港使用拖船靠離碼頭所發生之成本費用，解纜業務為協助船舶碇泊所使用之纜工、帶纜船及帶纜車所發生之成本，其成本有帶纜船、車之折舊、保養、維護、燃料、用人費用。

(3) 垃圾清理業務係為船舶在港口使用垃圾船清理垃圾所發生成本。

2. 有關貨櫃輪棧埠作業費用：包含裝卸業務、機械使用業務、人工成本、物料成本、行政管理成本及場租業務，分別敘述如下。

(1) 裝卸業務係指貨櫃自輪船卸至碼頭或貨櫃場之裝卸作業，故其相關成本為人工成本及行政管理成本，分別敘述如下。

① 人工成本：即貨櫃機具操作人員配合橋式機作業之大班工人以及貨櫃裝卸管理員之勞力成本。

② 行政管理成本：係指間接參與貨櫃裝卸作業之行政管理人員薪資辦公室及設備折舊及什項費用。

(2) 機械使用業務：其成本項目為資本成本即有關各類機具設備每年所攤提之折舊費用其計算公式為不動產折舊＝造價÷壽年，動產折舊＝購價÷(壽年＋1)，動力成本指各項設備設備操作所耗用之電費或燃料費之成本。

(3) 人工成本：貨櫃機具維護人員成本。保養維護成本係指各項營運設施每年之保養、維護費用。

(4) 物料成本：為維持機具設備有效運作，所需投入之各種零件物料所發生之成本費用。

(5) 行政管理成本：如前裝卸費所述。

(6) 場租業務：係指貨櫃場提供貨櫃存放於貨櫃場、碼頭空地所發生之業務成本。故其有關成本主要為資本成本、電力成本、人工成本及行政成本。

2.1.3 船席指派相關文獻

黃明居(1991)利用與實務者訪談對基隆港船席調派之整體情境加以分析，建構一知識庫系統，並引用排列組合的概念，進行求解船舶即時可用之時空。船舶即時可用之時空，可以提供航商有關可靠泊船席之訊息並且輔助航商選取可靠船舶；利用一定的指泊規則對船舶進行指泊，可以針對突發性之狀況、人類思考變數不足時，輔助航商作整體分析進而提升決策品質，此種船席指派之整合系統，雖能輔助航商做出較佳的決策，但對於整體港口之效益，並無實質考量(僅僅以航商為主體)。知識庫系統輔助人工決策，須對既定已擬定之規則做層層推理，

此種船席指派整合系統中規則多達 276 條，雖然對於船席指派的情境分析得宜，但是幾乎以航商為主體。

Lai and Shih (1992) 將碼頭當成離散的碼頭，把碼頭切成幾個區段，每一個區段在同一時間內只能停泊一艘船舶將船舶分成大型、中型與小型三種，並將四種貨櫃碼頭分派策略加以比較與探討，如表 2-2 所示，結果顯示，策略二與策略四的等待時間較短，但是碼頭使用率方面則是以第一及第三較高。

表 2-2 Lai and Shih (1992) 四種指派策略

Strategy_1	不考慮船舶類型的優先指派法則，只考慮先到先指派的法則進行貨櫃碼頭分派作業。
Strategy_2	不考慮船舶類型的優先指派法則。為了將船舶停留在系統內的時間與等待時間最小化，因而考量到船舶離開時間，作者以 PASCAL 撰寫模擬程式進行方案比較，當船舶到達時，系統會估出服務時間與離開時間，並以離開時間最早的船舶優先指派。
Strategy_3	認為大型船舶所花費的成本較高，因此應該優先指派較大型之船舶。以碼頭是否有閒置的情形下，分為不同指派原則： 1. 碼頭閒置時：以離開時間最早的船舶優先指派。 2. 當碼頭沒有閒置時：指派順序為大型、中型、小型。
Strategy_4	將大型船、中型船的等待時間最小化。將船舶分兩類，第一類為大型、中型船舶；第二類為小型船舶。 1. 第一類的船舶優先指派，且在第一類中，以離開時間最早的船舶優先指派。 2. 第二類中，以先到先指派的原則進行碼頭分派作業

Brown et al. (1997) 探討的是海軍港口的船舶指派問題，利用混合整數規劃的方式構建模式，其目標式為利潤最大化，藉以幫助管理者迅速掌握港口運作情況。此篇文獻與一般的商用港口不同，最大的差別在於，對於潛艇給予最大的優先權，即潛艇不會有「到達港口後，產生等待指派作業的情況」，因此，假設有其他非潛艇的船舶正在該潛艇的船席上進行相關作業，該船舶須移置其他的船席；且須要有特定服務時，會有所謂「船席轉換(Berth Shift)」的情況發生，會有船舶被指派到一個已經有另一艘船舶停泊的船席上。

Imai et al. (1997) 使用 0-1 整數規劃的問題，探討船席指派問題，等候線上的船舶，大都是以先到先服務 (FCFS) 的原則指派，此篇文獻做出一個推斷 (假設)，即提高港口的吞吐量的目標下，最佳化的船席指派應該不考慮先到先服務的規則。其目標式分為兩部分，第一部份為船舶等待時間與作業時間 (裝貨卸貨時間) 最小化，第二部份為船舶誤點時 (晚於 ETA 到達)，須被指派到其他位置，要將此種情況的發生的次數最小化，最後以權重法配合 C 語言加以求解。

Lim(1998)首先將船席指派問題當成「連續性」的問題加以處理，此篇文獻的目標式是以最小化船舶的安全空隙，使得港口可以在同樣條件下服務更多的船舶。其假設船舶的長度、作業時間與到達時間已知的情況下，利用先到先指派的原則進行指派，連續型船席指派示意圖如圖 2-1 所示。

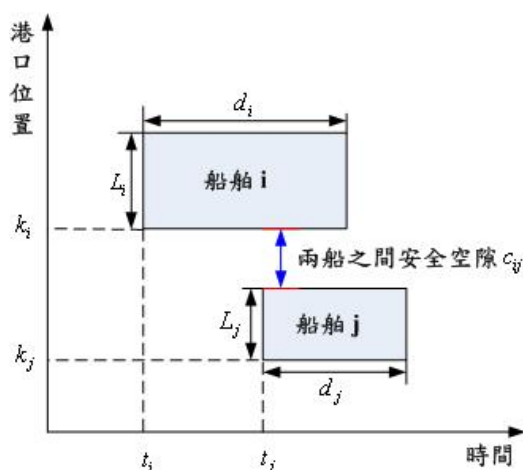


圖 2-1 連續型船席指派示意圖

Nishimura(1998) 模擬公用船席 (public berth system) 上的船席指派，目的是要測試船舶的 averages 的等待時間，其結果如表 2-3 所示，即使在只有 4 個船席下，平均等待時間差不多為半小時。

表 2-3 Nishimura 等待船席的時間

Number of berths	3	*4	5	6	7
Time(hours)	221	36	10	2	0

李育達(1999)的動機在於基隆港曾於民國 84 年 10 月開始著手進行船席指派電腦化之研發，但是過程並不順利，現行的船席指派作業仍然停留在以會議的方式進行，此篇文獻基於此，將船席指派電腦化應注意的問題與如何建構良好之船席調派邏輯進行分析。利用與實務者訪談所得之結論，兼顧「航商」與「港口」之權益，並分別考慮指泊船舶之順序、船舶可靠泊船席之界定與如何選定可靠泊船席等影響調配船席之因素，建立調配船席的邏輯，並擬定八種自動選定船席的方法，配合四種不同指泊船舶之順序，試驗選定船席之方法與指泊船舶之順序對船席調配的影響，指泊船舶順序與選定船席之方案如表 2.1-4 所示。實驗的結果得知，在四種指泊船舶的順序下，利用電腦自動選擇可靠泊船席，優先選取佔用尚未指泊之船舶可靠泊船席時空比率較小之船席，並將船舶優先指泊於船席之兩端位置，所得之船舶等候時間與延遲離港時間均降低了；同時，利用即時調整已指泊但尚未靠泊之船舶指泊的位置，可使直線船席更具有連續性，有助於調配船席。

表 2-4 指泊船舶順序與選定船席方案說明

指泊船舶之順序	<ol style="list-style-type: none"> 1. 先到先排，同時間到達依較早提出申請者先行指泊。 2. 先到先排，同時間到達依 ETD 較早者先行指泊。 3. 先到先排，同時間到達依船長較短者先行指泊。 4. 依現行優先貨櫃船合約規定，決定船舶指泊之順序。
選定船席之方案	<ol style="list-style-type: none"> 1. 以隨機選定直線船席與靠泊位置。 2. 隨機選取直線船席，但優先將船舶指泊於船席之兩端。 3. 優先選取船舶可靠泊空間較大的船席。 4. 優先選取船舶可靠泊空間較小的船席。 5. 優先選取佔用船席時空比率大者之船席。 6. 優先選取佔用船席時空比率小者之船席。 7. 優先選取佔用未指泊船可靠泊時空比率較小者之船席。 8. 利用方案七選取船席，加入即時調整船舶靠泊於直線船席之位置。

Imai et al. (2001) 認為貨物量需求的降低，會使得港口的設施與裝備變的多餘，因此使得私人船席系統的成本增加；日本港口的收費過高，部分增加的成本是因為相對較少貨物量所造成生產能力過剩所造成的結果，因此作者想要找出船席分派最佳化作業。此篇文獻以 0-1 整數規劃的方法來構建「靜態」與「動態」的船席指派問題，「靜態」與「動態」的最大差別在於工作的進行中其他船舶仍然可以到來。靜態船席指派問題(Static Berth Allocation Problem, SBAP)與動態船席指派問題(Dynamic Berth Allocation Problem, DBAP)目標式方面都是以最小化「船舶等待作業時間」與「船舶本身作業的時間」，在「船舶作業時間」上，作者認為船舶的作業時間影響在同一船席上的後面所有船舶，因此船舶作業時間上，以特殊的係數加權($T-K+1$)，以圖 2-2 所示，第四艘船的作業時間可能為後續船舶 5、船舶 2 與船舶 3 的部份等待時間的交集。此外，在其文獻中舉出日本神戶港口(Kobe)某個月份中的資料，資料顯示出在 29 天中，所有的船席同時被利用到的時間只有 9 小時，如表 2-5 所示。因此，作者想要維持在相同服務水準下，藉由船席指派來降低所須的船席數量，達到相同的績效。最後，使用拉式鬆弛法將問題簡化，並以拉式鬆弛法為基礎，發展出一個啟發式解法。

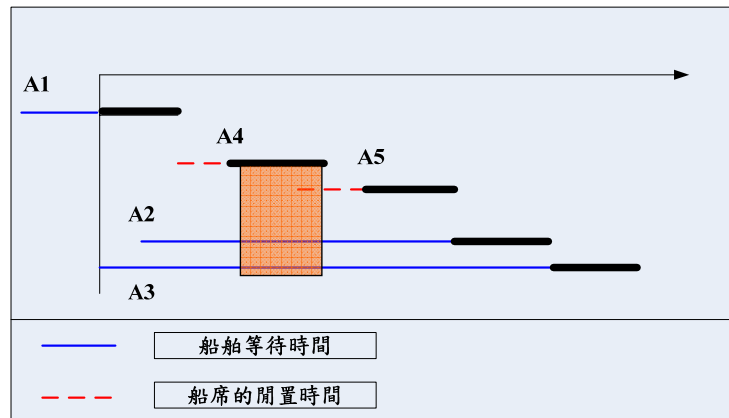


圖 2-2 船席指派後的示意圖

註：第四艘船的作業時間可能為後續船舶 5、船舶 2 與船舶 3 的部份等待時間的交集，因此以對於船舶 4 的作業時間給予特殊的權重 $T-K+1$

表 2-5 船席被同時佔據的長度

Number of berths	0	1	2	3	4	5	6	7	Total
Time(hours)	19	76	138	159	136	90	62	9	689
%	3	11	20	23	20	13	9	1	100

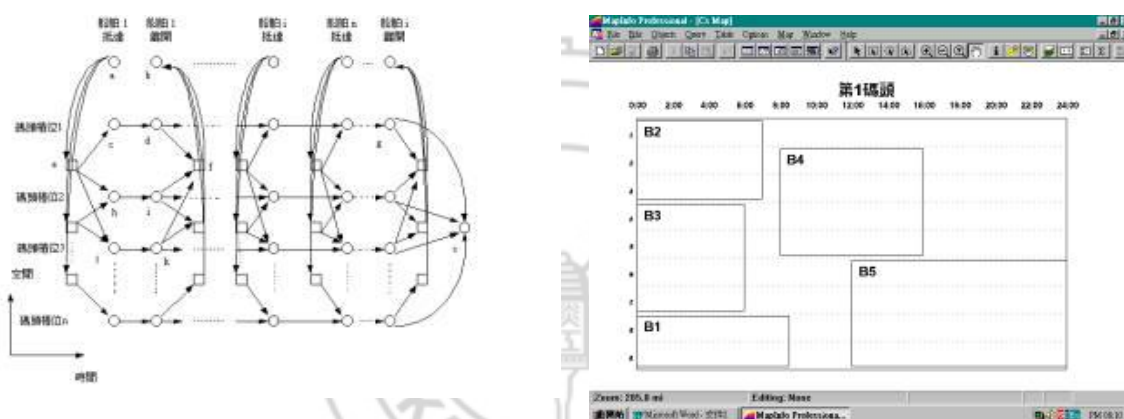
Nishimura et al.(2001)與 Imai et al.(2001)所構建的模式相似，在目標式方面，沒有船舶作業時間的權重，目標式為最小化船舶等待時間與作業時間。其所構建出的模式，為一個非線性規劃，為求解這種 NP-HARD 的問題(求解時間不能於有限時間內求出)，使用基因演算法求解。模式中對於船舶作業時間有做出相關的定義與解釋，實務上，船舶的處理時間會與所指派的船席位置、貨櫃場站與船席的距離、起重機數量等有關；其模式對於船舶的處理時間參數上，假設為確定性(deterministic)，即船舶的作業時間假設為事先已知條件且不會改變。目前的公用船席系統對於指派的優先原則大致上有二，為先到先服務(FCFS)及依據貨物量而定，即到達時間較早、有較大貨物量的船舶會有較高的服務優先權；此篇文獻的模式不保證一定會有上述兩種原則。此篇文獻的限制式，除了一般原則外(船舶只能被一個船席服務外)，另外考量了船舶吃水與船席水深的配合、船舶到達後才可以被服務，在其模式中，與過去一般文獻不同的是，將現實世界的多船服務考慮進去，所謂「多船服務」是指，只要船舶的總長度不會超過碼頭的總長度，就允許有多船(Multi-Ship)的情況發生，即允許船席「同時服務兩艘以上的船舶」。最後以基因演算法求解，並以兩種不同的「編碼方式」比較求解的品質。

Kim and Moon(2003)，將船席指派問題界定為「連續型」的問題來加以研究。其決策變數為船舶停泊的時間、位置，且假設船舶到達的時間為一種隨機分配，離開時間已知的情況，以先到先指派的原則進行指派作業。其目標式中包含兩種

成本，一種為船舶未停到成本最低的碼頭所產生的額外成本，第二類為當船舶比預定時間晚離開所造成的處罰成本。在求解方法上，以 LINDO 求解 7 艘以下的小型問題，此外，以模擬退火法進行大型問題的求解。

謝東緯(1997)指出船席調派問題為一「指派」問題，而船席調配作業為港埠作業重要之一環，船席調配作業長久以來都以會議的方式來決定，這種方式不但耗時耗力，再者，船席的調配攸關航商之權益甚大，在短短的時間內要思考整個情境，能否做出最佳的決策，值得商榷。因此，此篇文獻的主要目的就是希望能藉由電腦優越計算能力的幫助，改善此種船席調配會議方式，協助船席調配人員，提高其作業效率。此文獻所考慮的因素有(1)船舶到達時間(2)船舶離開時間(3)船舶長度，構成一含額外限制式網路流量問題，最後以基隆港過去船舶的相關資料作模式的驗證，在其模式中，成本是為假設的條件，並且在文獻最後，建議可以將指派的成本進行更深入的探討。此篇文獻之網路流動圖與指派結果展示如圖 2-3 所示：

圖 2-3 時空概念網路流動圖與指派結果展示圖



時空概念之網路流動圖

求解之圖示展示

陳朝文(2002)以基隆港貨櫃船舶停泊碼頭作業為例，進行現況分析與改善，進行現況分析與改善。以民國 88 年基隆港的營運資料，考量目前的優先停泊制度，在此所提的優先制度航商每年必須依據自己的需求與港務局簽訂優先碼頭使用合約，並透過統計的方法對相關作業找出適當的機率分配。並利用 FORTRAN 語言模擬基隆港採用先到先服務原則所產生的情況，在與現況比較，研究之結果適用於基隆港貨運量並沒有大幅回升，及其他條件相同情形下參考，除平均等候進港時間其差異幅較大外，優先靠泊制度並不特別優於先到先服務制度，因此作者建議應該取消優先停泊制度。

張企宏(2004)將「貨櫃碼頭分派問題」與「起重機指派問題」進行探討，以 0-1 整數規劃的方式構建模式，在船舶到達時間、卸貨時間已知下，將在規劃期間內到達的船舶以總等待時間最小化做為目標函數。其方式為先構建貨櫃碼頭分派、起重機分派模式，並整合貨櫃碼頭分派與起重機指派作業模式。

2.1.4 小結

1. 實務上的船席指派

- (1) 以港口當局的立場而言，在船席指派的過程中，港方必須扮演著服務的角色，以降低到港船舶之平均等候時間、提升機具與船席使用效率及增加港口的競爭力為準則，維持港埠設施為公有，秉持公平、公正、公開的作業方式對到港船舶進行指泊。此外，港埠屬公營事業之一環，其經營目標應為最有效使用各種資源，使全民福祉達最大，如何使港埠之各項資源組合達到總成本最小化經營亦如其他企業體一般，不僅可維持與他港競爭力，並能持續生存發展。
- (2) 船席指派在現實世界的所考慮的因素有：船舶長度、船舶吃水與碼頭深度之配合、內陸運輸、倉儲能量、裝卸設備、船舶特性、天候因素。
- (3) 船席指派此種決策過程，在合乎實體環境的限制下(船舶與船席地理環境限制等考量)，並且考量港埠當局(提升機具、船席使用效率、減少成本)與航商(避免產生額外給付的費用、縮短在港時間)兩者的立場，決定出翌日或之後的船席指派。

2. 文獻中對於船席指派的探討

船席指派相關文獻整理如表 2-6 所示，由文獻可以得知，學術上在面對此種問題的情形時，使用的研究方法有系統模擬、知識庫系統、時空網路圖、最佳化模式。

在目標式方面，大都是以最小化船舶等待時間為目標為基礎，或搭配不同的特殊權重係數(Imai, 2001)，並且也有動態船席指派的觀念產生(Imai, 2001)；在以時空網路圖所構建的模式中(謝東緯，1997)，目標式為最小化成本，且其成本為假設，而在最佳化的模式中，大部分的文獻都以單一目標並以航商的角度為出發點。

船席指派要考量港埠管理當局與航商的立場，即船席調派的主要目的，莫過於找出船應於何時靠泊、靠泊於哪一船席、何時能離開港口，再者就是希望能夠通盤考慮航商的滿意度(例如船舶等候時間是否降低、船舶在港時間能否縮短)及港埠整體資源是否充分被利用(例如提高機具、船席之使用率等資源有效利用以節省成本)藉以提高港口競爭力，因此，本研究將會同時考量雙方的角度，搭配多目標的概念，構建出船席指派模式。

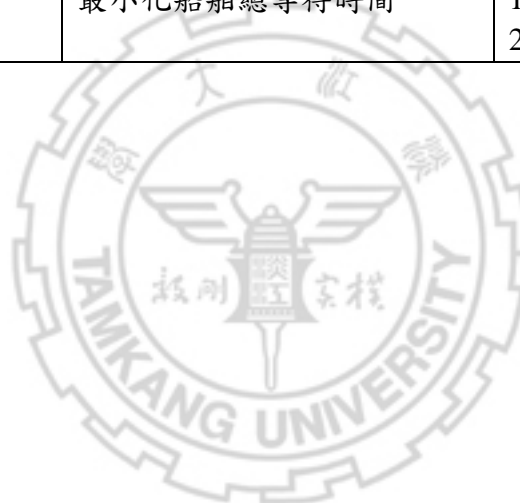
船席指派的現況與相關文獻如上所述，從實務上的船席指派中，我們可以得知，船席指派此種決策過程而言，在相關環境限制下(船舶長度、船舶吃水與碼頭深度之配合、內陸運輸、倉儲能量、裝卸設備、船舶特性、天候因素)，考量航商與港埠管理當局雙方立場下，以增進港埠機具、船席使用率、提升港口競爭力並減少航商的等待時間或在港時間，協調出船席指派的順序。

表 2-6 船席指派問題相關文獻整理表

篇名	作者	研究方法	模式目標	評析
A study of container berth allocation	Lai, K.K., Shih, K., 1992	以 PASCAL 撰寫模擬程式進行方案比較。		將船舶(大、中、小)配合四種指派策略以模擬的方式，從「船席使用率」與「船舶等待時間」績效指標來評估結果。
Optimizing submarine berthing with a persistence incentive	Brown, G.G., Cormican, K.J., Lawphongpanich, S., Widdis, D.B., 1997.	混合整數規劃	利潤最大化	<ol style="list-style-type: none"> 1. 給予潛艇最大優先權，且潛艇到達港口後不會有等待船席的情況發生。 2. 會有會有船席轉換(berth shift)的情況發生。 3. 此種情況在一般商港不可能發生，模式不適用於一般港口的指派作業。
Efficient planning of berth allocation for container terminals in Asia	Imai, A., Nagaiwa, K., Chan, W.T., 1997.	0-1 整數規劃	最小化 1. 等待時間 + 裝卸貨時間。 2. 船舶到達時間晚於 ETA 時指派到其他位置次數。	其認為等候線上的船舶，若是以先到先服務的指派原則下，並不能達到最大的吞吐量。
Public use effects on the performance of container berth	Nishimura, E., 1998	模擬		<ol style="list-style-type: none"> 1. 測試不同船席數下的船舶平均等待時間。 2. 四個船席數下的平均等待時間約半小時。
The berth scheduling problem	Lim, A, 1998	最佳化	最小化兩船所須保持的安全空隙	<ol style="list-style-type: none"> 1. 第一篇以連續性的角度來探討船席指派問題 2. 假設船舶長度、作業時間與到達時間已知的情況下。
The dynamic berth allocation	Imai, A., Nishimura, E.,	0-1 整數規劃	最小化「船舶等待時間」與「船舶作業時間」	<ol style="list-style-type: none"> 1. 構建靜態(SBAP)與動態(DBAP)的船席指派問題

problem for a container port	Papadimitriou, S., 2001			<ol style="list-style-type: none"> 2. SBAP 與 DBAP 最大差異在於工作的進行中其他船舶仍然可以到來(ships arrive while work is in progress)。 3. 對於船舶作業時間給予特殊權重。 4. 試圖維持相同的服務水準下降低船席數量。
Berth allocation planning in the public berth system by genetic algorithms.	Nishimura, E., Imai, A., Papadimitriou, S., 2001	0-1 整數規劃 以 GA 求解	最小化「船舶等待時間」與「船舶作業時間」	<ol style="list-style-type: none"> 1. 船舶作業時間無權重 2. 對於船舶的作業時間有做說明與假設，且定義為確定性(deterministic)事先已知的條件。 3. 考量到船舶吃水與船席水深的限制 4. 允許船席可以同時服務兩艘以上船舶的情形 (Multi-Ship)
Berth scheduling by simulated annealing	Kim and Moon, 2003	0-1 整數規劃		<ol style="list-style-type: none"> 1. 同時決定停泊時間與位置 2. 視為連續性的船席
知識庫系統應用於基隆港貨櫃船席調派作業之研究	黃明居，1991	知識庫系統		<ol style="list-style-type: none"> 1. 可以針對突發性之狀況、人類思考變數不足時，輔助航商作整體分析進而提升決策品質。 2. 雖能輔助航商做出較佳的決策，但對於整體港口之效益，並無實質考量(僅僅以航商為主體)。
船席調派問題之研究	謝東緯，1997	含額外限制式的網路流動問題	最小成本	<ol style="list-style-type: none"> 1. 考量船舶到達時間、船舶離開時間、船舶長度 2. 利用時空網路的概念構建模式，並用分枝界定法求解 3. 以網路流動的角度最小化成本，成本部分定義為船舶指派給某個船席所產生的成本，如操作成本(較大的船有較大的操作成本)

基隆港貨櫃船 席調派電腦化 之分析	李育達(1999)	系統模擬		<ol style="list-style-type: none"> 1. 利用與實務者訪談的結論，兼顧航商與港口的權益。 2. 分別考慮指泊船舶的順序、船舶可靠泊的船席與應如何選定可靠泊船席等影響因素，建立船席調派的邏輯。
貨櫃碼頭分派 問題之研究	張企宏(2004)	0-1 整數規劃	最小化船舶總等待時間	<ol style="list-style-type: none"> 1. 將貨櫃碼頭指派與起重機指派進行探討 2. 以模擬退火法(SA)進行求解



2.2 機門指派

機門指派與船席指派類似，除了考量的限制、服務主要對象不同外，兩者在概念上非常相似，因此將對機門指派相關文獻做一整理。

航班機門指派一直是機場營運效率的關鍵。除牽涉到機門需求數量外，更需要針對起降此機場的客機的停泊時間以及停泊相關作業做一整體的考量。過去往往以人工方式進行作業，不過此法既無效率，又無法因應大型且班次密集的機場。因此，諸多學者亦曾針對此一問題進行研究，其方法大致可分類為「數學解析方法」及「非數學解析方法」。

2.2.1 機門指派因素

停機坪指派(Gate Assignment)就是將落地航機指派至一個停機坪，使得航機能在停機坪作業，到場、離場旅客能在登機門上下飛機；基本上，影響機門指派因素有以下幾點(Ali Hanghani, 1994)：

1. 到達、離開及轉機旅客之步行距離。
2. 到達、離開及轉機旅客行李處理之運送距離。
3. 航空公司飛航時刻表。
4. 飛機機型與登機門之相容性。

「旅客步行距離」為大部分機門指派模式的主要考量因素，到達旅客的步行距離為到場航機機門至航站大廳出口的距離，離開旅客的步行距離為航空公司 check-in 櫃檯至行李拖運處至離場航機機門的距離，轉機旅客的步行距離為航機到、離場兩機門之距離，為了旅客的便利及減少旅客延誤時間，必須將旅客步行距離最小化。

Wirasinghe 及 Vandebona(1987)針對四種不同的航站概念型式(如圖 2-4 所示)用模擬技術得到旅客步行距離的分佈，根據數據顯示 T 型雙航站大廳步行距離比其他型式短。

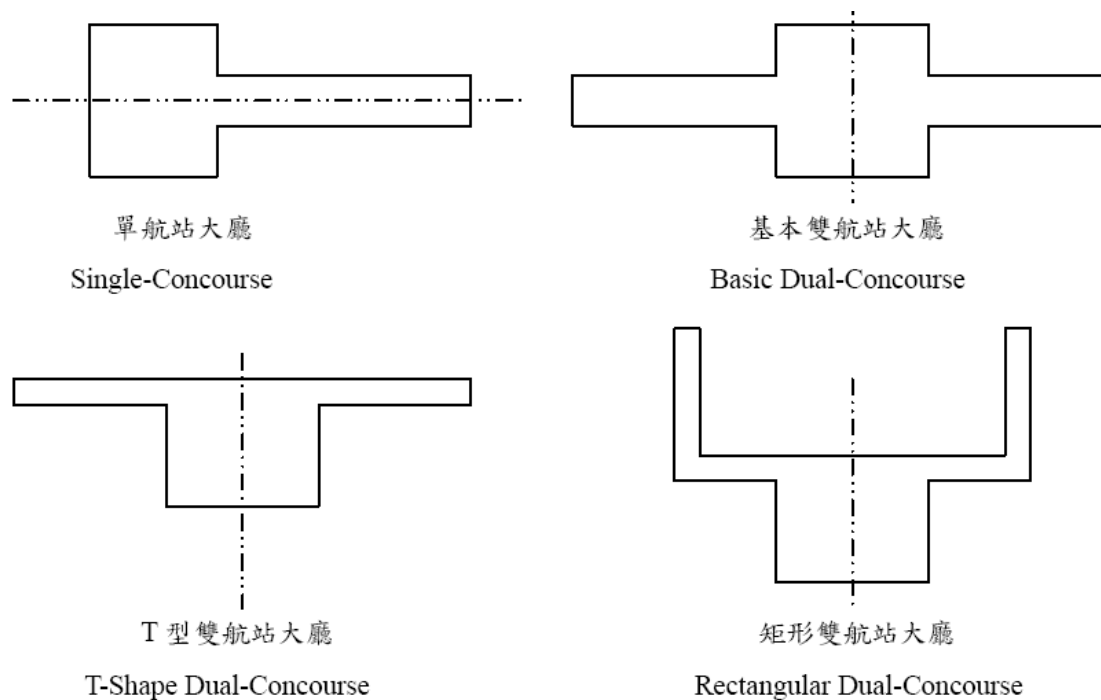


圖 2-4 四種航站概念型式

Bandara 及 Wirasinghe(1992)進一步考慮在機坪數量及航站大小固定下將到達、離開及轉機旅客之步行距離公式化並最小化，並分析各種航站型式比較其旅客步行距離進而得到最佳之航站型式。

到達、離開及轉機旅客行李處理之運送距離與航空公司之地勤作業成本成正比，也是機門指派的考慮因素；而航空公司的飛航班表是機門指派的依據及輸入資料，機型與機門的相容性為機門指派之限制條件。

2.2.2 機門指派相關文獻

Babic et al. (1984) 針對機門指派問題，以旅客步行距離最小化作為目標，發展以分枝定限法(Branch and Bound)為主的模式進行求解。

Mangoubi and Mathaisel (1985)考量旅客步行距離最小化為目標式，將其定式為混合整數問題並鬆弛成線性問題進行求解，且另外發展啟發式解法進行比較，其所提出的啟發式搜尋法是以準備降落的班機中，旅客最多的班機可以優先指派到最適當的機門。

Haghani and Chen (1998)把機門指派問題轉換為二次指派問題(Quadratic Assignment Problem, QAP)來求解，此篇文獻較不同的是，考量旅客步行距離與行李轉運距離最小化作為目標函數，並發展一套啟發式解法求解，並與套裝軟體 CPLEX 的解進行結果與運算時間的比較。

Bolat (1999)則認為班機延誤對於機門指派作業影響相當大，不僅會造成原先排定好的機門處於閒置狀態，也可能造成機門指派作業必須配合現況來加以改變，有效率的機門指派應該和機門的閒置時間(slack time)相關，因此以機門總閒置時間最小化為目標處理即時擾動問題。作者假設班機到達機場的時間已知，且班機停留在機場的時間會服從一種機率分配之情況下，利用 0-1 整數規劃的概念

來構建數學模式，在解題過程中，先以分枝界限法來求解小型指派問題，並發展一種啟發式演算法來求解大型問題。

Yan and Hao (2001)曾針對多目標的機門指派規劃，建立一多目標零壹整數規劃模式，並利用權重法、連續變換權重法、變數產生法、單體法及分枝定限法，發展一求解演算法，並以中正國際機場為例測試，效果良好。

張東珍(1993)以 0-1 整數規劃來構建機門指派模式，並以旅客總步行距離與總延誤時間最小化做為目標函數。並依先到達先指派、大型班機優先指派、交替指派、過夜班機先指派等機門使用策略來構建動態機門指派模式，進而模擬不同指派作業方式。作者所指的旅客總步行距離包含兩部份，其中一部份為出入境旅客來往於班機所停泊的機門至中央機門的距離，另一部份為轉機旅客於兩機門之間的步行距離。延誤的時間則是指入境旅客因到達班機未被即時指派而產生的額外等候時間。

顏上堯、張家銘(1997)使用網路流動技巧構建模式於則定式機門指派問題為一多重貨物網路流動問題，構建一模擬航班起降的時空網路進行求解。對於航班和機門一對一服務的特性，則以額外限制式加以滿足。而求解方式則以拉氏鬆弛法暨次梯度法，配合一拉氏啟發解法加以求解。此為文獻中首度以網路模式應用於機門指派問題。

陳春益等人(1997)則參考顏上堯與張家銘(1997)的時空網路模式，另行構建一單一貨物網路流動模式。

謝致遠(1998)以 0-1 整數規劃的方式來構建機門指派問題的數學模式，並以旅客總步行距離最小為目標，再利用單體法(Simplex Method)來求解此靜態機門指派的問題，若是所求出的解非整數最佳解，則繼續利用分枝界限法(branch and bound)來求出整數最佳解。作者並參考中正國際機場的實際運作資料，在所設計的 10 種班機到達緩衝時間方案中，利用系統模擬的方式來進行動態即時機門指派作業，並探討班機到達緩衝時間設計對班機延誤的影響，結果顯示無緩衝時間的方案所產生的解果較差，因此，適當的班機到達緩衝時間可以降低擾動的影響。

朱橋榮(2001)則是以最小化機門數量作為其目標函數，並利用網路流動技巧來構建機門指派模式。其模式有「基本模式」、「混合大小型機門之營運模式」、「班機等候策略下的單型機門模式」與「班機等候策略和大小型機門雙重考量模式」，此篇文獻所發展之模式，是以其基本模式為基礎，故將其基本模式詳述如下，模式示意圖如圖 2-7 所示：

圖中的路徑 a，代表一機門於單日中的服務序列。路徑 a 上的流量依序流經節線段 1—2、5—6、3—4，代表航班 A、航班 C 以及航班 B 於當日依序停靠該機門。

其餘節線段，如 2—5 與 8—3，則代表該機門處於無航班停泊的狀態。彙總機場所有的可能的機門指派作業形成網路，再經由網路流動技巧來達成最佳化。圖中節線可以分為航班節線、航班接續線、起始及結束節線以及循環節線。分別敘述如下：

- (1)航班節線：如 1—2、5—6、3—4、7—8，表示此班機在此機門預訂停泊時間加上緩衝時間。
- (2)航班接續線：如 2—5、8—3，若此節線流量為 1，表示相連的兩航班，將先後停泊再同一機門。
- (3)起始、結束節線：如 0—1、0—3、0—5、2—9、4—9，代表當日各機門指派作業的起始或終了。
- (4)循環節線：起訖節點間加入一循環節線，將首尾兩點轉為流量守衡，循環節線的流量即為服務全部航班所須的機門總數。

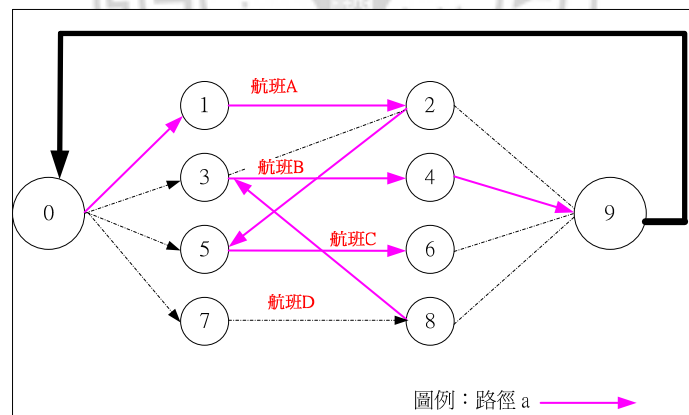


圖 2-5 單日機門指派作業網路圖

游守田(2004)，探討在班機不準時離到場或班機臨時更換機型情況下，機場航務單位調度機坪之策略，以多目標構建機坪即時性調度模式，以使得總旅客步行距離最小、航空公司地勤作業成本最小、班機等候時間最小，並收集中正機場航務組之班機指派資料作為實證。

2.2.3 小結

機門指派與船席指派的文獻如上所述，除了服務主要對象不同，一個為人，一個為貨，在目標式的考量上也不相同；此小結將兩者相異點整理如表 2-7。

表 2-7 船席指派與機門指派之比較

	船席指派	機門指派
主要對象	貨物	人
考量因素	船舶長度 船舶吃水 內陸運輸、裝卸設備、船舶特性	旅客之步行距離 旅客行李處理運送距離 航空公司飛航時刻表 機型與登機門之相容性
考量角度	航商 港埠管理當局	旅客 航空公司 機場管理當局
研究方法	單目標規劃 時空網路圖 系統模擬 知識庫系統	單目標、多目標 網路流動技巧 模擬模式(Hamzawi, 1986) 專家系統(Gosling, 1990)
目標函數	船舶的等待時間 船舶的作業時間 最小安全距離	旅客步行距離 旅客閒置時間 機門閒置時間 最小化機門數量
	皆有靜態與動態指派的探討	
其他	離散型、連續型 結合起重機指派	
主要目的	充分運用船席 船舶皆能迅速完成貨物裝卸工作	航機能在停機坪作業 旅客能在最短距離上下飛機

2.3 模糊及多目標規劃

船席指派的運作過程如 2.1 節所述，係結合各種現場因素下的決策，並考量「航商」與「港埠管理當局」立場，做出船席指派的決策，此與多目標下的均衡解概念類似。以傳統單一目標的作法，可能不能完全反應出問題的特性且給予決策者有較佳的依據，因此將會把船席指派問題朝向多目標規劃的方向來思考。

又考量到目標達成值之不確定性，有學者開始結合模糊理論配合多目標模式進行規劃。本節就文獻探討多目標規劃部份作一整理，作為模式構建之基礎。

2.3.1 簡介

在現實社會中，許多決策者面臨的問題常是多目標而非單一目標，譬如生產者想使收益最大，同時又想要使成本最小、存貨最少、生產能量使用最大等；都市管理者想使都市生產力最大，又想使都市環境品質最好，又能達到社會公平；運輸業者希望成本最小，又希望服務品質顧客最滿意。問題是這些多目標常常彼此衝突，很難找到每一個目標皆最佳的情況。換言之，改善一個目標值可能導致另一個目標值下降。

2.3.2 多目標規劃

多目標規劃根源於 Von Neumann and Morgensten 於 1944 年提出「數個衝突最大化問題的困境」之概念，Koopmans 於 1951 年提出運用效率向量(efficient vector) 處理多目標問題，隨後的學者如 Kuhn & Tucter 提出向量最適化(vector optimization)概念求解多目標問題，多目標規劃主要是在特定限制下，以數學規劃之技巧，求解一組折衷解，

Zeleny (1982) 曾指出，在單一目標規劃中並無所謂的決策，因為決策已經隱含在目標函數價值係數之估計當中，當此目標函數之係數一經決定後，決策者僅能對模型所求得之解予以接受或放棄，故決策者也無法由模式中獲得其他資訊。在規劃的過程中，傳統的單一目標規劃方法可能無法適切地描述或反映決策問題的真實情況，及滿足現代決策者的需求。代之而起的是真正能符合決策者或管理者需求的多目標規 (Multi-objective Decision Making; MODM) 理論。

多目標規劃的發展係若僅以單一目標來考量會有不周全之情事，決策者對於所得之解只有接受或拒絕兩種選擇。而多目標在決策的過程中可同時考慮多個決策目標的數學規劃，重點在於各目標衝突時之權衡得失 (trade-off)，當目標相互間有衝突時，讓決策者能有效地尋找到有用之非劣解及滿意之妥協解，國內有關多目標應用文獻整理如表 2-8。

表 2-8 多目標規劃應用文獻整理

沈進成，1989	分析營運計劃對大眾運輸服務績效之影響。 <u>利潤最大化</u> 、 <u>旅客平均旅行總成本最小化</u>
杜世文，1991	利用模糊理論，將 <u>運輸業者與顧客</u> 運送時間限制之間予以模糊處理，以提高 <u>顧客服務水準與運輸業者運送成本最小化</u> 為目標，建立一多目標規劃模式
楊博文，1991	求最適決策變數以擬定適當的公車營運整合計劃， <u>使用者每日總旅行時間成本最小</u> 、 <u>營運者每日總營運成本最小</u>
林秀璘，1992	考慮多目標問題與部份參數之不確定性，使營運計劃能符合實際應用之需要。營運者 <u>每日長期營運總成本最小化</u> 、 <u>使用者每日總旅行時間最小化</u>
霍俊明，1996	以零壹整數規劃方法方式，考量 <u>旅客步行距離</u> 、 <u>延後航次</u> 、 <u>取消航次</u> 等因素，以及機門指派限制，構建一單目標及二多目標模式。
黃木才，1996	以數學規劃法透過排程來輔助業者作車輛調度。 <u>降低車輛里程</u> 、 <u>減少行車成本</u> 、 <u>各車行駛里程差異最小</u>
張立偉，2001	考量搶修單位之 <u>旅行時間最小化</u> 、 <u>搶修時間最小化</u> 及 <u>災點之搶修風險最小化</u> 構建多目標式，並以模糊多目標 Maxmin 將多目標轉換為單一目標
洪子盛，2002	以多目標模糊規劃法求解消防站之最適設置，使得在兼顧 <u>效率</u> 、 <u>公平</u> 及 <u>被重複服務人次最大化</u> ，並且藉由模式了解增設消防站的各個目標的滿意度。
游文松，2003	考量經營業者與司機員兩層面，建立多目標 <u>人員與車輛排班模式</u> ，並發展以勤務為染色體之基因演算法進行模式求解作業。
游守田，2004	探討班機不準時下，以 <u>旅客使得總旅客步行距離最小</u> 、 <u>航空公司地勤作業成本最小</u> 、 <u>班機等候時間最小</u>

基本上，多目標規劃與單一目標規劃最主要的相異之處如下：

1. 單一目標規劃求取的是最佳解，即在某一價值衡量下之「最好」的可行解，決策者只能無條件地接受或拒絕此一最適解。相反地，多目標規劃則強調選擇彈性與可行方案彼此間的替代性，亦即規劃先規劃出一組非劣解，然後交由決策者進行比較選擇，如此決策者往往能從比較中獲得一些與決策相關之資訊；同時能避免單一目標規劃時，決策者與規劃者角色混淆之情形發生。
2. 多目標規劃可以同時處理多種不同的目標，如公斤、個、元、人等單位。在單一目標規劃時，如遇到不同單位問題時，則須換成同一單位方能處理，因此也增加了規劃者的主觀性。茲因個人之換算標準不同，甚至規劃與決策者之標準亦不同，因而產生不一致之換算結果；而多目標規劃則無此種困擾。
3. 多目標可以處理多個目標衝突之問題，例如：核四建廠增加能源與環保問題，然而，單一目標即使利用敏感度分析，其分析的本質仍是在某一目標水準限制之下，如何達到另一目標之極大而已，並無法有效處理此類目標衝突問題。
4. 多目標規劃可處理優先順序不同問題。例如：決策者希望獲利最高，市場佔有率最大，生產力能提高，且又能兼顧企業形象等目標。在此多目標中，決策者若能排定重要性或優先順序，規劃者即能根據此順序，求取最滿意的解。

唯有在面對多目標、多評準或多屬性之抉擇時，才有所謂決策的意義，多目標規劃方法即屬於此類。此法可分為三個主要步驟：(1) 確認及量化多目標函數 (2) 確定決策變數及限制式 (3) 產生及評估可行方案；其主要精神在於更廣泛地考慮決策者希望同時達成的多個目標，而有別於傳統上單一目標規劃只考慮一個目標函數的情況。

多目標規劃依是否包括不確定性因數可分為確定性多目標規劃與不確定性多目標規劃，前者包括多目標線性規劃、多目標非線性規劃與多目標混合整數規劃；後者依不確定性之描述方法分為機率性多目標規劃、模糊性多目標規劃、灰色多目標規劃及灰色模糊性多目標規劃等。

多目標規劃模型基本土是單一目標線性規劃之擴充，其不同之處在於多目標規劃可同時處理兩個或兩個以上之目標，而單一目標線性規劃只能處理一個目標，兩者相同之處為皆有決策變數(decision variable)與限制式(constraint)。

2.3.3 模糊多目標規劃

模糊數學規劃 (Fuzzy Mathematics Programming, FMP) 的觀念源自於 Belleman and Zadeh (1970)，之後，Zimmermann 首先將模糊集理論引入傳統之線性規劃問題，再將模糊線性規劃模型與多目標規劃結合成為模糊多目標線性規劃。Martinson 在 1993 年提出模糊多目標規劃方法的精神，係將限制式的模糊邊界定義為一個「符合限制滿足程度」的隸屬函數並轉移至目標式，再對每個目標式求出兩個參考值，同時定義一個「達成目標滿意度」的隸屬函數，最後將總目標式定義為同時滿足各滿意度的交集，再計算一次折衷解 (Compromise Solution)。當規劃所考慮之目標式並非唯一，而各目標之間的權重值無法得知時，其規劃方法可採模糊多目標中最小值最大化 (max-min operation) 運算方式，將多目標問題轉變為單目標以便於求解。其方法如下：

1. 決定各目標上下界

假設決策者對第 i 個目標 z_i ，有最滿意之理想值下界 (z_i^+) 與最不满意之不理想值上界 (z_i^-)，在決定上下界之方法上，可將目標視為可行解空間之函數，再由計算而得。如構建報酬表 (payoff table)，分別求取各目標函數達到單獨最佳化之條件，而其他所有的目標函數只達到部份滿意的程度，再從其中找出各目標在權衡關係中最佳與最差的值，定為上下界。報酬表之內容如表 2-9。

表 2-9 報酬表計算內容

步驟	第一目標式(z_1)	第二目標式(z_2)
求解 $\{\min Z_1(X) X \in FD\}$ 得 X_1	以 $Z_1(X_1)$ 為滿意度最高的參考點 Z_1^+	
求解 $\{\min Z_2(X) X \in FD\}$ 得 X_2		以 $Z_2(X_2)$ 為滿意度最高的參考點 Z_2^+
找滿意度最低的參考點	在求解 $\min Z_1(X)$ 及 $\min Z_2(X)$ 過程中，紀錄目標值最高者為 Z_1^-	在求解 $\min Z_1(X)$ 及 $\min Z_2(X)$ 過程中，紀錄目標值最高者為 Z_2^-

資料來源：[黃木才，1996]

2. 構建隸屬函數

對於第 i 個模糊目標之隸屬函數：

$$u_{Gi}(x) = \begin{cases} 0, & z_i \leq z_i^- \\ \frac{z - z_i^-}{z_i^+ - z_i^-}, & z_i^- \leq z \leq z_i^+ \\ 1, & z \geq z_i^+ \end{cases}$$

3. 建立決策集合之隸屬函數

決策集合之隸屬函數 $u_D(x)$ 為
 $u_D(x) = u_{z_1(x)} \cap u_{z_2(x)} \cap \dots \cap u_{z_k(x)} = \min\{u_{z_1}, u_{z_2}, \dots, u_{z_k}\}$

上式表模式的可行模糊集合為所有目標式之交集，由於決策者需要明確之決策提議，故需要此決策集合中隸屬程度最高的一組解，故取其最大值可得到一對應之隸屬函數：

$$u_D(x^*) = \max \min\{u_{z_1(x^*)}, u_{z_2(x^*)}, \dots, u_{z_k(x^*)}\} \geq \max \min\{u_{z_1(x)}, u_{z_2(x)}, \dots, u_{z_k(x)}\}$$

4. 將多目標轉化為單一目標求解

最後將原目標式以隸屬函數來表示，並以 1 代表 $u_D(x)$ ，則可將多目標規劃式以模糊的概念轉化為明確的線性問題求解。

$$\text{Max } \lambda$$

$$\text{s.t.}$$

$$\frac{z_i - z_i^-}{z_i^+ - z_i^-} \geq \lambda, \forall i$$

$$Ax \leq b$$

$$x \geq 0$$

2.4 車輛繞徑

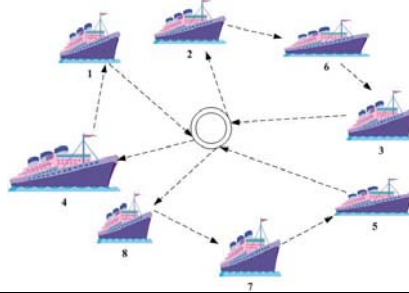
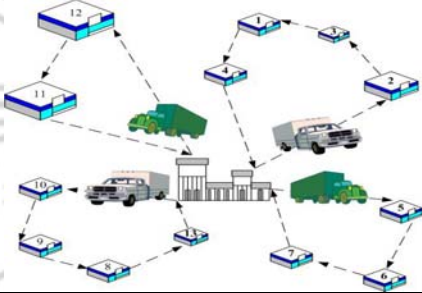
若是將船席指派問題中，船席視為車輛，船舶視為需求點，船席指派可以視為車輛巡迴於不同的需求點間，以期求出最佳的繞徑順序。因此，此小節先將船席指派與車輛繞徑問題做一轉換與簡略說明，詳細的說明與定義會於第三章的模式構建中呈現，並回顧車輛繞徑的相關文獻。

2.4.1 船席指派與車輛繞徑

船席指派的決策為什麼時間，船舶停於何處，而指派的結果可以從船舶的等待船席的時間、作業時間等績效指標去做評估，而車輛繞徑問題為車輛巡迴於不同需求點，以求出最小的運輸成本及違反顧客需求點時窗限制。

乍看之下，兩者似乎沒有很大的關連，但若是將船席視為車輛，船舶視為需求點，在預定到達的所有船舶中，找出最佳得服務船舶順序(繞徑順序)。因此，將船席指派問題概念稍作轉換後，可以以車輛繞徑模式來處理船席指派的問題，表 2-10 為船席指派與車輛繞徑的轉換對照表。

表 2-10 「船席指派問題」轉換為「車輛繞徑問題」之構想

船席指派	VRP 相對應內容
	
<p>船席指派概念：</p> <ul style="list-style-type: none"> ● 每一船舶由一船席服務一次 ● 每一船舶均被服務到(此處所謂之服務乃指貨物裝卸作業) ● 船舶等待時間最小 ● 港埠經營者使用成本最小 	<p>VRP 概念：</p> <ul style="list-style-type: none"> ● 每一顧客點只由一部車輛服務一次 ● 每輛車由場站出發後，再回到原場站。 ● 每輛車所服務的顧客需求量，不得超過車輛之容量限制 ● 每個顧客的需求點均須被滿足。 ● 尋求最短路徑
碼頭營運時段	場站(depot)營運最大時間
船舶靠泊碼頭之時間長度	各節點服務時間
船舶航次編號	各節點編號
船席服務船舶所產生的成本	節線成本
虛擬中心	場站(depot)
每一船席所服務之船舶順序	分群路徑
每一船舶只能由一船席服務一次	每一顧客只由一部車輛服務一次
船舶的 ETA、ETD	時窗限制(time windows)

2.4.2 車輛繞徑問題

車輛途程問題，在考慮一些資源（如車輛與時間等等），且在某些限制（如車輛容量、工作時間限制等因素）之下，每部車輛如何由配送中心出發，找出車輛最佳繞行路線，最後再回到場站，以達到配送成本之最小化。因此車輛途程問題乃安排車輛行駛路線最佳化之問題，而安排路線以及途程規劃為其兩大重點。

對 VRP 而言，其目標函數（Objective Function）通常為一般化成本之極小化。所謂一般化成本係將車輛配送過程中可能發生的各種負面效果予以數量化及幣值化。最常見的負面效果是運輸成本，又可分為「固定成本」與「變動成本」：固定成本係指車輛使用成本（例如，購買成本、折舊）及司機薪資；變動成本則與配送路線之排程有關（例如，路線行駛距離或時間、油耗、輪胎磨損、裝卸貨時間）。

在限制式（Constraints）方面，VRP 所面臨的限制條件甚多，共可概分成以下五種類型：

- (1) 網路結構：描述節點與節線之空間位向關係，例如方向性、完全性、對稱性、連結性、歐氏網路。節線成本通常為一固定值，某些特定情形下會考慮變動的節線成本（例如：尖峰與離峰時段）。嚴格來說，網路結構並不會以限制式的形態出現在數學規劃模式中，但是會影響目標函數之變動成本參數。
- (2) 節點服務：規範車輛服務顧客之方式，會影響解的基本結構，包括流量守恆、避免子迴圈及配送方式。
- (3) 顧客需求：指顧客的需求量或配送條件，會影響車隊規模及路線組成。分別為固定需求與隨機需求、不可分割、時間窗、配送頻率。
- (4) 設施資源：指場站、車輛及人員等配送資源之特性或限制，可以分為單一場站、多場站、單一車種、多車種，且有車輛載重限制。
- (5) 變數型態：指定數學規劃模式中各變數之型態，包括，實數（Real）、整數（Integer）及雙元整數（Binary Integer）。

隨著實務狀況與上述限制條件之不同，VRP 自 TSP 與 VRP 衍生出許多種更複雜的問題類型，例如：時間窗車輛路線問題（VRP with Time Windows, VRPTW）、多車種車輛路線問題（Heterogeneous Fleet VRP, HVRP）、週期性車輛路線問題（Periodic VRP, PVRP），等等。

各種類型 VRP 問題之間的差異並不大，往往只改變一或二個限制條件而已。儘管如此，不同類型問題的求解方法卻可能有南轅北轍之設計理念；限制條件的改變也可能使問題求解更加困難。

2.4.3 小結

上述說明瞭車輛繞徑與船席指派轉換與對照，主要是將船席視為車輛，船舶視為需求點；此外，車輛繞徑問題的目標式大都為最小化成本，其成本大致為車輛的固定成本、變動成本，與船席指派問題的對照，相當於碼頭機具使用所產生之成本，此可作為後續模式構建的基礎。

2.5 類免疫演算法概述

靈感來自生物免疫系統所衍生出來的類免疫演算法，發展至今雖不過才十幾年，卻廣受眾多學者們的青睞，不論在其理論架構上或實際應用上，皆有許多的書籍與文章論述之。在本節中將先會簡單介紹「生物免疫系統」，之後說明利用免疫系統概念所產生的「類免疫演算法」演算流程。

在最近幾年，有越來越多的研究都是以人類的免疫系統為基礎，來建構人工免疫系統進行一些生物演算法的發展與應用，主要有感測器診斷、生產排程 (Fukuda 1998)、訊號的不規則偵測 (Dasgupta 1997) 與二元辨識 (De Castro 2000) 等，故人工免疫系統可以應用的領域十分廣泛，免疫系統、神經網路以及基因演算法三種都是仿生物系統的智慧型演算法，它們利用生物的學習與運作的特性，量化形成了資訊處理系統的模式。

在生物免疫系統中是利用兩種機制來完成免疫的工作：一種是免疫網路，免疫網路的功能在於免疫細胞間的所有互動，主要是抗體 (Antibody) 與抗原 (Antigen) 及其它免疫細胞之間的合作；另一種是組織相容複合機制，其功用在於分辨外來分子或物質，是否和自己本身的細胞有所差異。

2.5.1 生物免疫系統介紹

在我們的生活環境中有許許多多各樣的感染性微生物，例如病毒、細菌、原蟲等等，其中任何一種微生物都可引起病理性的傷害，甚至會殺死宿主。在健康人的身上，絕大多數的感染均屬於短暫性，極少留下永久性的傷害，這是因為人體內的免疫系統 (immune system) 能夠有效地對抗感染性病原體 (infectious agents)。

當人體遇到某些病原體的刺激或感染時，體內會自動產生保護性的反應能力，這種反應能力稱為『免疫反應』 (immune-response)。這些入侵人體的病原體稱為抗原 (antigen)，而保護人體的物質則稱為抗體 (antibody)，又稱為免疫球蛋白 (immunoglobulin) (Roitt et al., 1998)。

免疫系統大致包含兩道防線：

1. 第一道防線：非特異的天然免疫反應 (innate immune response)，包含皮膚、黏液、巨噬細胞 (macrophage) 等天然屏障。當一個感染性的病毒 (抗原) 入侵到人體時，體內先天性反應的巨噬細胞或專業的抗原呈現細胞 (如樹狀細胞等) 便會將病毒 (抗原) 吞噬。

2. 第二道防線：適應性免疫反應(specific immune response)，包含細胞性 (T 細胞)的免疫和體液性(B 細胞)的免疫兩大部份，通常他們具有記憶性，是透過淋巴細胞來發揮免疫作用。

生物免疫系統主要如上述的兩種免疫反應，其他像是株落選則、T 細胞等介紹詳見附錄七，此一小節主要是先對生物之免疫系統之反應有一概觀的認識，可以作為後續演算法的對照。

2.5.2 類免疫演算法

1. 類免疫演算法之特性

一般而言，許多專家學者發展類免疫演算法至今，主要是取生物體免疫系統的幾個特性，特性說明如下，並將此概念與應用到類免疫演算機制中的法則比較，如表 2-11 所示。

(1) 獨特性與專一性：

免疫系統的反應不是隨意發生的，是針對不同的病原而發生的；抗體的受器只能與一種相對應的抗原決定部位(epitopes)結合而誘發免疫反應；這相當於類免疫系統中，對特定的求解問題，進行類免疫演算機制，且求得之解答僅針對解決問題的本身。

(2) 適應性：

生物體的免疫系統能對所有各式各樣侵入的病原體，做出專一性的免疫反應，除此之外，甚至對以前從未出現過的新病原體，亦可作出適當的免疫反應，來保護生物體；這相當於類免疫演算法能適用於各種不同領域的問題解決上。

(3) 分散式偵測：

生物體的免疫系統不需要由某個器官來集中控制或管理，部分參與免疫系統反應的細胞，即可自行偵測侵入之病原，並刺激其他免疫系統的成員，產生適當之免疫反應。這相當於類免疫演算法有別於基因遺傳演算法，可平行演化至答案，無須受限於整體演化的趨勢。

(4) 學習與記憶性：

生物體的免疫系統在感染過某種病原體而痊癒後，會記憶該病原體，並於下次再感染時，能夠立即反應消滅與抑制之。這相當於類免疫演算能夠紀錄每一次搜尋到的可能解答。

表 2-11 生物免疫系統與類免疫演算法特性對照表

	生物免疫系統	類免疫演算
專一性	抗體受器只能與一種抗原反應。	求得之解答僅針對解決問題本身。
適應性	對以前從未出現過的新病原體，亦可作出適當的免疫反應。	適用於各種不同領域的問題解決上
分散式偵測	參與免疫系統反應的細胞，即可自行偵測侵入之病原，並刺激其他免疫系統的成員。	有別於基因遺傳演算法，可平行演化至答案，無須受限於整體演化的趨勢。
學習與記憶性	下次再感染時，能夠立即反應消滅與抑制之。	紀錄每一次搜尋到的可能解答。

2. 類免疫演算法之架構(Artificial Immune Algorithm)

類免疫演算法是利用免疫反應的特性所衍生出來的，與基因演算法的概念類似，都是經由學習模仿生物的行為或是反應而成的演算法。藉由免疫反應中抗體辨認抗原的專一性、記憶性，將問題的目標視為抗原並將問題的解視為抗體，針對抗原目標作搜尋，尋找問題的抗體解答。

族群為群聚個體的集合，在生物的免疫系統中，通常已經有許許多多種的抗體，基於此種概念，類免疫演算法中是以隨機的方式創造初始世代，接著產生一個新世代，直到收斂得解。

免疫演算法中有明確的目標，當有抗原侵入時，先會啟動第一道免疫防線，(隨機產生合乎問題特性的解)，以對抗原進行消滅的動作。在第一道防線啟動後，接著會進行第二道之適應性免疫反應(搜尋其他可行解)，進行株落選擇後，增殖為記憶細胞與漿細胞，近以產生更多的抗體來對付抗原。

圖 2-6 為類免疫演算流程圖，在圖的上半部為生物的免疫系統。首先免疫系統辨識入侵的抗原，並檢驗抗體的親和力，到此一步驟為天然之免疫反應，並對適合度(親和度)較佳的個體進行選擇與記憶。

之後仿照生物免疫系統中的株落選擇，無性增殖為兩個族群，並對此兩族群進行兩種運算，其中一個族群進行抗體基因 V-J 和 V-D-J 的隨機重組(抗體變異區上的變化)，另一個族群則進行受體多樣性的演變(受體是由一些基因的區段和函式所構成的，它是由不同的函式當中隨機組合而成，產生指數的可能組合，所以有許多不同的受體結構)，在上述兩種運算過後，依據所設定的比例，以隨機的方式將兩個族群取出抗體，以維持一開始所設定的族群數，最後測試是否已設定的世代數，若是，則結束，若否，則回到親和力的計算步驟。

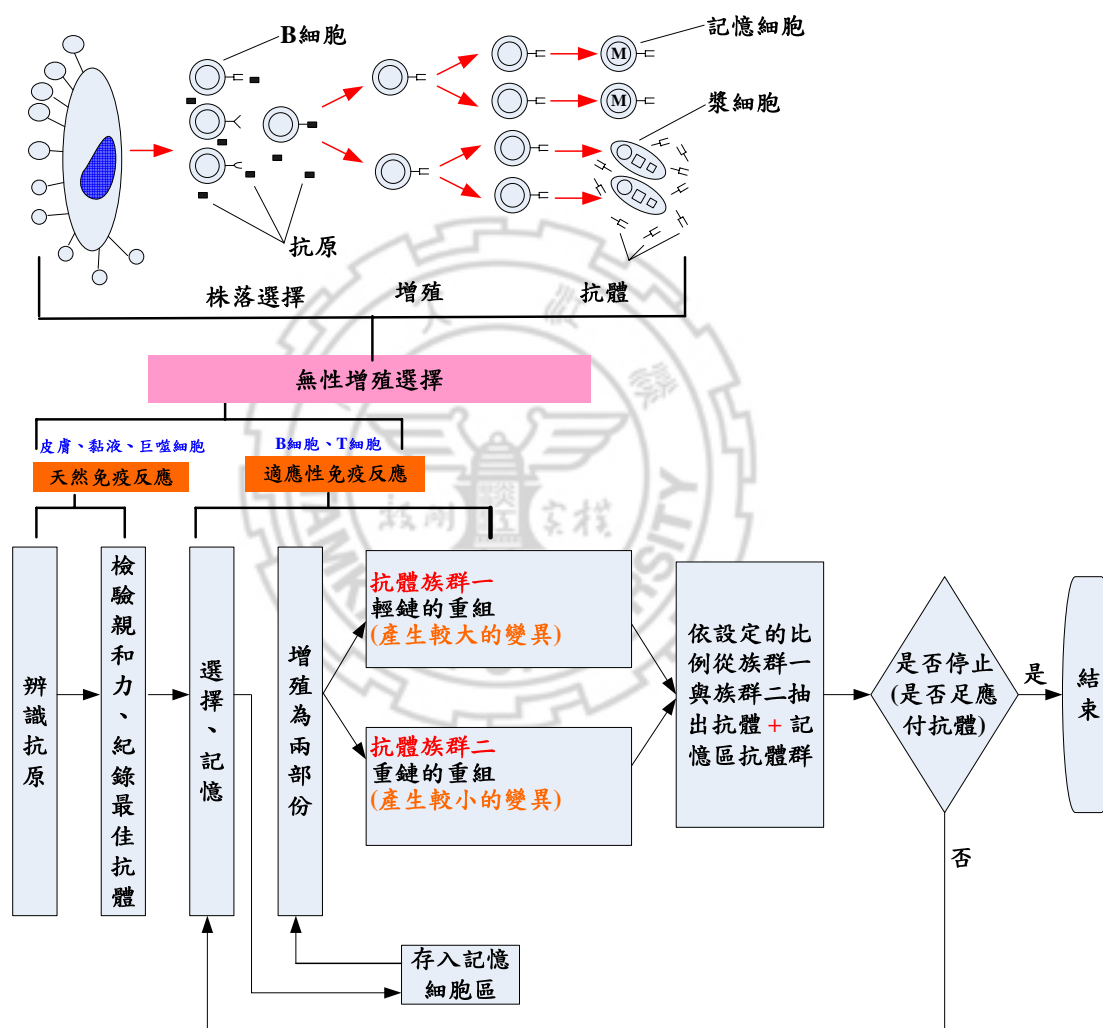


圖 2-6 類免疫演算流程圖

3. 類免疫演算法之流程

類免疫演算法的基本原則與程式步驟如下所述，並將本演算法之演算機制名詞與生物免疫系統成員作成如表 2-12。

(1) 第一道天然免疫反應

辨識抗原，隨機產生初始抗體族群，進行免疫系統的第一道防線，並將世代數設定為零， $t = 0$

(2) 計算抗體的親和度與細胞記憶

將抗體所代表的變數字串解碼成數值，並將這些數值代入目標函數而得到各染色體的親和度，並將最佳的抗體選出存入記憶細胞。

(3) 株落選則

依據所設定的抗體數目，配合蒙地卡羅的概念，選擇親和度較佳的抗體複製到記憶細胞區 A，並將記憶細胞區 A 進行細胞分裂，進行第二道適應性免疫反應：

此部份依「變化規模的大小」分為輕鏈及重鏈的重組這兩種形式進行：

① 輕鏈的重組：第一部份進行抗體受體多樣性演變，此部份將抗體進行大範圍的改變，方式為抗體與抗體之間的交互作用(如基因轉置或取代)，最後依親和性的比率、設定的抗體數目，選出較佳的抗體，儲存於記憶細胞區 B。

② 重鏈的重組：第二部份進行抗體本身的基因重組，此部份將抗體進行小範圍的改變，例如突變等，原則為不要大幅更動抗體本身，最後依親和性的比率、設定的抗體數目，選出較佳的抗體，儲存於記憶細胞區 C。

(4) 形成新抗體族群

將記憶細胞區 A、記憶細胞區 B、記憶細胞區 C 的抗體形成一個新族群

(5) 重複演算

完成上述步驟後，跳回步驟(2)，繼續執行下一動作，如此一直反覆進行到所設定的世代數 T。

表 2-12 類免疫系統與生物免疫系統之對照表

原始問題	類免疫演算法
問題的目標	抗原
問題的解	抗體
目標值評估的指標	親和力、結合強度
選取較佳的的解	株落選擇(clonal selection)
目前的解產生較大的變化	輕鏈的重組
目前的解產生微小的變化	重鏈的重組

2.5.3 類免疫演算法相關研究

許多學者利用上述免疫系統的特性進行對各種領域的應用，除此之外，也有越來越多利用電腦模組來模擬免疫系統中的內容，並且以生物學的觀點來模擬免疫的行為，因此，有許多科學或工程的應用以免疫為基礎。表 2-13 整理出利用免疫系統在各個領域應用的文獻。

雖然這些衍生自生物免疫系統的演算方法皆可稱作類免疫演算法，但依對於生物免疫反應所著重觀點的不同，所發展出來的演算流程也迥然不同。



表 2-13 類免疫應用相關文獻

Hoffmann, 1986	比較過神經系統與免疫系統，並將許多相似之處列出。且其假設一種對稱性的神經網路模式，其模式所產生預期刺激反應行為與免疫反應相似。
Mori et al., 1993	Mori 等人於 1993 年也提出了一個處理函數最佳化問題的類免疫演算法，這個演算法能夠實現免疫的多樣性，體現網絡理論和 clonal selection。他引入了一個抗體群多樣指標，並把多個解決方案的屬性作為系統記憶保留下來。
Dasgupta, 1997	提出類神經網路和免疫網路兩者由生物系統所發展出來的方法，它們都具有學習與記憶功能，所以常常被應用於解決辨識或分類的任務。但是辨識和學習是不同的機制，免疫網路提供不同的全域或區域的收斂最佳解，免疫網路處理自組織的記憶並且記住本身的分類有較長的時效性。
Dasgupta, 1997	又提出理論和數學模式來解釋生物免疫系統的現象，並且以生物學的觀點來模擬免疫的行為，所以有越來越多科學或工程的應用以免疫為基礎。免疫網路系統在各個領域之應用包括：電腦安全維護、時間序列資料的不規則偵測、錯誤診斷與人工免疫辨識系統。
Mori et al., 1998	Mori 等人提出了排程問題包含兩個子問題，一個是批量大小、另一個是訂單順序，提出了一套演算法模擬實際的免疫系統，這個演算法以抗體理論和網路假設為基礎，利用這套演算法找出排程批量和訂單合併的最佳解，並且發展出一個適應動態環境下的生產系統之最佳化排程系統，但是在其研究當中並沒有對動態環境有太多的研究。
de Castro, 2000	de Castro 等人 探討免疫系統當中株落選擇理論以及學習和記憶的功能，最後再加入成熟的概念。 接下來提出株落選擇演算法，並將該演算法應用於字元辨識、多值域最佳化與旅行推銷者問題。
Shenglian et al., 2000	亦有部份學者為解決基因遺傳演算法於最佳化應用上的缺點，而將類免疫演算法的部份機制加入基因遺傳演算法中，發展出一套 Immune genetic algorithm (IGA)
de Castro, 2002	定義類免疫系統為一種計算方法，奠基於自然界免疫系統的象徵

2.6 禁忌演算法概述

2.6.1 禁忌搜尋法簡介

禁制搜尋法 (TS) 的觀念架構最早是由 Glover [1986] 及 Hansen [1986] 所提出，專門用來求解組合最佳化的問題，其應用領域包含排程、車輛途徑、通訊、圖形著色等方面。經過多年來的發展與演進，TS 法已經成為當代最著名的巨集啟發式方法之一。TS 法的理念是想構建一個智慧型的問題求解程式：在現有解的鄰域進行搜尋，並應用人工智慧的記憶機制，將已經搜尋過的解及其特徵記錄在禁制列 (Tabu List)，以避免重複性或毫無目標的搜尋；等到整個鄰域都搜尋完畢後，再選擇一個最佳的方向進行移動，以逐漸逼近最佳解。

在搜尋過程中，依鄰近解的定義方式，每一個解都會存在著相對應的鄰近解，而將目前的解轉換為其中一個鄰近解的步驟，稱之為“移動” (Move)。塔布搜尋法的主要的精神就是藉由彈性記憶架構 (Flexible Memory Structure) 記錄前幾次的移動，避免尋找最佳解過程中，又恢復到過去曾經拜訪過 (Visited) 的解，重複相同的路徑，形成迴圈 (Cycle)。這種短期記憶結構稱之為塔布列表 (Tabu List)。而當候選的移動方向已被記錄在塔布列表時，稱這一個鄰近解被塔布限制住 (Tabu)。塔布列表記錄的法則是採先進先出法 (FIFO)，亦即下一個新的移動路徑進來，就把串列中最舊的移動路徑移去，只存放最近幾次記錄，釋放後的屬性又可以再度被安置與拜訪。

2.6.2 禁忌演算法

禁忌演算法相關重要名詞說明如下：

1. 移步 (Move)：禁制搜尋法首先藉由隨機或是其他方式產生一起始解，藉由各種移步方式搜尋其鄰近解 (Neighborhood)，並選擇最好的鄰近解進行移步的動作，而目前常用的方式為 2-opt/3-opt，如圖 2-7 所示：

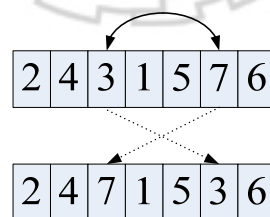


圖 2-7 兩兩互換之移步方式

2. 候選名單 (Candidate List)：候選名單即為可以進行移步屬性的集合，候選名單內的每一個被允許的移步皆對應一鄰解，禁制搜尋法中的移步，即是由這些名單中挑選最好的鄰解，進行移步的動作。
3. 禁制名單 (Tabu List)：在進行移步的同時，禁忌名單會將其移步的屬性紀錄下來，以作為未來進行移步時參考依據；而禁忌名單的大小，決定該移步被禁制的期間，當禁制的期間愈長時，愈能避免在求解過程中可能發生的循環，因此較不容易陷入區域的最佳解。

4. 渴望準則 (Aspiration Level)：當某一移步被禁制時，但其所獲得的解優於目前搜尋所得之最佳解時（設定為當目標函數值獲得改善），則透過渴望準則將其禁制的狀態解除，使該移步可以進行。
5. 搜尋停止準則 (Stopping Criterion)：搜尋停止準則通常有預設運算過程中最大允許遞迴數、預設目標函數值持續未改善之最大允許遞迴數及預設允許 CPU 計算時間這三種。預設達到可接受目標函數值範圍；一但達到預設的範圍即停止搜尋，而所搜尋的最佳尋獲解即為最佳解。

一般基本禁忌搜尋法的主要演算流程，可分為起始解的求取、鄰域的定義、禁忌名單的設計、與移動後的評估。首先以一起始解作為搜尋的起點，接著進行搜尋的程式。在搜尋的過程中，為避免產生循環的現象，故建立禁忌名單(Tabu List)來記載搜尋的過程。禁忌名單的結構特性及其長度，可避免求解的過程之中，陷於區域解的現象。但禁忌名單可能會限制了求解的範圍，所以必須運用免禁準則來加以克服。其步驟如下所述：

- (1)：於可行解當中任意找出一組解，作為起始解 x_0 ，並成為目前解 x ，最後再令此一起始解為目前之最佳解，即 $x^* = x_0$ 。設定禁忌名單為空集合。
- (2)：自目前解 x 的鄰域中，選出最佳的鄰域解 x' 。
- (3)：檢查此一移動 $x \rightarrow x'$ 是否存在於禁忌名單中，若是則到步驟(5)，否則繼續步驟(4)。
- (4)：將此移動記錄至禁忌名單中，並判斷 x' 是否優於 x^* ，若是則以 x' 取代 x^* 。並重新回到步驟(2)中。
- (5)：檢查此移動是否符合免禁準則，若是則解除此禁忌名單並移動，再回到步驟(2)繼續搜尋，否則繼續步驟(6)。
- (6)：檢查是否到達停止準則。若是則停止演算，否則令次佳解為目前解，記錄禁忌名單並回到步驟(2)。

2.6.3 禁忌演算法相關應用文獻

禁忌演算法之相關應用文獻如下：

Renaud, Laporte 和 Boctor (1996) 考慮在車輛容量及路線長度的限制的情況下，應用禁制搜尋法求解多場站的車輛巡迴路線問題 (Multi-Depot VRP)，並發展一個演算法 FIND，在該演算法中包含了 Fast improvement、Intensification 及 Diversification 三個階段，而每個階段都包含了 1-route、2-route 與 3-route 移步法則；該演算法測試了 Chrisofides, Eilon, Gillette 和 Johnson 的 11 個典型的問題，與 Chao et al. 的 12 個新的問題，發現該演算法能計算出較好的解。

Brandao 和 Mercer (1997) 以禁制搜尋法求解多旅次車輛巡迴路線問題與排程問題 (multi-trip vehicle routing and scheduling problem, MTVRSP)，該問題基本的限制與一般車輛巡迴路線問題並沒有太大的不同，額外的限制如下所述：(1)每輛車一天之中允許被使用一次以上；(2)各需求點有時窗限制；(3)考慮不同容量的車種；(4)特定的需求點必須由特定的車輛服務；(5)必須考慮到駕駛人的排程；(6)考慮卸貨的時間等。

Barbarosoglu 和 Ozgur (1999) 應用禁制搜尋演算法求解車輛巡迴路線問題，並在土耳其以一實例進行求解，該篇文獻主要探討一新的鄰近搜尋法稱之 TANE 與 TANEC，TANE 主要是給定一搜尋範圍，以目前的解進行搜尋，其中最佳的非禁制移步 (non-tabu move) 是被決定的；而 TANEC 只要是尋找最佳的非禁制移步，並找出最小的目標函數值，其中移步的方式皆是採用 2-opt 的方式進行。

林明俊 (1998) 在車輛巡迴路線問題中考慮了隨機的需求、隨機的運送時間和不同的車輛容量，應用禁制搜尋法求解隨機環境下多車種派車的問題，但由於作者未能從模式中找出確切解，因此僅針對較小的例題進行驗證，但無法得知解的品質。

王文貞 (1997) 使用兩階段求解一個圖書配送車輛排程問題時，第一階段以節省法產生起始解，第二階段則以插入法與節點交換法做為路線改善的方式，在模式正確性的驗證上，作者針對假設性的路網資料，以 LINDO 進行最佳解的求解，進行比對，在四個範例中，有三個可達到最佳解，而未達最佳解的範例，其誤差為 3.1%。

黃金智 (1999) 求解一個隨機的需求量、車輛數有限之多車種隨機車輛途程問題，該研究以最鄰近解法 (Nearest Neighbor) 作為禁制搜尋法之起始解，而利用 2-exchange、2-swap 或插入法為搜尋鄰近解的方式。

第三章 問題定義與模式構建

本章先對「船席指派」問題做一描述，讓我們對於船席指派現況有一概略的瞭解，接著以「車輛繞徑」的概念構建模式，以定義出符合現況的解題模式。

3.1 問題定義

船席指派決策在實務上係以會議的方式進行，此方式須要由港埠經營者、航商出席會議決定，此種方式相當耗時耗力。適當的指派對於航商所產生的影響，莫過於在港的時間停留的長短，而對於港埠單位而言，適當的指派可以使港埠資源組合達到有效的利用、總成本最小化經營，為了同時滿足此種兩種目標，本研究會以多目標規劃的角度來構建模式，以反映現實世界的決策。

3.1.1 問題特性與描述

以國內目前有關港口的裝卸貨櫃而言，貨櫃船舶數往往大於港口的貨櫃碼頭數，因此會有船席不敷使用的情況發生，造成某些船舶必須在港外等待的情況發生。在這樣的情況下，如何在對最短的時間內完成裝卸貨作業成為一個相當重要的課題，作業效率決定了一個港口的競爭力，也是港務局相當重視的績效之一。然而，目前國內在此方面作業流程上仍是以人工指派為主，可能無法有效地降低船舶等候時間，因此，找出一套有效的方法來求解船席指派的問題是相當值得探討的。

船席指派在實務操作上，在考量相關的實體環境限制下，如碼頭水深、船舶吃水、裝卸設備等條件，並經由船席指派會議來決定船隻停泊的位置。

圖 3-1 為從時間的觀點來看「單一船席」指派後情形，由圖可以得知，船席部分時段是處於閒置狀態，部分時段處於船舶停泊狀態(裝卸貨)，此外，從圖中可以得知船舶的到達時間、開始接受服務時間點、等待時間、作業時間與此船席的指派順序。

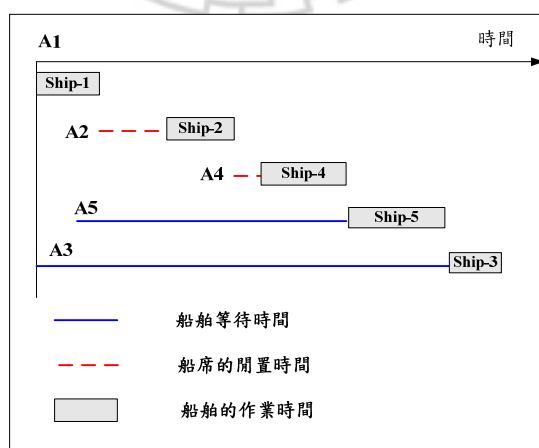


圖 3-1 單一船席指派示意圖

為使得港埠相關資源有效的運用，以及船隻的在港時間最小，提高港口的服務品質，在考量相關限制下，進行船席的指派作業；由於每個船席上起重機數量及船席後線作業環境不同，因此，同一艘船舶於不同的船席上會有不同的作業時間；另外一方面，港口每項資源的成本也不盡相同，故本研究將會將港埠設備的成本反映在本研究所構建的模式中。

在此將問題特性整理如下，示意圖如圖 3-2 所示：

1. 實務上，船席指派乃是由「港埠當局」與「航商」經會議所做出的決策。
2. 模式建立的主要角度是從「經營者」與「顧客」兩個角度為出發點，以此方式思考較能反映現實的狀況。
3. 船席指派的考量條件有船舶長度、水深、內陸運輸等條件(詳見 2.1.1 節)。
4. 船席指派的原則是使充分運用船席，使船席的空閒時間減至最少，並使泊於船席之船舶皆能迅速的完成貨物裝卸工作[王丘明，2005]。
5. 針對貨櫃船而言，港埠經營者所產生的成本有港灣費用、棧埠作業費用等(詳見 2.1.3 節)，例如橋式起重機的成本、船席作業人員的成本、電力等。
6. 對於顧客(需求點)而言，指派所產生的影響，可以從港埠績效指標方面來進行分析，像是港灣績效指標、船席效率指標等方面來評估，例如船舶等待時間、船舶平均服務時間、在港時間等。
7. 本研究以「船舶在港的總時間」、「港埠管理單位所產生的成本」兩方面來構建模式，以反映出經營者、顧客的關係。
8. 本研究模式構建上的主要觀點，是以指派後所產生的影響進行考量。主要方式為觀察指派後對於參予船席指派決策的角色有何種影響。兩個主要的角色，即「航商」與「港埠管理單位」，過去的文獻探討，大都考量如圖 3-1 指派後的在港時間或等待時間，並未對於港埠管理單位有進行探討，因此，本研究僅以不同的船舶對於船席(港埠管理單位)會有不同的操作成本表示之。
9. 此方式主要觀點，並不是在強調營運收入、成本、利潤之間的關係，而主要表達的概念為港埠管理單位也是船席指派的主要參予者。以此方式除了可以突顯船席指派的另一參與者，另一方面也代表船席會因應不同的船舶，而產生不同的資源使用組合、相對的成本。
10. 與船席指派相當類似的機門指派，過去的文獻探討會考慮緩衝時間，所謂的「緩衝時間」主要目的是在吸收營運時航班到離的延誤，如朱橋榮(2001)；本研究的模式中，主要觀點不是強調在此一部分，因此，並沒有考量類似機門指派中的緩衝時間。

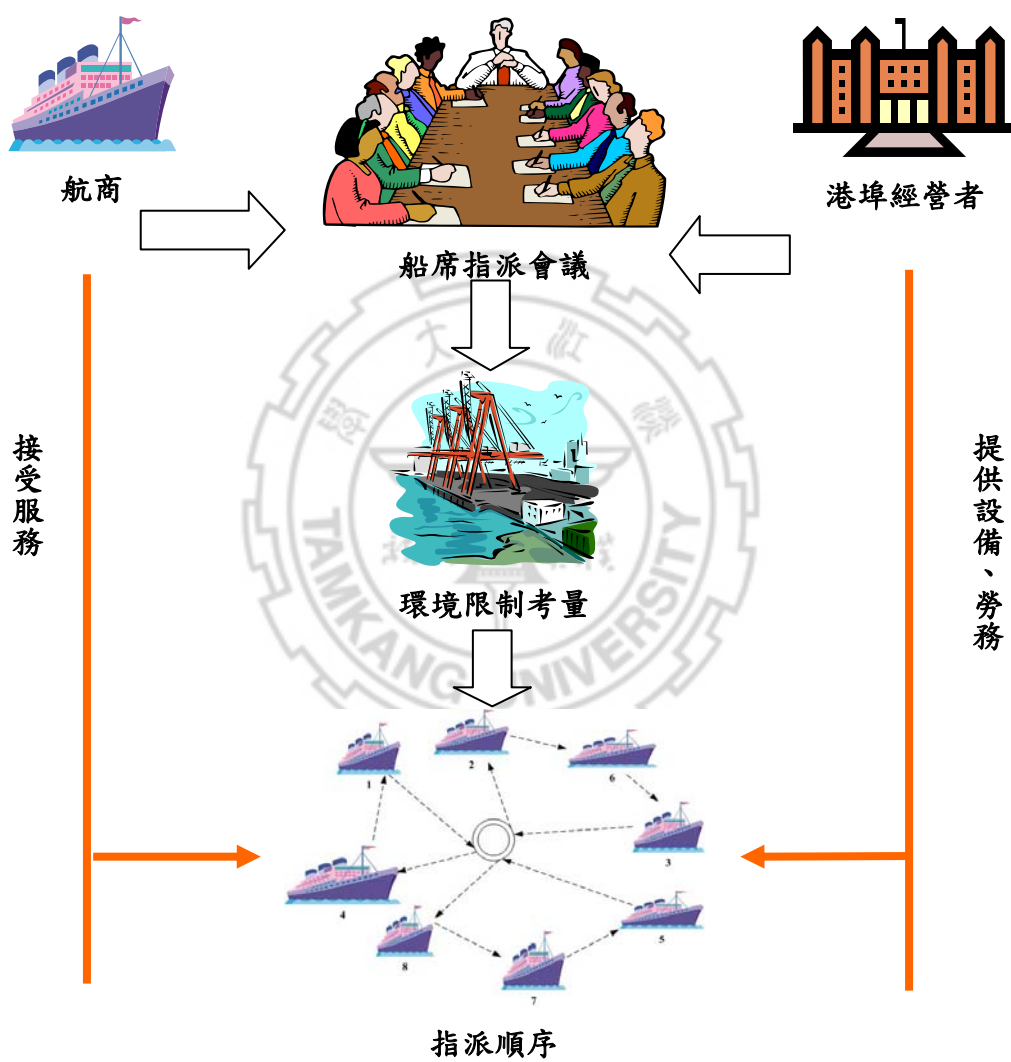


圖 3-2 問題描述示意圖

3.1.2 問題定義

依據問題的描述，本研究將船席指派問題定義如下：

有 n 艘船舶(船舶集合 N)， k 個船席(船席集合 K)，每艘船舶的作業時間 t_i^k 已知，且作業時間會隨船席的不同而有所改變，即相同一艘船舶於不同船席上的作業時間會有所差異，而每艘船舶有其服務時間限制 $[a_i, b_i]$ (時窗限制)，船舶開始接受服務的時間點為 T_i^k ；而每個船席有相對應的成本 c_i^k ，且此成本會與所服務的船舶有關。本研究模式在考量港埠當局立場、航商立場來求出在港時間與使用成本最小化。

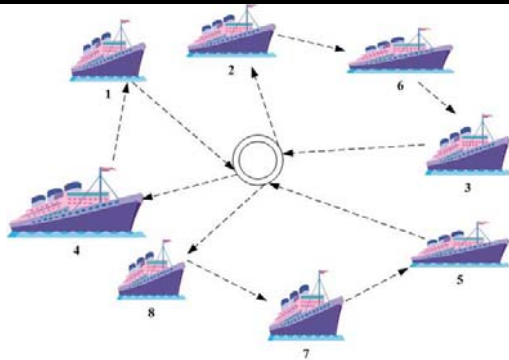
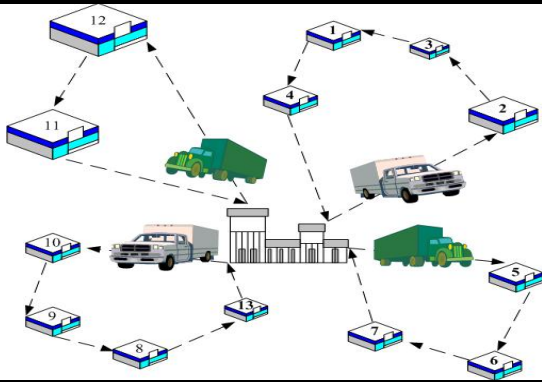
3.1.3 問題的轉換

由問題定義，可以將船席指派問題，轉變為單一場站，多部車輛、含時窗限制的車輛繞徑問題，有關船席指派問題轉換為車輛繞徑問題的對應內容如表 3-1 所示，說明如下：

車輛繞徑的問題大都應用在貨物運送的情況，其應用的環境與本研究不同，本研究為探討船席指派的問題，故在問題的定義上，稍有不同。在車輛繞徑中，會有單一場站或是多場站的情形，是真實的場站，在船席指派中，是假設為單一場站，且此場站為一虛擬的中心。在車輛繞徑中，「真實」車輛巡迴於不同的需求點間，但在船席指派中，我們把「船席」當成車輛繞徑問題中的車輛。

實務上，不同的船席上，有不同的作業機具、作業人員，且每個船席的操作、維修、變動成本也不盡相同，此與多車種車輛繞徑問題類似，因此我們將船席指派問題視為多車種的車輛繞徑問題；繞徑問題中，例如 TSP 與 VRP 等，車輛運送完畢後須返回場站，但在船席指派中，由於並沒有所謂的「實體車輛」，因此沒有所謂返回場站的動作；車輛繞徑問題中，以車輛容量限制作為「分群」的依據，船席指派在轉換為車輛繞徑模式時，是以「港口營運的最大時段」作為分群的依據，因此，假設每個船席的營運時段均相同。

表 3-1 「船席指派問題」轉換為「車輛繞徑問題」之構想

船席指派	VRP 相對應內容
	
船席指派概念： <ul style="list-style-type: none"> ● 每一船舶由一船席服務一次 ● 每一船舶均被服務到(此處所謂之服務乃指貨物裝卸作業) ● 船舶等待時間最小 ● 港埠經營者使用成本最小 	VRP 概念： <ul style="list-style-type: none"> ● 每一顧客點只由一部車輛服務一次 ● 每輛車由場站出發後，再回到原場站。 ● 每輛車所服務的顧客需求量，不得超過車輛之容量限制 ● 每個顧客的需求點均須被滿足。 ● 尋求最短路徑
碼頭營運時段 船舶靠泊碼頭之時間長度 船舶航次編號 船席服務船舶所產生的成本 虛擬中心 每一船席所服務之船舶順序 每一船舶只能由一船席服務一次 船舶的 ETA、ETD	場站(depot)營運最大時間 各節點服務時間 各節點編號 節線成本 場站(depot) 分群路徑 每一顧客只由一部車輛服務一次 時窗限制(time windows)

3.2 前提與假設

藉由第二章文獻回顧，使我們對於研究主題「船席指派」的特性與運作情況有所瞭解後，將以車輛繞徑的概念構建最佳化船席數量之模式；然而影響船席指派問題的因素眾多，想要將所有的因素都納入模式中來處理是相當複雜的，加上各因素的重要性與量化的困難度都不盡相同，因而列出模式的假設條件，相關的假設與前題如下：

1. 船舶、船席的數量為已知。
2. 每艘船舶只能被一船席服務一次，船舶時窗限制為已知。
3. 每一個船席只能服務一條路徑(服務順序)。
4. 船舶的到達港口的時間點以預定到達時間(ETA)為準，且假設不會有誤點的情況發生。
5. 規劃之時，假設所有碼頭處於閒置狀態，沒有隔夜之船舶，即目前沒有任何船舶停在船席上。
6. 假設需求點的作業時間會隨車輛的不同而有所變異，即船舶的作業時間與所在的船席作業環境有關，為確定已知條件(Deterministic)，如圖 3-3 所示，相同路徑下，需求點的作業時間 t_i^k 會隨車輛 k 的不同而有所變異。此方面之假設主要因為每個船席的資源不同，如起重機、人力，因此對於船舶之作業時間也隨之不同。

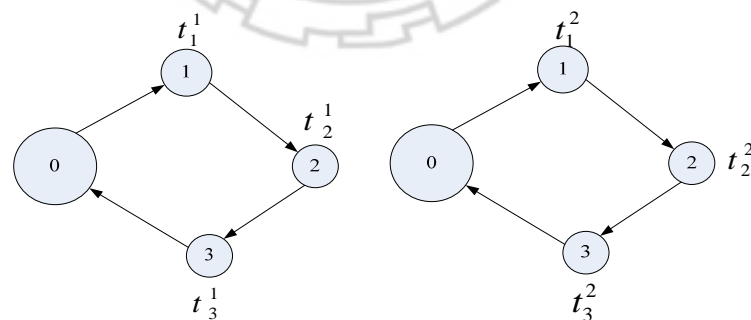


圖 3-3 節點作業時間說明圖

7. 在車輛繞徑問題中，節線的成本通常為旅行距離或時間，即資源使用所產生的成本。本研究將修改車輛繞徑模式中節線的定義，使其與研究主題配合。在本研究模式中節線的成本 c_i^k ，被定義為服務某船舶所產生的成本，且假設此成本會隨船舶的不同而有所改變。如圖 3-4 所示：在船席指派中，若船席 k 服務船舶順序為 $1 \rightarrow 2 \rightarrow 3$ ，則於轉換後的車輛繞徑模式中，節線的成本 c_1^k 、 c_2^k 與 c_3^k 分別代表船席 k 分別服務船舶 1、2、3 的成本，而 c_0^k 的成本視為 0。

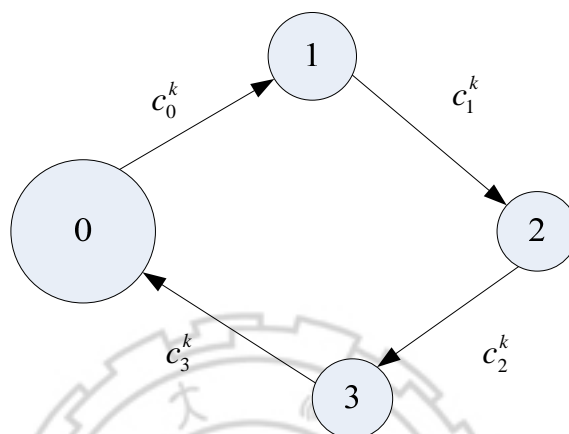


圖 3-4 節線成本說明圖

3.3 問題架構

本研究所定義的問題架構整理如下：

輸入：船舶的到達時間、作業時間、船席作業成本。

目標：最小作業成本、最小等待時間。

限制：流量限制、路徑時間限制、時窗限制、避免子迴路限制。

輸出：每個船席之船舶指派順序。

3.4 模式構建

一、符號說明

參數說明

K ：車輛的集合，即船席的集合， $K = \{1, 2, \dots, k\}$

N ：需求點的集合，即船舶的集合， $N = \{0, 1, 2, \dots, n\}$ ，其中「0」表示為船席指派中的虛擬場站

S ：需求點的次集合，船舶的次集合，即 N 集合中任兩個以上節點所形成之集合，非空集合

A ：節線的集合， $A = \{i, j\}, \forall i \in N, \forall j \in N$

$A(S)$ ：屬於 S 集合之船舶節點所形成的節線集合

c_i^k ：節線的成本，即船舶 i 在船席 k 的路徑上所產生的成本，本模式以車輛繞徑的角度解釋船席指派。

t_j^k ：需求點的作業時間，即船舶 j 在船席 k 上的作業時間

a_j ：需求點的時窗限制，即船舶 j 的時窗下限

b_j ：需求點的時窗限制，船舶 j 的時窗上限

s^k ：港口營運開始時間點

e^k ：港口營運結束時間點

WD_k ：船席 k 水深

D_j ：船舶 j 的吃水

決策變數說明

T_j^k ：= $\max\{T_i^k + t_i^k, a_j\}$ 船舶 j 接受船席 k 服務的開始時間點，在同一船席上，船舶 j 須等上一艘船舶結束服務後或船舶 j 本身到達港口後始能接受服務

x_{ij}^k ：= 1，表示船席 k 服務完船舶 i 之後，接著又服務船舶 j
= 0，其他

二、模式建立

$Min \sum_{j \in N} \sum_{k \in K} [T_j^k - a_j + t_j^k \sum_{i \in N} x_{ij}^k]$	3.1
$Min \sum_{i \in N} \sum_{k \in K} c_i^k [\sum_{j \in N} x_{ij}^k]$	3.2
S.T	
$\sum_{j \in N} \sum_{k \in K} x_{ij}^k = 1, \forall i \in N$	3.3
$\sum_{j \in N} x_{ij}^k - \sum_{j \in N} x_{ji}^k = 0, \forall k \in K, \forall i \in N$	3.4
$a_j \leq T_j^k \leq b_j, \forall k \in K, \forall j \in N$	3.5
$s_j \leq T_j^k \leq e_j, \forall k \in K, \forall j \in N$	3.6
$\sum_{i \in B} (WD_k - D_j) x_{ij}^k \geq 0, \forall j \in N, \forall k \in K$	3.7
$\sum_{(i,j) \in A(S)} x_{ij}^k \leq S - 1, \forall k \in K$	3.8
$\sum_{j \in N} x_{oj}^k \leq 1, \forall k \in K$	3.9
$x_{ij}^k \in \{0,1\}, i \in N, j \in N, k \in K$	3.10

三、模式說明

- 3.1：違反需求點時窗限制、需求點作業時間最小化，即求取顧客(船舶)等待時間與作業時間最小化
- 3.2：節線成本最小化，即求取經營者(港埠當局)作業成本最小化
- 3.3：需求點只能被服務一次，即每一艘船舶只能被一個船席服務一次
- 3.4：流量守衡限制式
- 3.5：需求點的時窗限制，即船舶接受服務的時窗限制
- 3.6：虛擬場站的營運時段，即船舶接受服務的時間須在港口的營運時段內
- 3.7：車種與需求點的配合，即每艘船舶的吃水須與船席水深配合
- 3.8：避免次迴路的產生
- 3.9：每部車輛只能服務一條路徑，即每個船席只能有一群船舶服務順序名單
- 3.10：二元限制式

3.5 模式修正與求解

在構建多目標模式後，必須將多目標式轉化為單目標以求解。在求解的模式上過去文獻中多半採用權重法，亦即依據偏好設定各目標間之權重，藉由權重的調整將求解模式轉換為單一目標之線性模式求解。但由於權重之決定涉及主觀意識，且各目標間可能產生因量度不同而產生之偏誤，因此，本研究改以隸屬函數方法，構建各目標之滿意度值，以 Maxmin 之模糊多目標線性規劃程式改寫數學式並進行求解。

在做法上，先行就兩個目標式分別求解以建立報酬表，找出各目標在求解

過程中出現之最佳值 (f_1^+ 、 f_2^+) 及最差值 (f_1^- 、 f_2^-)，並建立隸屬函數。之後在求解時，將各目標之求解結果以隸屬函數表示，令各目標之隸屬函數必須大於或等於某個 λ 值 (滿意度)，即以兩個目標中滿意度較低之 λ 值作為該組解之代表值，最後，找出滿意度 (λ) 最大之一組解作為代表解，即可將多目標問題轉化為單目標進行求解。

多目標式轉換為單目標之規劃式如下：

$$\begin{aligned} & \text{Max } \lambda \\ \text{s.t. } & \frac{f_1 - f_1^-}{f_1^+ - f_1^-} \geq \lambda \\ & \frac{f_2 - f_2^-}{f_2^+ - f_2^-} \geq \lambda \end{aligned}$$

其中， λ 為兩個目標中滿意度代表最低值， f_1 為第一目標式之求解值， f_2 為第二目標式之求解值。

3.6 小結

1. 在本研究中，以車輛繞徑的模式來解釋船席指派問題，目標函數中，探討的是經營者使用成本最小化與顧客總服務時間最小化
2. 本研究車輛繞徑模式中，節線的成本異於一般車輛繞徑模式，節線成本的定義係服務該船舶所使用的成本，且此成本會因為船席與船舶的組合而有所不同。
3. 本研究中目標函數以最小化使用成本與總服務時間(包含等待時間)，並以多目標的方式來呈現，以此種方式是希望能夠反映出現實世界中船席會議的主要參予者及對此兩各角色所產生的影響。



第四章 求解策略與方法

本章將探討如何應用演算法(類免疫演算法與禁忌搜尋法)應用於船席指派問題之求解，包括整個演算法的參數設計與運算流程。本研究會選擇此兩種演算法的原因如下：

1. 基因演算法具有平行搜尋的能力，且近年來基因演算法被廣泛的應用在求解最佳化問題；類免疫演算法屬於近年來較新的演算法，其與基因演算法概念類似。本研究選用類免疫演算的主要原因，一方面為文獻中有將類免疫演算法與基因演算法做過探討比較的研究，其結果顯示，類免疫演算法似乎有比基因演算法更佳的收尋能力，另一方面，目前應用類免疫的研究較少，所以選擇類免疫演算法作為本研究的求解工具。
2. 禁忌演算法藉由 Tabu List 可以避免重複性或毫無目標的搜尋，利用移步方式來進行鄰域搜尋，具有系統性，與類免疫在演算流程上較不同。
3. 類免疫的演算流程與基因演算法類似，且都有利用到機率的概念，且過去文獻已經有對此兩種演算進行比較；相對於基因與類免疫，禁忌法則是用具人工智慧的系統機制。因此，選擇兩種演算機制不同的演算法(禁忌、類免疫)來作為比較，可以作為比較的基礎且能夠顯示出演算法的特性。

4.1 求解策略概念

而一般 VRP 的相關問題，在考量網路特性、顧客需求、設施資源及配送條件等限制，各類型的 VRP 問題整理如 2.5.2 節所述，各類型間的差異不大，往往只改變一個或兩個限制式而已，其目標在追求一般化成本總和最小之目標下，決定出最佳的車隊組成與配送路線方案。

而在此種問題的數學規劃模式上，其目標式通常為一般化最小成本，而此成本大約可以分為：

1. 固定成本：購買成本、折舊、司機薪資。
2. 變動成本：與路線有關，例如行駛距離、時間、油耗、輪胎磨耗、裝卸貨時間。
3. 懲罰成本：例如違反顧客時窗限制所產生的成本。

在顧客需求限制方面，包含流量守衡、避免子迴圈、顧客需求量的固定與隨機、不可分割、時間窗。

設施資源的限制方面，包含場站(單一或多場站)、車隊(單一車種或多車種、車輛數限制)及人員(通常假設人車合一，並將人員的薪資反應在車輛的使用成本上)。

船席指派問題與車輛繞徑問題的轉換如第三章所述，若以車輛繞徑問題來看待船席指派；目標式方面，以多目標的概念來看待，因為每個需求點(船)對於不同的車輛(船席)所產生的成本都不盡相同，這是因為每個船席上的建造成本、機具、人員都不相同，另外，對於顧客的所產生服務時間與等待時間也不相同，因此，目標式在追求經營者的成本最小化與顧客的服務時間最小化。

在限制方面，本研究的限制式與一般車輛繞徑相同的有，需求點只能被服務一次、流量守衡、需求點的時窗限制、避免次迴路，而因應船席指派問題本身的則有場站的營運時間、需求點的時窗限制、船席與船舶之水深與吃水之限制。

因此，在問題的求解上，主要考量限制在於車輛與需求點間屬性的配合(水深與吃水)及不違反最大營運時間。因此，只要是分群的每一部車輛不要違反營運時間、需求點與車輛數性的差異，都是一個可行解，至於違反需求點時窗限制部分，則會被反應於目標式中。

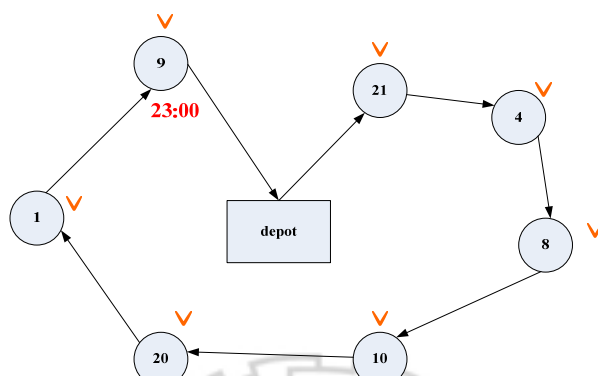


圖 4-1 求解策略示意圖(一)

如圖 4-1 所示，依本研究目標與限制的定義，需求點有時窗限制，但是實務上，會有發生船舶等待的情況發生，即會有違反時窗限制的情況，而此種情形會被反映在目標式裡面。

而我們所要考量的因為，需求點與車輛屬性的配合(水深與吃水)與不違反最大營運時間，找出最佳路線。

在車輛數方面，車輛繞徑問題，可以決定出最適車隊的決策，以多次的演算並進行統計計算，找出較有可能的車隊規模，通常會假設車輛無限，或者當超過其既有車輛數時，採取的方案為租用車輛等；但是對於港口而言，其船席數通常已固定，短時間內要增加不太可能，因此，針對問題的特性，車輛數在本研究中(船席)假設為固定。圖 4-2 顯示出當違反車輛屬性與需求點的情況及違反最大營運時間的情況。

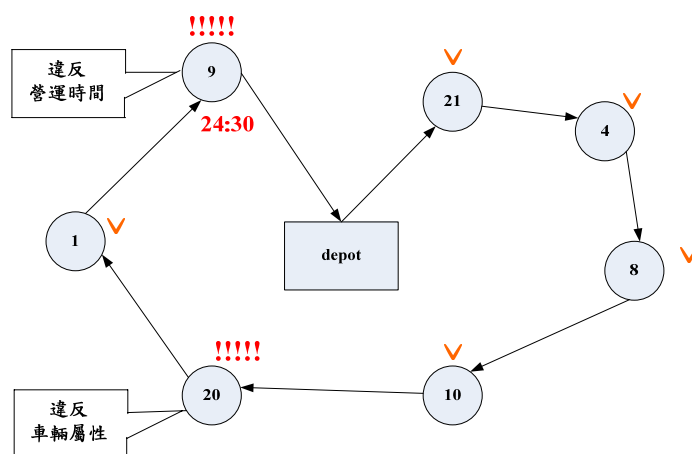


圖 4-2 求解策略示意圖(二)

4.2 問題輸入與輸出

輸入項：

1. 船席的數量、水深、使用成本
2. 船舶的數量、吃水、時窗限制、裝卸貨時間
3. 港口營運時段

輸出項：

1. 指派順序

4.3 資料結構

關於其解過程中對於資料的處理，本研究除了利用陣列(Array)外，還建立以下四種結構體，利用結構體的優點，可以快速的擷取資料，此外，結構體除了有自由建立型別的特性外，還擁有類似一般變數設定初值的特性，將一個結構體直接設值給另一個結構體，此方面比運用陣列在處理資料的時候快速許多。此外，結構體陣列的特性，對於本研究演算法類免疫演算法、禁忌搜尋法產生一群解的儲存方式上，也有相當大的便利性。



```
1. typedef struct ship{
        } Ship
```

船舶編號	時窗限制	服務時間點	作業時間	離開時間	吃水深度	在港時間
------	------	-------	------	------	------	------

圖 4-3 船舶屬性結構體

此一結構體儲存船舶的相關屬性資料，結構體的相關成員值在指派順序與車輛(船席)決定後，在程式中會進行比較與設定，結構體相關成員如圖 4-3 所示。

```
2. typedef struct num_ans{
        int order[berth_n][max_order]
    }Num_ans
```

1	7	5	8	0	0
10	2	8	12	0	0
11	4	3	0	0	0
6	13	14	0	0	0

船舶編號

船舶順序

圖 4-4 船舶編號二維陣列結構體

如圖 4-4 所示，列數為船席的數量，而行數則為一個船席最大的服務量，當陣列值為 0 時，表示沒有船舶，以此方式可以記錄船席服務船舶的順序。

```
3. typedef struct ship_ans{
        Ship order[berth_n][max_order];
    }Ship_ans
```

把 Ship 結構體所構成的結構體陣列當成一個新的結構體，此為結構體之結構體的概念，可以在程式中，以一為陣列的方式，把所有的解都存在此一維結構體所構成的陣列。

```
4. typedef struct one_goal{
        } One_goal
```

儲存目標值的結構體，有利於進行多目標規劃時的運算。

在主程式中，則有：

1. Ship ship_data[max_order]

船舶原始資料，吃水深度、編號、時窗限制等屬性，以讀入外部檔案的方式，存在此陣列。

2. float berth_data[berth_n]

船席原始水深資料，以讀入外部檔案的方式，存在此陣列。

3. float time_data[berth_n][max_order]

船舶於每個船席間的作業時間，以讀入外部檔案的方式，存在此二維陣列。

4. float cost_data[berth_n][max_order]

船席在不同船舶上的作業成本，以讀入外部檔案的方式，存在此二維陣列。

5. Num_ans ini_num_group[group_size]

此一陣列紀錄用來記錄群體解，且會隨著程式的流程而更新。

6. Ship_ans ini_ship_group[group_size]

此一陣列紀錄用來記錄群體解(所有鄰近解)，且會隨著程式的流程而更新。且可以作為下面計算目標時基礎。

7. One_goal goal_group[group_size]

此一陣列用來記錄群體解(所有鄰近解)的目標值，且會隨程式流程而更新，在多目標運算時有較大的便利性。

8. Num_ans opti_order

紀錄最佳的指派順序，作為輸出項目。

4.4 求解方法(一)類免疫演算法

本研究主要是將以 VRP 的觀點來探討船席指派問題，輸出、輸入項，目標式、限制式的說明如 4.1 節所述，在此節將會說明應用類免疫演算法於本研究主題上的解題流程，並解釋相關細節。

4.4.1 類免疫演算法應用步驟與流程

類免疫演算法的基本原則與程式步驟如下所述，如圖 4-5 所示：

1. 第一道天然免疫反應：辨識抗原，隨機產生初始抗體族群，且此抗體族群須符合港口的營運時間限制，此即免疫系統的第一道防線，並將世代數設定為零， $t = 0$ 。
2. 計算抗體的親和度與細胞記憶：將抗體所代表的變數字串解碼成數值，並將這些數值帶入目標函數而得到各抗體的親和度，並將最佳的抗體選出存入記憶細胞。
3. 株落選擇：依據所設定的抗體數目，配合蒙地卡羅的概念，選擇親和度較佳的抗體複製到記憶細胞區 A，並將記憶細胞區 A 進行細胞分裂，並產生兩個和記憶細胞 A 一模一樣的抗體族群，以進行下一步驟之適應性免疫反應。
4. 第二道適應性免疫反應：此部份依「變化的規模大小」分成輕鏈及重鏈的重組兩種形式進行：
 - (1) 輕鏈的重組：第一部份進行抗體受體多樣性演變，此部份將抗體進行大範圍的改變，方式為抗體與抗體之間的交互作用(如基因轉置或取代)，最後依親和性的比率、設定的抗體數目，選出較佳的抗體，儲存於記憶細胞區 B，此一方式與基因演算法的交配類似，原則在於必須比原先的解產生較大的變異。
 - (2) 重鏈的重組：第二部份進行抗體本身的基因重組，此部份將抗體進行小範圍的改變，例如突變等，原則為不要大幅更動抗體本身，最後依親和性的比率、設定的抗體數目，選出較佳的抗體，儲存於記憶細胞區 C，此一方式與基因演算法的突變概念相仿。
5. 形成新抗體族群：將記憶細胞區 A、記憶細胞區 B、記憶細胞區 C 的抗體形成一個新族群。
6. 重複演算：完成上述步驟後，跳回步驟 2，繼續執行下一動作，如此一直反覆進行到所設定的世代數 T。
7. 輸出親和力最佳抗體的指派順序：當演化達所設定的迭代次數後，將最後一個世代的最佳抗體、最差抗體的親和力(目標值)輸出。

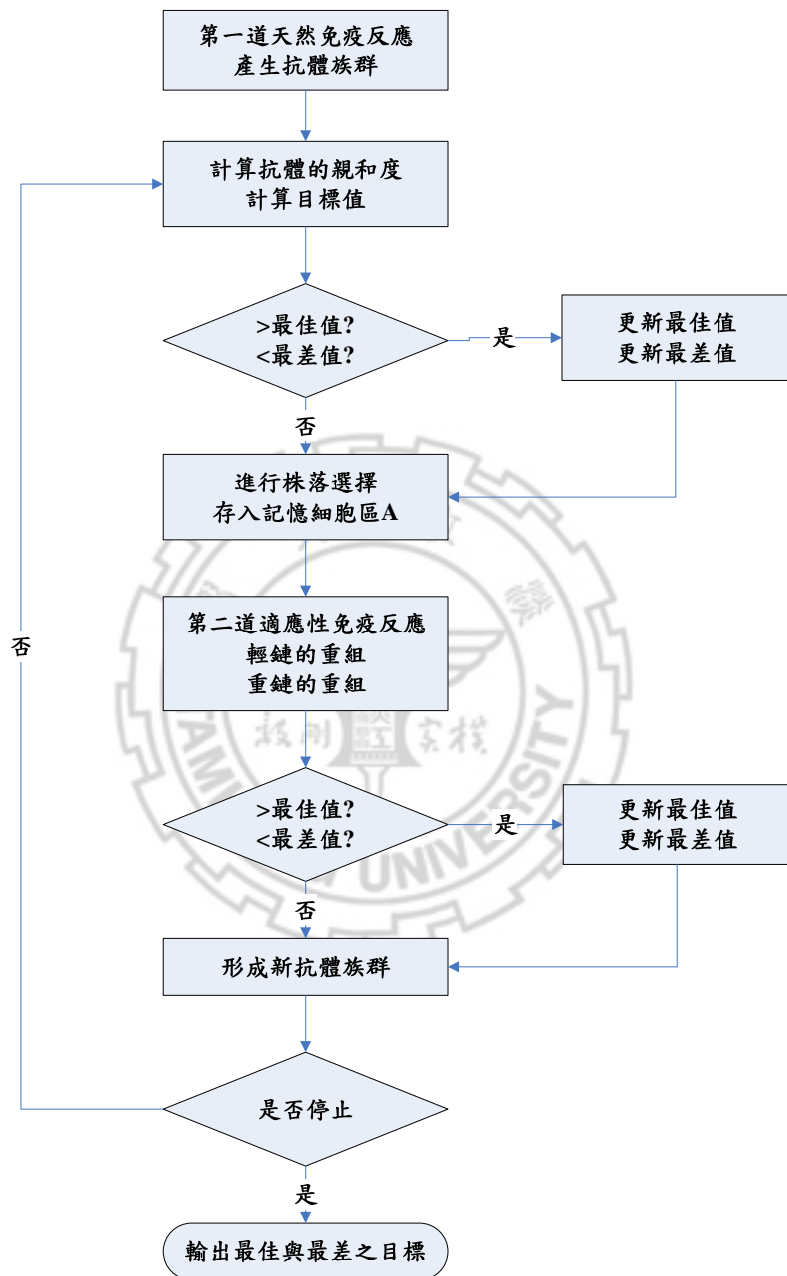


圖 4-5 本研究運用類免疫演算法流程圖

4.4.2 類免疫演算法於船席指派問題求解

在問題及目標的求解上，本研究將兩個目標函數，即在港總時間與使用船席的成本同時放入親和度評估中，藉由每個船席服務的船舶順序來計算各目標值，因為各個目標值皆為求最小化，經由模糊轉換後求出同時滿足兩個目標值之最大滿意度 λ ，親和度的評估方式則以最大的 λ 作為判斷的依據，最後輸出最佳抗體內各個船席服務船舶的順序，得出最佳指派順序。問題的求解概述如下，流程如圖 4-9 所示。

1. 第一道天然免疫反應：辨識抗原，產生抗體，以隨機的方式產生合乎港口營運時段限制之各船席服務船舶順序，在此方面是以隨機的方式一次產生一組解，例如，產生 1 到 20 的船舶編號不重複的順序，且在此一階段，必須注意的是，不能超過港口的營運時間限制。
2. 計算抗體的親和度與細胞記憶：計算上一步驟抗體族群的親和度，其主要目的是要作為步驟 3 的依據。配合多目標計算方法之一致性，這邊所指的親和度是以兩個目標值中(在港時間與使用成本)經模糊轉換後最低的 λ 值為代表。
3. 株落選擇：依據所設定的抗體數目，選取數個 λ 最佳的抗體，並以蒙地卡羅輪盤選擇法，選擇 λ 較大的抗體複製到記憶細胞區 A，並將記憶細胞區 A 進行細胞分裂，並產生兩個和記憶細胞 A 一模一樣的抗體族群，以進行下一步驟 4 之適應性免疫反應。

此部份在程式的撰寫上，各個抗體的 λ 值經計算後，存於一為陣列 goal_group[] 中，經排序之後，先選出前幾個最佳的抗體，之後再以機率的方式選出其他抗體，如圖 4-6 所示。

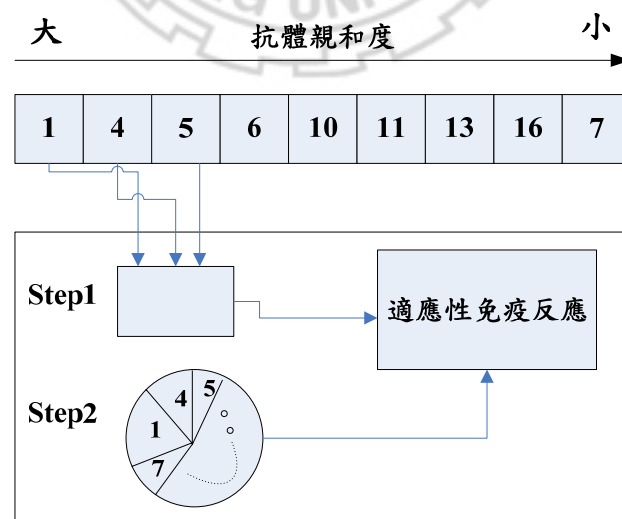


圖 4-6 株落選擇說明圖

4. 第二道適應性免疫反應：此部份依「變化的規模大小」分成輕鏈及重鏈的重組兩種形式進行：
 - (1)輕鏈的重組：第一部份進行抗體受體多樣性演變，此部份將抗體進行大範圍的改變，如圖 4-7 所示；最後依親和性的比率、設定的抗體數目，選出較佳的抗體，儲存於記憶細胞區 B，
 - (2)重鏈的重組：第二部份進行抗體本身的基因重組，此部份將抗體進行小範圍的改變，如圖 4-8 所示；最後依親和性的比率、設定的抗體數目，選出較佳的抗體，儲存於記憶細胞區 C。
5. 更新最佳抗體，在步驟 2 與步驟 4 之後，會紀錄最佳的抗體，此一部份在程式中，會判斷每個抗體的 λ 是否有大於目前的最佳解，如果有的話，則將程式中儲存 λ 與指派順序的變數進行更新。
6. 形成新抗體族群：將記憶細胞區 A、記憶細胞區 B、記憶細胞區 C 的抗體形成一個新族群。
7. 重複演算：完成上述步驟後，跳回步驟 2，繼續執行下一動作，如此一直反覆進行到所設定的世代數 T。
8. 輸出親和力最佳抗體的指派順序：當演化達所設定的迭代次數後，將 λ 值最大的抗體解的結果輸出，作為本研究多目標下的最佳指派順序。

1	3	12	8	9	0	0	1	3	36	12	9	0	0
2	5	36	16	0	0	0	2	5	25	8	0	0	0
11	19	25	7	10	0	0	11	19	7	16	10	0	0

圖 4-7 輕鏈重組示意圖

1	3	12	8	9	0	0	1	3	25	8	9	0	0
2	5	36	16	0	0	0	2	5	36	16	0	0	0
11	19	25	7	10	0	0	11	19	12	7	10	0	0

圖 4-8 重鏈重組示意圖

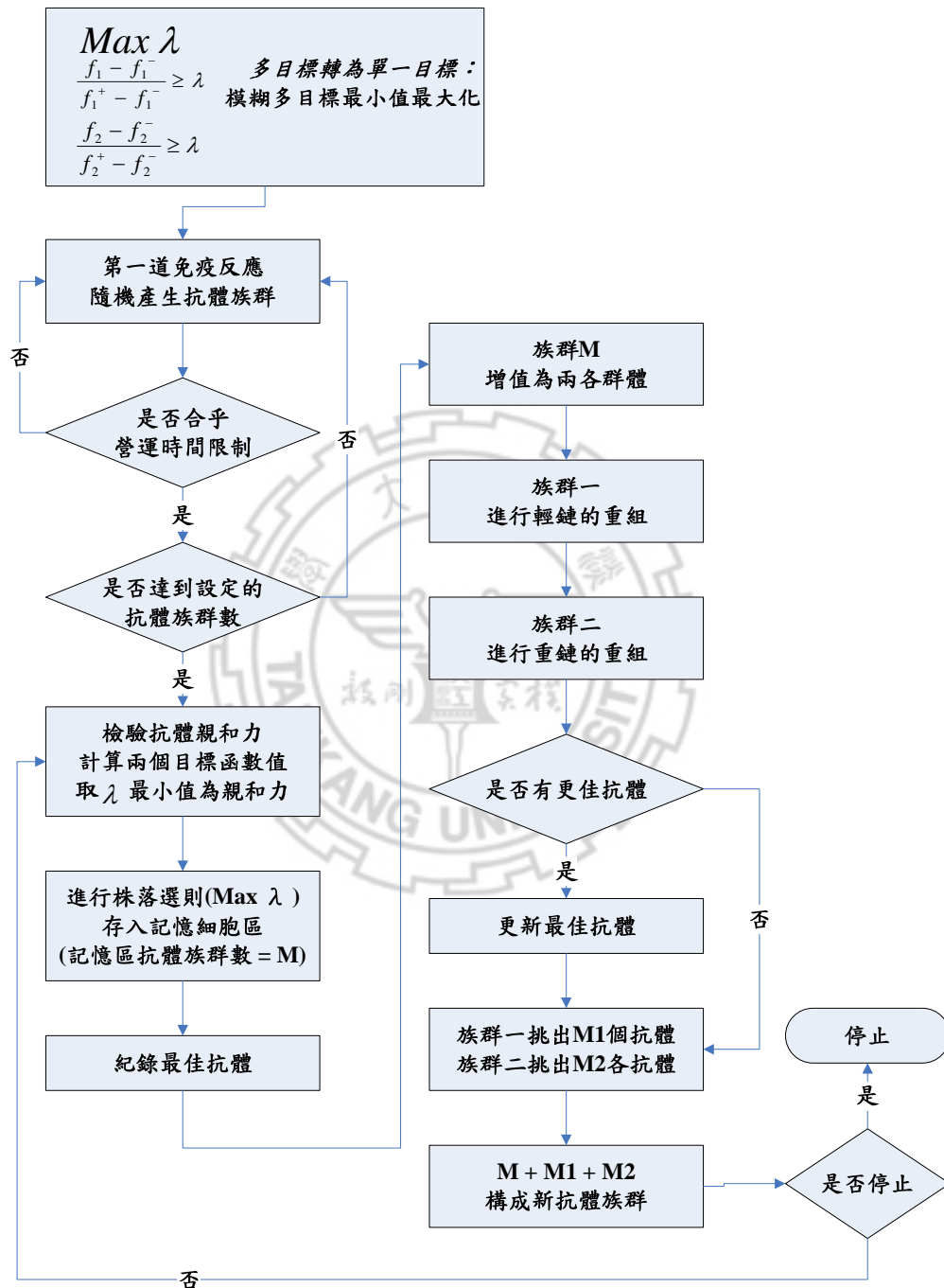


圖 4-9 船席指派問題求解流程圖

4.5 求解方法(二)禁忌搜尋法

4.5.1 禁忌搜尋法應用步驟與流程：

禁忌搜尋法的基本原則與程式步驟說明於下，如圖 4-10 所示。

1. 設定起始狀態：可以由先前的研究或是其他方法找到一個起始解 x_0 ，並另 x_0 為目前最佳解 x^* ，目標值為 $f(x^*)$ 。設定起始迭代次數 $k = 0$ ，禁忌名單 $T = \phi$
2. 移步：假如 x^* 的鄰近解集合 $V(x^*)$ 扣除 T 集合後為空集合(即 $V(x^*) - T = \phi$)，則進行步驟(4)。否則令 $k = k + 1$ ，並從 $V(x^*)$ 中選取能使目標值最低的鄰近解 x_k 。
3. 更新：若 $f(x_k) < f(x^*)$ ，則另此解為目前最佳解，即 $x^* = x_k$ ，並且將此解置入禁忌名單 T 集合中，如果 $|T|$ (可行解個數) $>$ 設定的禁忌名單長度，則剔除 T 集合中最舊的解，若是 $f(x_k) > f(x^*)$ ，則不變動最佳解，並將此解也置入集合 T 中。
4. 測試停止條件：測試停止條件是否成立，如果是，則停止，如果為否，則進行步驟 2。



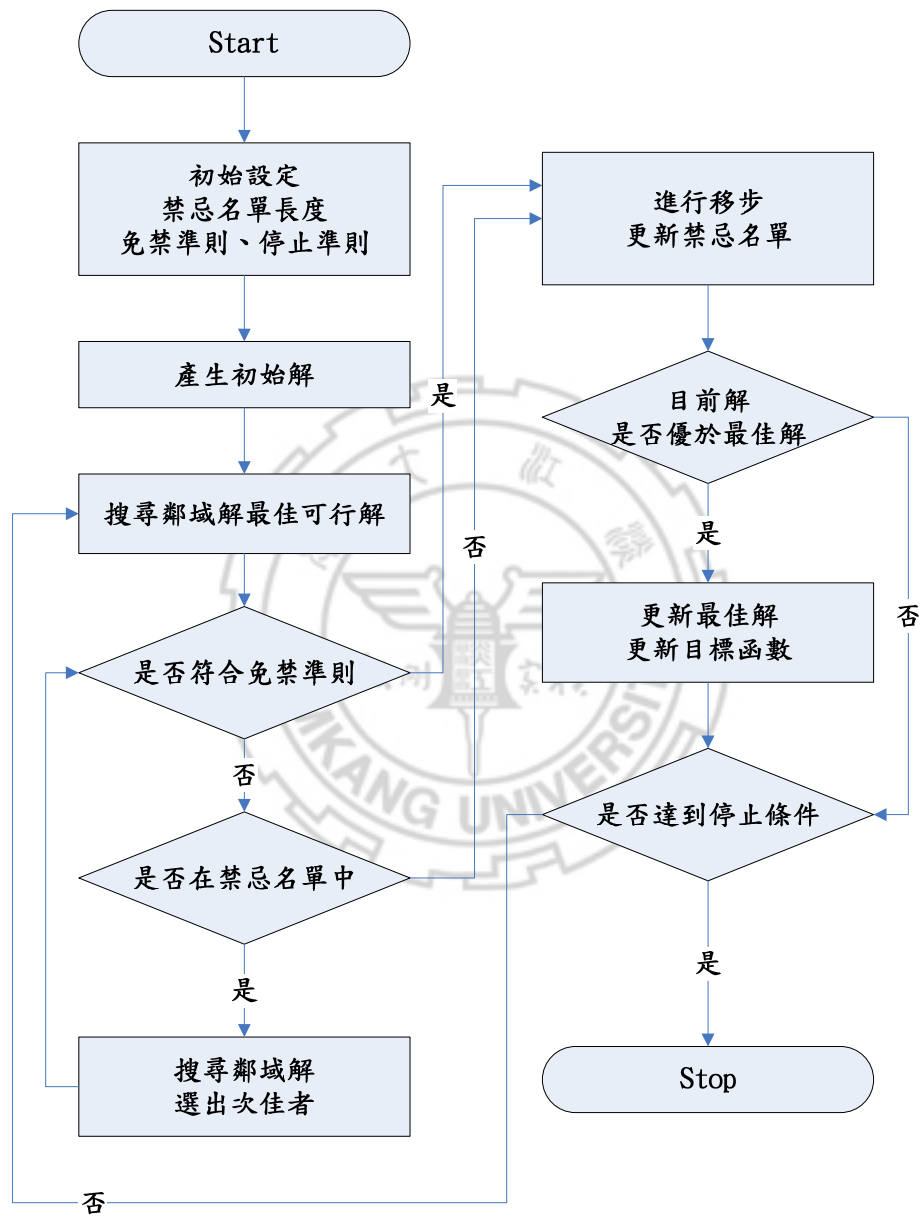


圖 4-10 本研究應用禁忌搜尋法流程圖

4.5.2 禁忌搜尋法於船席指派問題求解

在問題及目標的求解上，本研究將兩個目標函數，即在港總時間與使用船席的成本同時放入鄰近解的集合內，藉由每個船席服務的船舶順序來計算各目標值，因為各個目標值皆為求最小化，經由模糊轉換後求出同時滿足兩個目標值之最大滿意度 λ ，在目標值的評估方式則以最大的 λ 作為判斷的依據，得出最佳指派順序。

本研究將船席指派問題構建為車輛繞徑模式，因此在求解方法上，會參考車輛繞徑問題的相關文獻與解法，以尋求較合適的求解方法。

1. 設定起始狀態：

一般最常用於產生起始解的方式是隨機的方式產生，但若以隨機的方式產生一新的解，將會造成後續搜尋的時間加長，並影響最後的求解品質。由於起始解對於搜尋的時間與品質均會有影響，故大部分的研究多採用一些較為科學化的方法，以產生一組品質較好的可行解，以加速搜尋的速度。常用的方法包含最鄰近法(Nearest Neighbor Method)、節省法(Saving Method)、掃描法(Sweep Method)等。依據敖君瑋(1999)的測試結果，可發現最鄰近法明顯優於其他方法。故本研究將使用最鄰近法作為求取演算流程中的起始解。

所謂最鄰近法是由場站出發，選取與場站間距離最短的節點，並將此節點納入途中。接者再以所選取的節點為起點，在不違反相關限制條件下，尋找下一最近的節點做移動。若車輛無法滿足節點的需求量時，則指派另一車輛作服務，直到拜訪完所有的節點。

本研究的鄰近法的演算方式說明如下，如圖 4-11 所示：

- (1) 由到達港口預定時間最早(時窗限制最早)的船舶(需求點)開始選取，將此一船舶納入船席的服務名單內，若需求點時窗限制相同時，則選取節線成本較小的需求點(船席使用成本較小者)。
- (2) 以此艘船舶的完工時間為起點，再選取與此一艘船舶完工時間最接近的另一艘船舶，依此方式將船舶納入船席的服務名單內，當未選入船席服務名單的船舶均違反時窗限制(港口營運時段)時，形成一個船席的服務船舶順序。依此方式，直到所有的船舶都有納入到船席的服務名單中。

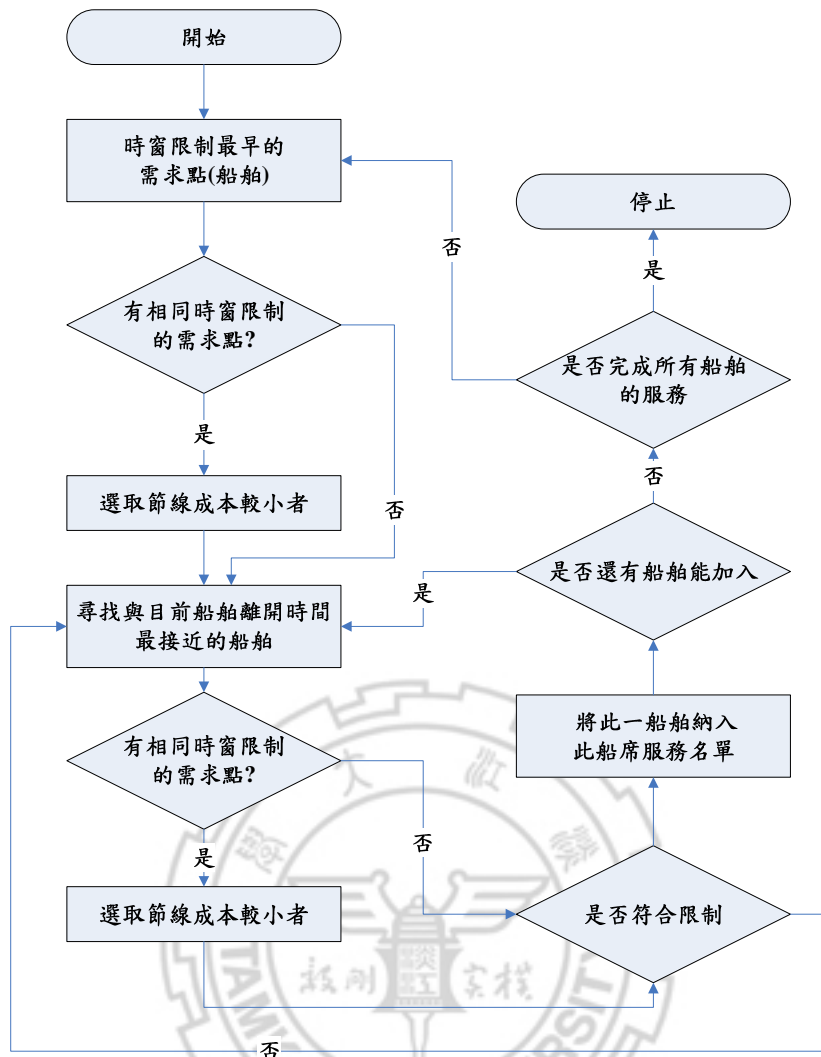


圖 4-11 最鄰近法流程圖

在程式撰寫的過程中，此一啟始狀態會被存在本研究的建立的結構體 Num_ans 當中。此一步驟與類免疫演算法原則一樣，不超過港口的營運時間限制。

2. 鄰近解：

鄰近解即為在目前解的路徑中，透過途程間或途程內，節線或節點以特定的交換方式，在不違反問題的限制條件下，搜尋所有的可行解。一般常用的交換方式包含 Swap Exchange、Or-Opt Exchange、Cross Exchange 等方式。

依據敖君瑋(1999)的分析，在途程內與途程間運用不同的交換方式來做比較可以發現，不論途程間或途程內均利用 Swap Exchange 來做交換，其改善的績效較其他方法來的優，故本研究將針對途程間的改善將利用 2-Swap Exchange 的方式，將兩條不同路線中的某一點做交換，即將不同船席中任兩個船舶編號進行交換。

如圖 4-12 所示。以此一方式作交換，較其他方式更能符合禁忌搜尋法中對於鄰近解的定義。

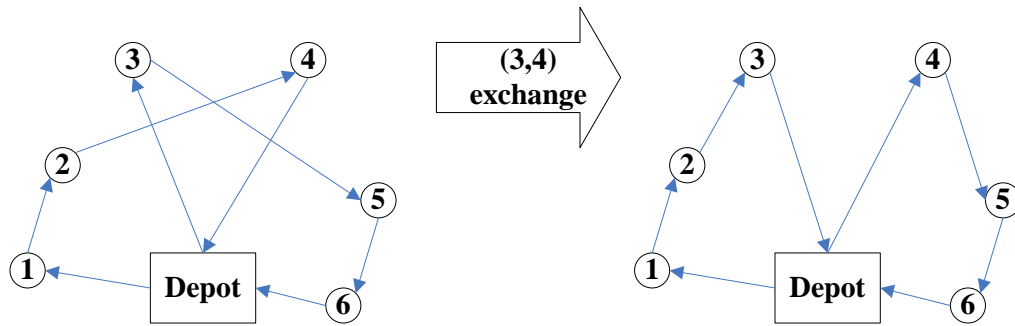


圖 4-12 節點(船舶編號)交換示意圖

在程式中，所有的鄰近解會被存在以一個 Num_ans 所構成的一維陣列當中，後續挑選鄰近解中的最佳解步驟中，會以不超過港口營運時段及不違反水深之條件為原則。

3. 更新禁忌名單：

在程式中，會以一個 int tabu_list[2][size]陣列儲存禁忌名單，先將每個元素起始值設為 0，在以 IF 來判斷某個鄰近解的交換節點是否存於此二維陣列。size 表示禁忌名單長度，當執行更新禁忌名單時，會以先進先出的原則來更新，虛擬碼表示如下，如圖 4-13 所示：

```

For( i = size-2 ; i >= 0; i-- )
{
    tabu_list[0][i+1] = tabu_list[0][i];
    tabu_list[1][i+1] = tabu_list[0][i];
}
tabu_list[0][0] = exchang_a;
tabu_list[1][0] = exchang_b;

```

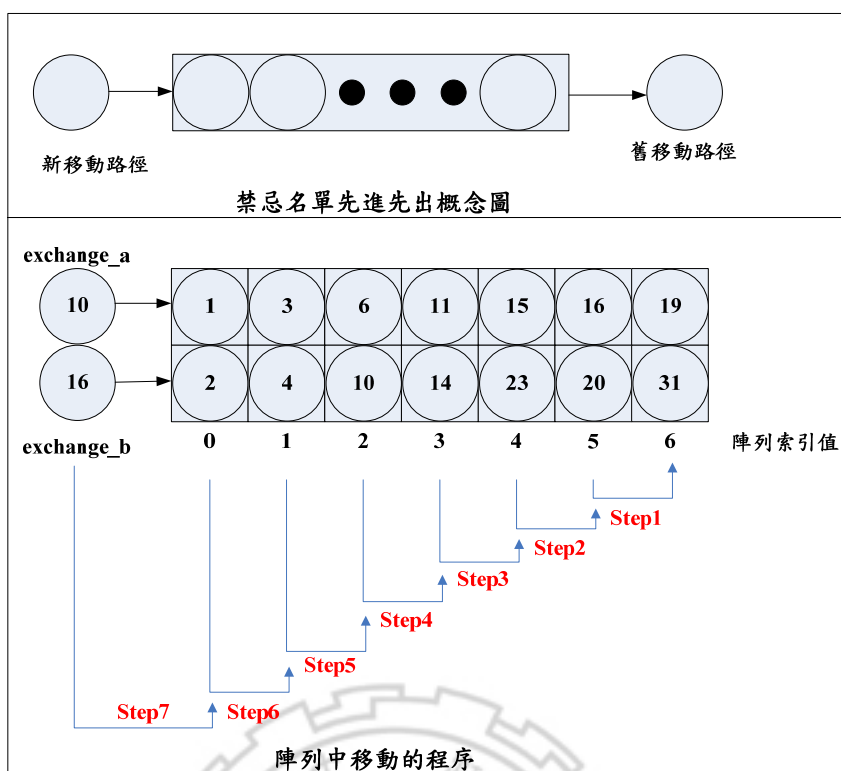


圖 4-13 更新禁忌名單示意圖

4. 免禁準則：

在免禁準則上，考慮一個移步 m ，其移動能達到的解答，會比目前所搜尋到的最佳解還要好，則應該接受此移步。本研究對於免禁的判斷，將以目前最佳的解作為依據，若其目標值優於目前最佳解的值，則符合免禁準則。

5. 測試停止條件：

測試停止條件是否成立，如果是，則停止，如果為否，則進行步驟(2)

針對停止的準則，本研究將採用目標函數持續未改善的迭代次數為依據，雖然廖麗滿(2001)的研究中指出當連續沒有改善的次數越多，其解的品質將越好，但同時也會增加執行的時間，故本研究將依題目不同設定一合理的迭代次數。

4.6 小結

此章主要說明本研究主題所採用的演算法，選擇此兩種演算法的原因是希望藉由兩種不同的機制的演算方式來進行後續演算法的比較，較能突顯出演算法的特性；並說明兩種演算法的求解流程、程式資料結構相關應用，以作為後續解題的基礎。

在多目標的求解上，將會先以個別目標去得出模糊隸屬函數的上下界，接著再以最小值最大化法求解；至於求解流程、演算法相關參數(例如重組率、抗體數目)將會於第五章有較詳細的說明與解釋。

第五章 範例測試與參數設計比較

針對研究主題，以不同船席數、船舶數建立小型範例進行測試，主要目的係為了分析驗證模式的正確性，求解則以窮舉的方式並配合 C 語言撰寫程式而成，窮舉法中主要是利用其函式庫 `next_permutation()` 找出所有排列組合功能，並配合本研究的相關限制找出最佳解。

接著將會以第四章類免疫演算法與禁忌搜尋法所撰寫之電腦程式，進行上述相同小型測試範例求解，測試演算法正確度，並以不同的參數設計，利如禁忌名單長度、輕鏈重組率等進行演算法特性之實驗設計，找出較佳的參數、比較兩種演算法的求解品質。

找出較佳的參數後，配合本研究所定義的輸出入項目，說明多目標船席指派求解流程；此外，以基隆港過去的船席指派紀錄作為實證資料，並與本研究之多目標的求解結果進行比較。

5.1 範例測試

5.1.1 窮舉法

1	7	5	8	0	0
10	2	8	12	0	0
11	4	3	0	0	0
6	13	14	0	0	0

船舶編號

船舶順序

圖 5-1 船舶編號二維陣列結構體

有關本研究應用於大型例題的啟發式解法已在第四章已經有詳細的說明，而此小節將會說明小型範例模式的驗證方式，首先在編碼方式方面，如圖 5-1 所示，列數為船席的數量，而行數則為一個船席最大的服務量，當陣列值為 0 時，表示沒有船舶，以此方式可以記錄船席服務船舶的順序。如圖 5-1 所示，表示有四個船席，14 艘船舶，以第一個船席(第一列)為例，其服務船舶的順序為 1→7→5→8，後面兩個 0 表示沒有服務船舶，第二列、第三列、第四列以此類推，程式撰寫之資料結構如下所示。

```
typedef struct num_ans{
    int order[berth_n][max_order]
}Num_ans ;
```

窮舉法求解步驟，主要是以一個迴圈將所有的排列組合列出，而每產生一個排列組合時，進行限制式檢查、目標式求解，若是有產生更佳的解，則進行最佳解的更新，以此方式找出小型範例的最佳解，求解的過程如下

1. 以窮舉的方式找出個別目標下的最高及最低解
2. 將兩個目標之最高解與最低解紀錄後，作為兩個目標均衡解之上下界。
3. 再以窮舉的方式分別求兩個目標的隸屬值，並以兩個目標中隸屬值低者作為多目標模式下均衡解 λ 值。
4. 第二次窮舉法中，則以所有的可行解中 λ 最大者為最佳解；窮舉法程式虛擬碼如下：

```
#include "my_head_file.h"
int main(int argc, char *argv[])
{
    int count = 0 ;
    int num[N]={0, 0, 0, 0, 0, 1, 2, 3, 4, 5} ;
    long int begin,end;
    begin = clock ( );
    do{
        array_trans_num_ans(num, temp_num_ans) ;
        *temp_time = get_total_in_port(temp_num_ans) ;
        *temp_cost = get_cost_in_port(temp_num_ans) ;
        judge_renew_multi_goal(temp_time,temp_cost,);
    }
    while( next_permutation(num, num + N) ) ;
    end = clock( ) ;
    printf("\n run_time = %g secoud ",(float)(end - begin)/CLK_TCK) ;
}
```

5.1.2 小型範例測試求解結果

船席指派問題四個小型範例於附錄一，測試範例的資料包含：

1. 需求點(船舶)的相關資訊，包含船舶數量、船舶的編號、到達時間 a_i 、船舶吃水 D_i 、作業時間 t_i^k
2. 車輛(船席)資訊，包含船席數量、船席水深 WD_k
3. 節線資訊，即船席 k 服務船舶 i 的作業成本 c_{ij}^k 。

窮舉法求解範例一、範例二、範例三與範例四的結果如表 5-1、5-2、5-3、5-4、所示，表格內容包含個別目標下的最高值與最低值、多目標求解下的最大隸屬函數、指派順序、船舶接受服務的時間點、窮舉法求解時間。

表 5-1 範例一窮舉求解結果

問題規模(2 , 4)				
單一目標求解	在港總時間	最高值 f_1^+	最低值 f_1^-	
		149	21	
	使用成本	最高值 f_2^+	最低值 f_2^-	
		205	160	
多目標求解	最大隸屬函數	0.60		
	指派結果			
berth_1	船舶編號	2		
	開始接受服務時間	14		
berht_2	船舶編號	4	1	3
	開始接受服務時間	33	39	52
求解時間(秒)	1.2			

表 5-2 範例二窮舉求解結果

問題規模(2 , 8)								
單一目標求解	在港總時間	最高值 f_1^+			最低值 f_1^-			
		1037			180			
	使用成本	最高值 f_2^+			最低值 f_2^-			
		201			131			
多目標求解	最大隸屬函數	0.67						
	指派結果							
berth_1	船舶編號	2	5					
	開始接受服務時間	37	51					
berht_2	船舶編號	7	8	6	3	1	4	
	開始接受服務時間	45	70	109	129	147	162	
求解時間(分)	72							

表 5-3 範例三窮舉求解結果

問題規模(3 , 5)						
單一目標求解	在港總時間	最高值 f_1^+			最低值 f_1^-	
		796			139	
	使用成本	最高值 f_2^+			最低值 f_2^-	
		381			145	
多目標求解	最大隸屬函數	0.41				
	指派結果					
berth_1	船舶編號	2				
	開始接受服務時間	32				
berht_2	船舶編號	3	5	1		
	開始接受服務時間	73	98	148		
berht_3	船舶編號	4				
	開始接受服務時間	66				
求解時間(秒)	3.52					

表 5-4 範例四窮舉求解結果

問題規模(3 , 7)							
單一目標求解	在港總時間	最高值 f_1^+		最低值 f_1^-			
		18501		3111			
	使用成本	最高值 f_2^+		最低值 f_2^-			
		1424		477			
多目標求解	最大隸屬函數	0.53					
	指派結果						
berth_1	船舶編號	2					
	開始接受服務時間	98					
berht_2	船舶編號	7					
	開始接受服務時間	95					
berht_3	船舶編號	4	6	3	5	1	
	開始接受服務時間	227	981	1762	2094	2507	
求解時間(分)	117						

5.1.3 小結

本研究從航商與港埠管理當局的角度來構建多目標最佳化模式，考量的項目為航商的總在港時間與港埠當局的營運成本，並藉由模糊多目標的求解方式來求得最佳解。

經小型範例的測試後，證實了模式在小尺度問題之正確性(有解)，範例中的相關參數，如成本、時間，採用張企宏(2004)所著「貨櫃碼頭分派問題之研究」內的部份船舶到達時間資料，其到達時間單位為「10 分鐘」，而成本項目則為本研究自行假設。

由範例一到範例四之測試結果可以得知，當問題規模的大小隨著船席數及船舶數的增加時，模式的決策變數與限制式將隨之增加，利用窮舉方式的運算時間也隨之增加(範例一求解時間 1.2 秒，範例二求解時間 72 分，範例三求解時間 3.52 秒，範例四求解時間為 117 分)，因此，當問題規模變大時，有必要以一個較有效率的啟發式演算法來尋求一可行解。

5.2 啟發式演算法

針對船席指派問題在上一小節以窮舉法的方式可以看出，求解時間會隨著規模的增加而遞增，因此大規模的船席指派問題不易在短時間內求得正確解，屬於 NP-hard 之問題，因此，本研究將會以啟發式演算法，在可容忍的時間內取得一可行解。

上一小節窮舉法，是以 Dev C++ 中 next_permutation() 函式來求得所有的排列組合，並配合本研究主題求出指派結果；本研究撰寫之演算法(類免疫演算法、禁忌搜尋法)正確度測試方式為：兩種演算法分別求解上一小節相同四個小型範例，再與窮舉法所得的結果來進行比較，比較單目標與多目標下的求解結果，求解多目標時，隸屬函數的上下界係採用窮舉法所得出的上下界。

5.2.1 類免疫演算法

類免疫演算法求解附錄一四個小型範例，其求解結果如表 5-5 所示，從表可以得知，以類免疫演算法求解所得的結果，不論是船舶總在港時間或是港埠管理當局的使用成本，在單目標下，均可以求得與窮舉法一樣的最佳解 f_1^- 、 f_2^- ；以類免疫演算法於求解多目標過程中，個別目標所求得的最差值 f_1^+ 、 f_2^+ ，如表內 999、786、195，很接近窮舉法的最差值，範例四 14333 則相差較多。

表 5-5 類免疫演算法求解小型範例結果

f_1^+ ：在港總時間最高值， f_1^- ：在港總時間最低值 f_2^+ ：使用成本最高值， f_2^- ：使用成本最低值										
單一 目標 求解	在港 總時 間		範例一		範例二		範例三		範例四	
			f_1^+	f_1^-	f_1^+	f_1^-	f_1^+	f_1^-	f_1^+	f_1^-
		窮舉法	149	21	1037	180	796	139	18501	3111
		類免疫	149	21	999	180	786	139	14333	3111
	使用 成本		f_2^+	f_2^-	f_2^+	f_2^-	f_2^+	f_2^-	f_2^+	f_2^-
		窮舉法	205	160	201	131	381	145	1424	477
		類免疫	205	160	195	131	381	145	1424	477
多目 標求 解	最大 隸屬 函數	窮舉法	0.6		0.67		0.41		0.53	
		類免疫	0.6		0.67		0.41		0.53	

5.2.2 禁忌搜尋法

禁忌搜尋法求解附錄一四個小型範例，求解結果如表 5-6 所示，從表可以得知，以禁忌搜尋法求解所得的結果，個別目標下，船舶總在港時間或是港埠單位的使用成本，均可以求得最佳解；禁忌搜尋法於求解多目標過程中，個別目標所求得的最差值 f_1^+ 、 f_2^+ ，如表內 110、680、319、174、331，相較於類免疫演算法，並不是很完全接近窮舉法中的最差值。

表 5-6 禁忌搜尋法求解小型範例結果

f_1^+ ：在港總時間最高值， f_1^- ：在港總時間最低值 f_2^+ ：使用成本最高值， f_2^- ：使用成本最低值										
單一目標求解	在港總時間		範例一		範例二		範例三		範例四	
			f_1^+	f_1^-	f_1^+	f_1^-	f_1^+	f_1^-	f_1^+	f_1^-
		窮舉法	149	21	1037	180	796	139	18501	3111
		禁忌	110	21	680	180	319	139	7517	3111
	使用成本		f_2^+	f_2^-	f_2^+	f_2^-	f_2^+	f_2^-	f_2^+	f_2^-
		窮舉法	205	160	201	131	381	145	1424	477
		禁忌	205	160	174	131	331	145	1237	477
	多目標求解	最大隸屬函數	窮舉法	0.6	0.67	0.41	0.53			
			禁忌	0.6	0.67	0.41	0.53			

5.2.3 小結

禁忌搜尋法、類免疫演算法求解附錄一四個小型範例的結果整理如表 5-7 所示，個別目標之下，禁忌搜尋法與類免疫演算法均可以求得最佳解，證實了兩種演算法的正確性。而個別目標下最差解的結果，類免疫演算法可以求得到較接近窮舉法下所得的最差解 f_1^+ 、 f_2^+ ，此可說明類免疫演算法的搜尋空間較禁忌演算法大；多目標指派結果方面，除了類免疫演算法求解範例二、禁忌演算法求解範例一(求解指派結果於表 5-7)，其他的求解指派結果均與窮舉法一樣。

「求解品質」方面，禁忌搜尋法與類免疫演算法在四個小型範例中，均可以得到最佳解，求解過程係經由多次的運算所得出；兩種演算法在「求解時間」均有不錯的表現，窮舉法於範例二、範例四的求解時間為 72 分與 117 分鐘，求解時間相當冗長，而禁忌搜尋法與類免疫演算法在四個範例的求解速度均能於 5 分鐘內求得最佳解。



表 5-7 禁忌搜尋法、類免疫演算法求解小型範例結果比較

單一目標求解	在港總時間		範例一		範例二		範例三		範例四		
			f_1^+	f_1^-	f_1^+	f_1^-	f_1^+	f_1^-	f_1^+	f_1^-	
		窮舉法	149	21	1037	180	796	139	18501	3111	
		禁忌	110	21	680	180	319	139	7517	3111	
		類免疫	149	21	999	180	786	139	14333	3111	
	使用成本		f_2^+	f_2^-	f_2^+	f_2^-	f_2^+	f_2^-	f_2^+	f_2^-	
		窮舉法	205	160	201	131	381	145	1424	477	
		禁忌	205	160	174	131	331	145	1237	477	
		類免疫	205	160	195	131	381	145	1424	477	
	多目標求解	最大隸屬函數	窮舉法	0.6		0.67		0.41		0.53	
禁忌			0.6		0.67		0.41		0.53		
類免疫			0.6		0.67		0.41		0.53		
演算法相關參數	類免疫	抗體數目大小 = 15 輕鏈重組率 = 0.8，重鏈重組率 = 0.2 演化迭代數 = 600 記憶區 A 抗體數目與記憶區 A 抗體數目比例 = 1：1									
	禁忌	禁忌名單長度 = 7，最大的迭代數 = 500									
類免疫演算法求解範例二結果			7 8 6 2 5 3 4 1								
禁忌演算法求解範例一結果			3 1 2 4								

5.3 參數設計比較

在上一小節藉由窮舉法於四個小型範例的求解結果測試禁忌搜尋法、類免疫演算法的正確性；此一小節先進行演算法參數設計比較，以求出較佳的參數設計，求出較佳的參數設計後，比較兩種演算法的求解品質，此一小節範例相關資訊，詳見附錄二，參數實驗設計目標值則取船舶總在港時間(單一目標)。

5.3.1 參數設計內容與比較方式

類免疫演算法測試的參數種類分為(1) 起始抗體數目(2) 輕鏈與重鏈重組率(3) 運算迭代數(4) 記憶區 A 與 B 取出抗體數目比例；禁忌搜尋法測試的參數種類分為(1) 禁忌名單長度(2) 鄰域搜尋次數。

禁忌搜尋法參數設計參考張世峰(2002)，類免疫演算法則為本研究自行設計，測試的方式為各種參數各別進行測試，如起始抗體數目大小(15、25、40、60)，運算迭代數(300、500、700、900)。參數測試相關內容如表 5-8、5-9 所示。

表 5-8 類免疫演算法參數設計相關內容

參數種類	參數設計一	參數設計二	參數設計三	參數設計四
起始抗體數目大小	15	25	40	60
抗體輕鏈重組率	0.5	0.7	0.8	0.5
抗體重鏈重組率	0.5	0.4	0.2	0.8
運算迭代數	300	500	700	900
記憶區 A 抗體大小： 記憶區 B 抗體大小	1：1	1：3	4：1	

表 5-9 禁忌搜尋法參數設計相關內容

	參數設計一	參數設計二	參數設計三
禁忌名單長度	5	7	14
鄰域搜尋次數	50、100、500、800、1000		

比較較佳參數的方法上，將會利用統計學上檢定多個母體平均數是否相同的方式來求得最佳參數，因此，我們將對演算法在參數實驗上做一些相關假設，以起始抗體數目設計為例，一共有 15、25、40、60 四種設計，假設每個抗體數目實驗下之目標值母體均為常態分配，且變異數具有齊一性(homogeneity)。求取較佳的方法如下：

1. 每個參數設計下分別演算 30 次，並紀錄每次的最佳值
2. 以 ANOVA 檢定每種參數所得到目標值平均數是否有顯著差異，得出最佳參數值
3. 檢定後，若有顯著的差異，則以平均目標值最低的參數設計為最佳參數，若是沒有的話，則挑選平均目標值最低的參數設計為最佳參數。

以抗體數目為例，每次運算的結果如表 5-10 所示，從表可以得知，F 值為 15.60，大於臨界值 2.68，可以得知，求解的品質會因為抗體數目實驗設計而有所差異，因此，較佳的抗體數目設定為 60。

表 5-10 抗體數目實驗設計運算結果

抗體數目 運算次數	15	25	40	60
1	104012.5	103378.9	105660.9	98023.68
2	105499.9	108545.7	102023.1	104002.9
3	103827.5	107708.9	103854.5	99731.52
.....				
30	104626.5	101537.1	105077.9	100075.7

單因數變異數分析				
抗體數目	次數	總和	平均	變異數
15	30	3182097	106069.9	6655362
25	30	3135477	104515.9	6771125
40	30	3104845	103494.8	3888410
60	30	3058586	101952.9	5743773

ANOVA						
變源	SS	自由度	MS	F	P-值	臨界值
組間	2.7E+08	3	89962231	15.6058	1.36E-08	2.682809
組內	6.7E+08	116	5764667			
總和	9.4E+08	119				

5.3.2 演算法較佳參數

以 ANOVA 方式檢定後，類免疫演算法與禁忌搜尋法的較佳參數整理如表 5-11，其中輕鏈重組率、重鏈重組率與鄰域搜尋次數檢定後發現沒有顯著差異，因此以較佳目標值(最小化)為準則，選出較佳的參數，相關的計算與檢定結果詳見附錄三、附錄四。

表 5-11 演算法較佳參數

類免疫演算法(*表示沒有顯著差異)	
參數名稱	參數值
起始抗體數目大小	60
*輕鏈重組率	0.5
*重鏈重組率	0.8
記憶區 A 抗體大小(輕鏈重組)： 記憶區 B 抗體大小(重鏈重組)	1：3
執行迭代數	900
禁忌搜尋法(*表示沒有顯著差異)	
禁忌名單	5
*鄰域搜尋次數	500

5.3.3 小結

經過參數實驗設計，並以統計 ANOVA 檢定後，得出較佳的參數，以作為後續相關分析應用的基礎。

5.4 演算法求解品質比較

此節主要分為兩部份，第一部分會針對禁忌搜尋法在鄰域搜尋演算步驟上進行變化，有分為隨機鄰域搜尋、非隨機鄰域搜尋，在此將兩種演算法稱為 Tabu-1、Tabu-2，因此，一共會有三種演算法進行比較。第二部份將會設計 8 個不同問題規模的範例，以類免疫演算法、Tabu-1 來進行求解，並經統計檢定後，比較兩者在求解品質上是否有顯著的差異。

禁忌搜尋法在鄰域搜尋機制上，所謂隨機鄰域搜尋係採用「機率」的方式，隨機交換兩個節點；非隨機鄰域搜尋的方式，類似「窮舉」概念，例如，當需求點有 20 個，則一個迭代中採用非機率性的搜尋方式會有 C_2^{20} 個解(鄰域搜尋次數)。

5.4.1 比較的方式

(1)演算法種類：

①類免疫演算法(IA)

②禁忌搜尋法(Tabu-1)-隨機鄰域搜尋

```
for( i ; i < 鄰域搜尋次數 ; i++)  
{ a = rand(); b = rand(); swap( &a, &b ) ; }
```

③禁忌搜尋法(Tabu-2)-非隨機鄰域搜尋

```
for( i ; i <  $C_2^{20}$  ; i++)
```

(2)比較的方式：

以統計學上檢定兩母體平均數差檢定的方式，即 T 檢定，來檢定 IA、Tabu-1 的求解品質。檢定的方式為求解附錄二範例 30 次，先以 F 分配檢定兩者的變異數是否相等，在選擇合適的檢定統計量。

①以上一小節求出之較佳參數進行求解 30 次，比較平均目標值

②進行獨立小樣本母體平均數差檢定(兩母體變異數未知→ t 檢定)：

先以 F 分配檢定變異數是否相等，再選擇母體變異數未知但相等或母體變異數未知的檢定統計量進行檢定。

5.4.2 演算法比較的結果

三種演算法求解範例(5,50)結果整理於表 5-12，就演算法求解品質而言，Tabu-2 於兩個目標中所得的解均為最佳，目標一中，以平均最佳解(將 30 次所得出的最佳解平均)而言，IA、Tabu-1 與 Tabu-2 的誤差率分別為 13.47%、10.92%；目標二中 IA、Tabu-1 與 Tabu-2 的誤差率分別為 13.91%與 11.06%。以所求的最佳解而言，目標一中，IA、Tabu-1 與 Tabu-2 的誤差率分別為 3.4%、2.88%；目標二中 IA、Tabu-1 與 Tabu-2 的誤差率分別為 5.24%與 3.75%。計算過程於表 5-12 下方。

在運算時間上，IA、Tabu-1 的運算時間均相當迅速，程式執行中，平均一個迴圈的運算時間大約在 20 秒上下；Tabu-2 運算時間，在 20 個迭內代所花的時間須八個小時多，大約為 IA、Tabu-1 的 48 倍左右 $((8*60*60)/(20*30))$ 。

以 T 檢定來檢定兩種演算法在個別目標時所求的解兩母體平均數差，T 值於表 5-12，從表可以得知，兩者有顯著差異，因此，在此一範例的求解品質上，Tabu-1 優於 IA。

表 5-12 禁忌演算法與類免疫演算法求解結果比較

	類免疫演算法		禁忌搜尋法		
目標	IA		隨機鄰域搜尋 Tabu-1		非隨機鄰域搜尋 Tabu-2
一	最佳解	93931	最佳解	93376.94	90757
	平均最佳解	102983.3	平均最佳解	100669	
	平均運算時間	6.459 秒	平均運算時間	10.05	>8 小時
	T 值 = 3.729(顯著)				
二	最佳解	4677	最佳解	4611.1	4444
	平均最佳解	5062.33	平均最佳解	4935.91	
	平均運算時間	7.22 秒	平均運算時間	15.428	>8 小時
	T 值 = 3.128(顯著)				
	與 Tabu-2 誤差		與 Tabu-2 誤差		Tabu-2
一	平均誤差	13.47%	平均誤差	10.92%	
	個別誤差	3.4%	個別誤差	2.88%	
二	平均誤差	13.91%	平均誤差	11.06%	
	個別誤差	5.24%	個別誤差	3.75%	
誤差求解說明：					
(102983 - 90757) / 90757 * 100% = 13.47 %					
(100669 - 90757) / 90757 *100% = 10.92 %					
(93931 - 90757) / 90757 *100% = 3.4 %					
(93376 - 90757) / 90757 *100% = 2.88 %					

5.4.3 不同範例規模的比較

除了附錄二的範例外，本研究另外自行設計了 8 個不同問題規模的範例，同樣以上述統計檢定的方式，比較 IA、Tabu-1 的求解品質，其結果整理於表 5-13、圖 5-2 所示，灰色區塊表示有顯著的差異，

8 個範例中，以統計 T 檢定兩者的求解品質結果如下：

1. IA 優於 Tabu-1 的範例有 (7, 100)、(7, 120)、(10, 60)、(10, 120)
2. Tabu-1 優於 IA 的範例有 (7, 80)、(10, 100)
3. 兩者沒有顯著差異的有 (7, 60)、(10, 80)，

四個大型範例中 **【(7,100)、(7,120)、(10,100)、(10,120)】**，IA 優於 Tabu-1 的範例有三個(底線部份)，此結果係經由統計的檢定所得之，因此相信大型範例中，IA 有不錯的求解品質。

表 5-13 不同規模範例下演算法求解品質比較

「灰色區塊」表示經統計檢定後，有顯著差異 「*」號表示 IA 優於 Tabu-1 $\alpha = 0.05$ ，雙尾檢定，臨界值 = ± 2.00								
類型	(7,60)		(7,80)		(7,100)		(7,120)	
	IA	Tabu-1	IA	Tabu-1	IA	Tabu-1	IA	Tabu-1
	26106	26288	48923	*48036	*71304	73135	*111728	115197
t	-0.715		2.1		-3.213		4.013	
類型	(10,60)		(10,80)		(10,100)		(10,120)	
	IA	Tabu-1	IA	Tabu-1	IA	Tabu-1	IA	Tabu-1
	*18156	18655	30512	30926	53318	52000	*81476	85222
t	-2.251		-1.408		2.67		7.459	

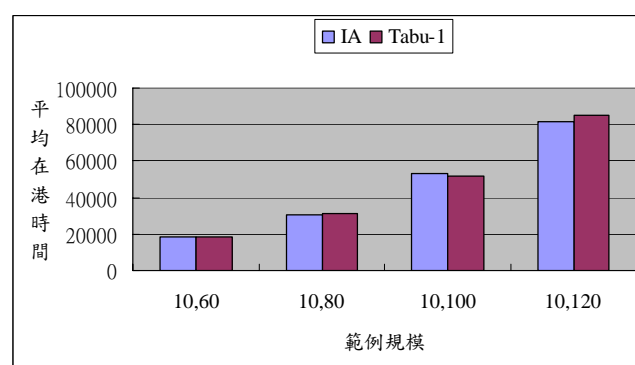
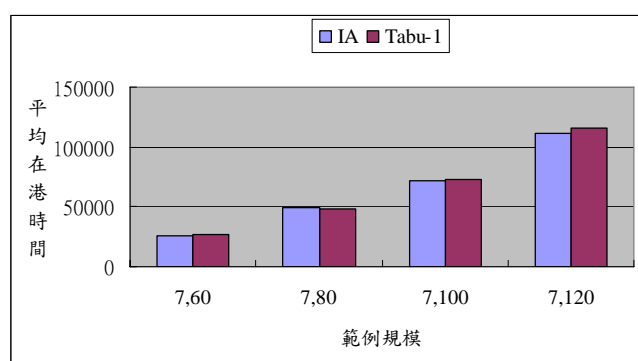


圖 5-2 不同規模範例下演算法求解品質比較

5.5 多目標船席指派求解流程

此一小節將會以本研究所提出的求解方法，來說明整個多目標的求解流程，例題仍延用附錄二 5 個船席，50 艘船舶的例題，並以「類免疫演算法」進行求解，希望藉由本研究所構建的多目標規劃，在滿足航商、港埠管理當局的目標下，藉由數理方法規劃輔助，協助其進行船席指派，達到較佳的決策。

本研究另一主要目的，係應用類免疫演算法於船席指派問題的求解，在上一小節已經與 Tabu-1 做過比較，其求解品質不差。禁忌搜尋法本身已被廣泛應用，因此，吾人欲嘗試以一種較新且求解品質不差的演算法進行求解。

5.5.1 案例內容

輸入：船舶預定到達時間等資訊(如附錄二)

目標函數：模式詳細內容可參照 3.4 節

目標一：船舶總在港時間

目標二：港埠當局營運成本

其他限制：船席水深與船舶吃水之配合

求解結果：建議船席指派位置與順序

5.5.2 求解過程

本研究分為兩個步驟：先就各個目標求其滿意度最高及最低解，繼之代入多目標求兩個目標函數下之均衡解。兩個步驟求解內容分別如下所述：

- (1) 以多目標規劃方法建立報酬表，先就兩個目標式（船舶總在港時間、港埠當局使用成本）分別求其滿意度最高解及滿意度最低解。為求滿意度最低及最高解，本研究先行就兩個目標式，以撰寫之類免疫法分別就兩個目標式進行測試求解，各測試 20 次，演算法的相關參數如 5.3.1 節，分別紀錄下每次測試後之最高及最低解，並由其中找到出現之最高解及最低解，作為滿隸屬函數滿意度最高及最低值。測試求解內容如表 5-14 至 5-15 所示，各目標式隸屬函數圖如圖 5-3 至 5-4 所示。

表 5-14 第一目標式測試求解值

第一目標式：船舶在港總時間					
測試次別	最低值 f_1^-	最高值 f_1^+	測試次別	最低值 f_1^-	最高值 f_1^+
1	100537	148981	11	98541	149198
2	96452	151571	12	98049	148775
3	99345	151531	13	101610	145630
4	98454	147478	14	103587	146605
5	100040	144006	15	100439	147661
6	99427	146592	16	103393	145463
7	100428	146319	17	97845	148953
8	102756	145736	18	103759	144660
9	101141	146304	19	98536	148680
10	96377	141672	20	106349	145985
隸屬函數上界 $f_1^+ = 151571$					
隸屬函數下界 $f_1^- = 96377$					

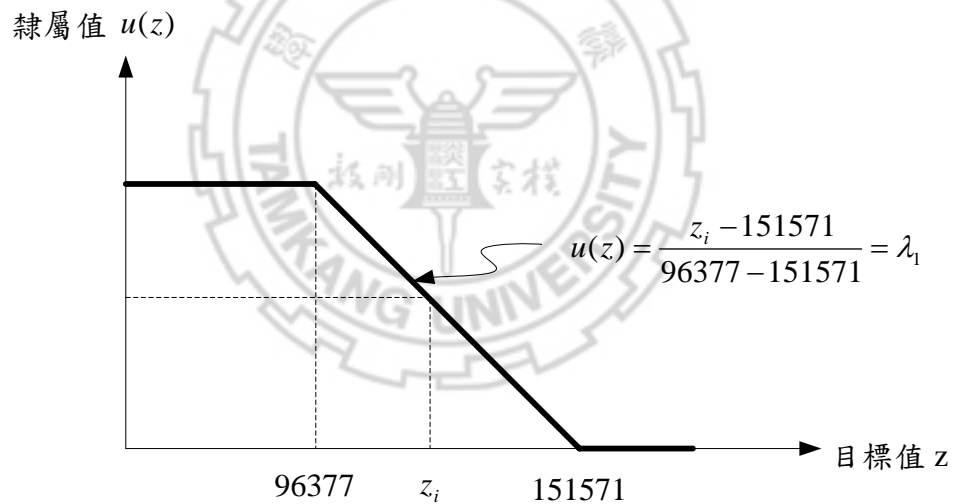


圖 5-3 第一目標式隸屬函數圖

表 5-15 第二目標式測試求解值

第二目標式：港埠當局使用成本					
測試次別	最低值 f_2^-	最高值 f_2^+	測試次別	最低值 f_2^-	最高值 f_2^+
1	4953	6711	11	4845	6808
2	5019	6930	12	4955	7045
3	4752	6915	13	4842	6903
4	4849	6979	14	4936	6983
5	4869	6791	15	5001	7042
6	4904	6744	16	4842	6903
7	4869	6757	17	5032	6970
8	4909	6857	18	4916	6815
9	4978	6757	19	4782	6859
10	4677	6452	20	4839	6916
隸屬函數上界 $f_2^+ = 7045$					
隸屬函數下界 $f_2^- = 4677$					

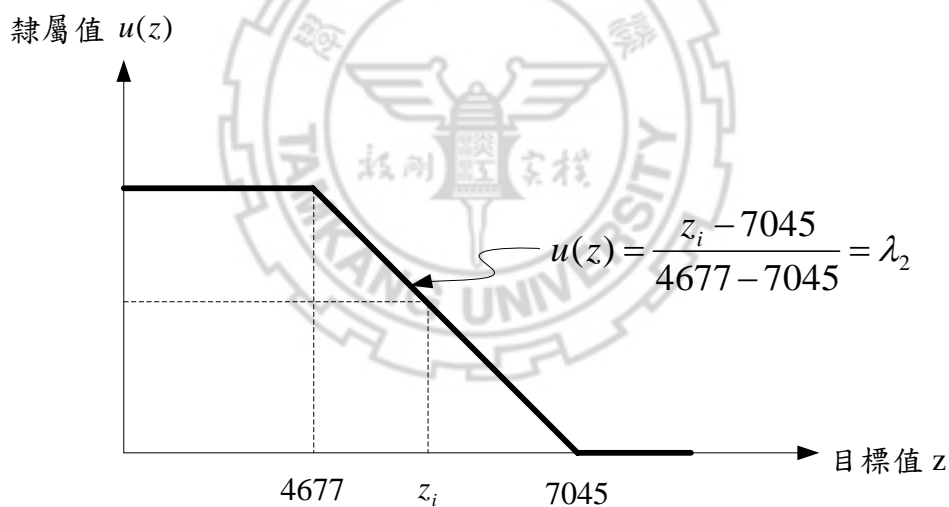


圖 5-4 第二目標式隸屬函數圖

- (2) 將兩個目標式之滿意度最高解及最低解紀錄後置入解題程式中，作為兩個目標均衡解函數之上下界。起始抗體產生後，分別求其兩個目標值 Z1 至 Z2，而由 Z1 至 Z2 中找出最低者作為該群體之均衡解 λ 值，而每一代中之 λ 值則取所有群體中最大者為代表，如此反覆循環運算， λ 值會呈現收斂狀態。

5.5.3 求解結果

多目標船席指派的求解，以 5 個船席，50 艘船舶的例題為例，運算的最大迭代數為 500，運算時間相當快速，大約為 5~10 秒，最佳隸屬函數值為 0.95，每一個迭代多目標隸屬函數值如附錄四所示，求解結果趨勢如圖 5-5 所示，輸出的建議船席指派位置與順序則如附錄五所示。

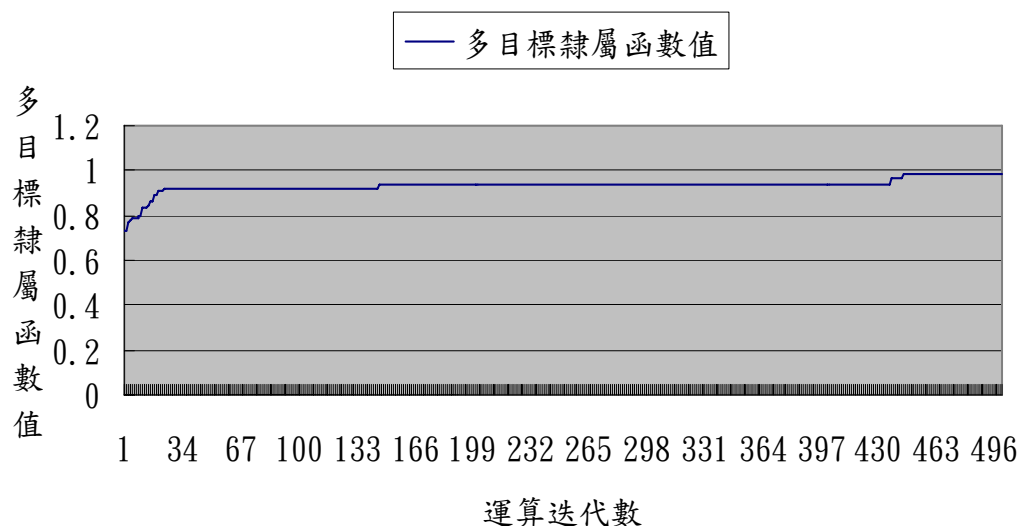


圖 5-5 多目標運算迭代圖

除附錄二大型例題外，本研究另外設定不同規模(船席數、船舶數)、船舶到達時間呈現不同分配分配的例題，並以類免疫演算法與禁忌搜尋法(Tabu-1)求解多目標船席指派，結果整理如表 5-16。

從表可以得知，在不同問題規模下、船舶到達時間呈現不同分配下，多目標船席指派求解結果，IA 最大隸屬函數的範圍在 0.81~0.98 之間，Tabu-1 最大隸屬函數的範圍在 0.79~0.96 之間，兩者求解結果相近，兩種演算法於多目標收斂之 λ 值最大差距為 0.15，最小差距為 0.09。

此外，並以 T 檢定來檢定母體平均數的差異，T 顯著代表變數一的平均值大於變數二的平均值，用此來檢定多目標下，類免疫演算法與禁忌搜尋法求解上平均數差，從表 5-16 得知，兩種演算法在不同問題規模、船舶到達時間求解上並無顯著差異。

表 5-16 不同船舶到達分配下多目標求解結果

說明：灰色區塊為 IA 求解之 λ 值大於 Tabu-1 的部份 *表示顯著						
例題規模	多目標求解收斂 λ 值					
(船席數，船舶數)	Uniform		normal		poisson	
	IA	Tabu-1	IA	Tabu-1	IA	Tabu-1
(5,25)	$f_1^+ = 36780$ $f_1^- = 24534$		$f_1^+ = 40255$ $f_1^- = 26031$		$f_1^+ = 37549$ $f_1^- = 26713$	
	$f_2^+ = 3147$ $f_2^- = 1999$		$f_2^+ = 3099$ $f_2^- = 1968$		$f_2^+ = 3057$ $f_2^- = 1950$	
T 值	-1.571		-2.05		-1.56	
λ 值	0.91	0.92	0.88	0.93	0.98	0.96
(7,35)	$f_1^+ = 48033$ $f_1^- = 34412$		$f_1^+ = 52303$ $f_1^- = 37126$		$f_1^+ = 66668$ $f_1^- = 37541$	
	$f_2^+ = 4188$ $f_2^- = 2526$		$f_2^+ = 4046$ $f_2^- = 2617$		$f_2^+ = 4135$ $f_2^- = 2697$	
T 值	-1.49		-2.23		-0.99	
λ 值	0.96	0.94	0.90	0.96	0.83	0.79
(5,50)	$f_1^+ = 152425$ $f_1^- = 96377$		$f_1^+ = 176288$ $f_1^- = 112892$		$f_1^+ = 172639$ $f_1^- = 107170$	
	$f_2^+ = 7045$ $f_2^- = 4677$		$f_2^+ = 7330$ $f_2^- = 4994$		$f_2^+ = 7327$ $f_2^- = 5002$	
T 值	-5.53		-7.89		-5.18	
λ 值	0.98	0.89	0.81	0.96	0.86	0.88

5.6 實證研究與參數敏感度分析

本研究以基隆港貨櫃碼頭船席指派資料紀錄作為實證資料，在收集資料後，進行本研究多目標下船席指派的求解，求解的方式如 5.5.3 節所述。實證流程如圖 5-6 所示。

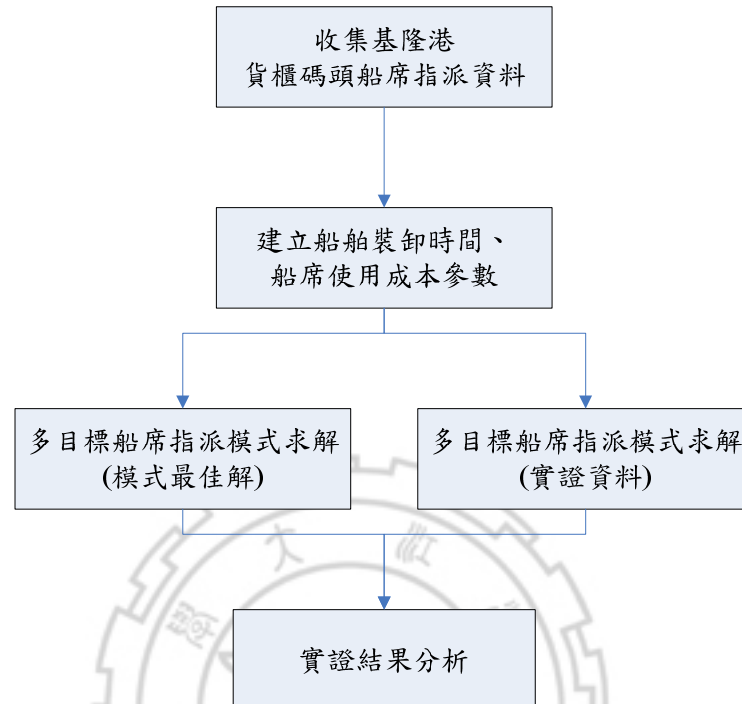


圖 5-6 實證流程圖

5.6.1 資料收集與比較方式

本研究的研究範圍係針對貨櫃碼頭船席，因此，在資料的蒐集上，非貨櫃船資料將不在本研究的搜集範圍內；從基隆港務局網站得知，其貨櫃碼頭船席一共有 15 個，東岸有四個，東岸租予聯興公司，西岸為基隆港務局主要管理部份，因此，將會以西岸 11 個碼頭船席資料進行蒐集。

本研究在模式建立時，假設船舶裝卸貨時間會隨船席不同，船席使用成本也會因為指派結果而有所變異，因此，從基隆港所須要蒐集的資料項目為：船席數量、船舶數量、船舶預期到達的時間與過去的指派紀錄，而成本項目、船舶裝卸貨時間皆會自行假設。

本研究與實證資料的比較方式為：

1. 以類免疫演算法求解出多目標船席指派之最佳隸屬函數
2. 求出實證資料指派順序下之多目標隸屬函數
3. 比較兩者求解的結果

5.6.2 實證比較的結果

本研究搜集了基隆港民國 95 年 5 月 22 日至民國 95 年 5 月 28 日船席指派紀錄，以實務中的船舶預定到達時間、吃水深度與碼頭水深當作輸入項目，以 5.5.2 節求解方式，求出多目標隸屬函數上下界，並求出最大隸屬函數；接著利用實務中之指派結果求出另一最大隸屬函數，並比較兩者的差異，求解的結果如表 5-17。從表中我們可以得知，本研究所構建之模式指派結果於單目標、多目標下之隸屬度均優於會議之調度。

在模式的求解、船席會議求解各目標隸屬函數時，隸屬函數上下界 f_1^+ 、 f_1^- 、 f_2^+ 與 f_2^- ，係利用模式求解所得出的上下界，以此方式的原因為實務上兩個目標之上下界並不容易得知，而此方式會使得兩者（會議、模式）隸屬函數的上下界均相同。

表 5-17 船席會議指派與模式指派結果比較表

	船席指派會議結果	模式最佳指派結果
在港總時間隸屬度 λ_1	0.15	0.75
營運成本隸屬度 λ_2	0.22	0.81
λ	0.15	0.75

5.6.3 參數敏感度分析

本節針對港口營運時可能產生變動的參數進行敏感度分析，主要目的係在測試模式在不同參數值下的反應，並可提供港務局作為調整的依據。本模式中，可能變動的相關參數為船舶裝卸貨時間、港埠當局使用成本，因此，將會進行此兩方面的敏感度分析。本研究利用上一小節模式所求出的最佳指派順序，並針對服務船舶艘次較多與較少的船席進行船席的使用成本、服務時間的變動，觀察不同的參數變動下，指派順序有何改變。

分析的方法為，設計不同的船舶裝卸貨時間、使用成本，並求出多目標模式下的最佳指派順序後，並與參數未變動前的指派結果做比較。表 5-18 為敏感度分析的整理表。本研究在成本、時間上均為自行假設，將參數變小的方法為乘以原有值的 0.1 倍，而參數放大的方式則是乘以 2 到 5 倍的範圍，經過多次測驗後發現，船席服務的船舶數與參數的變動成反向變動。

此結果可以說明，模式之使用成本、裝卸貨時間增加時，航商、港埠管理當局會選擇時間、成本較低的船席，此一結果與一般邏輯吻合，即本模式可以反映出合乎實務中的現況反應。

表 5-18 船席指派參數敏感度分析

碼頭編號		原有的服務船舶數	參數變動後服務船舶數
W16、W17	增加服務時間	18	12
	增加使用成本		11
W19、W20	減少服務時間	10	16
	減少使用成本		13
W16	增加服務時間	10	6
	增加使用成本		6
W26	減少服務時間	7	9
	增加使用成本		3



5.7 小結

本章主要內容有(1)模式的確認(2)演算法正確度測試(3)演算法較佳參數的選擇(4)演算法之比較(5)演算法應用於本研究所構建的多目標模船席指派求解流(6)IA、Tabu-1 求解多目標船席指派結果比較(7)實證研究與參數敏感度分析。

模式的確認，經由四個小型範例，以窮舉的方式，並利用 Dev-C++函式庫中 `next_permutation()` 函式，求出模式中的最佳解，証實了模式在小型範例中的正確性。

演算法正確度的測試上，以「類免疫演算法」與「禁忌搜尋法」來求解相同的四個小型範例，個別目標下之求解結果，皆可以得到與窮舉法相同的最佳解，至於在多目標的指派結果中，除了兩個求解結果不同之外，其他都與窮舉法的最佳解相同；除了求解品質之外，也將禁忌演算法與類免疫演算法的運算時間紀錄，兩者的運算時間都非常迅速，可以很快的得到結果，不必像窮舉法於求解範例二、範例四時需要相當長的運算時間。

在演算法的參數敏感度分析中，求出兩種演算法較適當的參數；類免疫演算的最佳參數為起始抗體數目大小 60、輕鏈的重組率 0.5、重鏈重組率 0.8、記憶區 A 抗體大小(輕鏈重組)與記憶區 B 抗體大小(重鏈重組)選出抗體數量的比例為 1:3，最佳迭代數則為 900。禁忌演算法的最佳參數則為禁忌名單為 5，鄰域搜尋次數則為 500。

範例(5,50)，Tabu-2 在求解的品質上最佳，經過統計檢定比較 Tabu-1、IA，發現 Tabu-1 有較佳的求解品質；在 8 種不同問題規模中，IA 於大型規模範例下，有較佳的求解品質。

在不同問題規模下、船舶到達時間呈現不同分配下，多目標船席指派求解結果，IA 最大隸屬函數的範圍在 0.81~0.98 之間，Tabu-1 最大隸屬函數的範圍在 0.76~0.95 之間，求解結果相近，相同範例下，兩種演算法多目標收斂之 λ 值最大差距為 0.15，最小差距為 0.09，以統計上 T 檢定兩母體平均數差來檢定兩種演算法於多目標的求解結果為兩者並無顯著的差異。

實證研究後，本研究於多目標模式下指派結果所求出的 λ 值為 0.75，船席指派會議所得出的 λ 則為 0.15，本研究所構建之模式指派結果優於會議之調度結果。

至於在參數敏感度方面，經由多次測試，可以發現船席服務的船舶數與服務時間、使用成本成反向變動，此合乎一般的邏輯性，更可證明本模式正確性，具有應用價值。

第六章 結論與建議

現今國際貿易而言，所使用的運輸方式仍以海運為主，且有不斷增加的趨勢，也為了在運輸過程中，可以迅速、安全與便利等目的，而有貨櫃化的趨勢產生，更成為現今海運中的主流。

船舶停泊作業為海運過程中首要作業，效率的好壞直接影響到後續作業的運作，而國內目前在停泊作業上，航商、代理行與港口船席調配小組以會議的方式進行，在避免擁擠、充分運用船席、增進港埠營運效能等目標下，指派船舶到適當的船席停靠；因此，有效提高港口的作業效率與資源規劃，不僅可以幫助港埠當局有效的使用資源，更可以吸引更多的船舶來港進行作業，進而提高港口本身的競爭力。

實務上，船席指派此種決策為航商、港埠單位所做出的決策，本研究在整理過去國內外文獻後，發現大都是以航商的角度為出發點，或是將船席視為離散、連續等進行探討，尚未發現有以兩者的角度為出發點進行探討。因此，本研究以多目標結合車輛繞徑的觀點來構建模式。

基因演算法為啟發式解法之一，在求解 NP-hard 此類的問題上，已被廣泛的應用並早已被證明為解決此等問題的最佳方法之一，類免疫演算法的概念與基因演算法非常相似，且文獻中對於其評價，似乎有比基因演算法更佳的求解效果，因此，本研究在航商、港埠當局整體目標的考量下，針對問題發展適合的求解方法，作為實務單位決策之參考。

6.1 結論

(1)問題的定義上：

國內在針對船席指派的問題上相關研究上，探討的方式大都以最佳化或是模擬的方式，而其觀點大都是以航商的角度為出發點。本研究基於實務上船席指派是由兩者所做出的決策，因此，考量航商的總在港時間、港埠當局的營運成本下，構建多目標模式來進行探討。

(2)模式的構建上：

本研究嘗試從車輛繞徑模式來處理船席指派問題。由於一艘船舶只會被服務一次，且每個船舶皆有其預定到達時間與預訂離開時間，此與車輛繞徑問題中的需求顧客相當類似。

因此，本研究將「船席」視為車輛繞徑問題中的「車輛」，「船舶」視為車輛繞徑中的「需求點」，船舶被指派到船席的順序，視為車輛繞徑中繞徑的順序，並假設營運者於不同需求點，會有不同的營運成本、服務時間，而每個需求點都有其時窗限制時間；並且考量船席水深與船舶吃水此條件限制，期望在同時考量營運者與顧客的角度下，求出最佳的指派順序。

(3)求解演算法的設計上：

本研究在演算法的設計上，以類免疫演算法來求解本研究的船席指派問題，資料結構部份，是以結構體的方式來儲存資料，而起始抗體的產生，係以一次產生一組解的方式。而選出抗體方面，首先以抗體的績效來選出優良的抗體，經過抗體的增值分裂為兩個群體，並進行抗體輕鏈的重組、重鏈的重組後，從兩個群體中，以蒙地卡羅輪盤方式，依照所設定的抗體比例數選出抗體，作為下一迭代演算的基礎。

(4)模式驗證上與演算法正確性的測試：

船席指派的文獻相當豐富，但是尚未有類似的案例可供測試作為比較，因此本研究以四個小型的例題，進行模式的驗證，利用 C 語言中 next_permutatio() 輔助窮舉法的演算，證實了模式在小規模下的正確性(有解)；此外並以此為基準，進一步做禁忌演算法與類免疫演算法演算的正確性。個別目標下，禁忌搜尋法與類免疫演算法於小型題目中，皆可以求出最佳解。

(5) 案例測試方面：

案例測試有四個小型案例、大型案例、船舶到達時間呈現不同分配下的案例，小型案例中，是以窮舉法的方式來確認模式的正確性；並以大型案例來進行演算法求解的比較並且求出指派結果，最後設計了不同類型的案例進行多目標船席指派的求解。

(6) 演算法的比較：

範例(5,50 中)，本研究對於禁忌演算法搜尋鄰近解的步驟中，分為「隨機性鄰域搜尋」與「窮舉式鄰域搜尋」，一共撰寫三種演算法 IA、Tabu-1、Tabu-2 程式，求解品質以 Tabu-2 為最佳，但是其運算時間約為 IA、Tabu-1 的 48 倍左右 $((8*60*60)/(20*30))$ ，此外，以統計 T 檢定兩母體平均數差的結果，Tabu-1 於(5,50)範例中有較佳的求解品質。本研究再另外自行設計的 8 種不同問題規模範例進行「單目標求解」，IA 於大型規模範例下，有較佳的求解品質。

(7) 不同案例求解：

自行設計三種不同問題規模、三種不同船舶到達時間一共 9 個案例，分別以禁忌演算法(Tabu-1)、類免疫演算法進行「多目標船席指派問題的求解」，以模糊多目標規劃的方式求得收斂之 λ 值以作為案例輸出最佳指派順序之代表，輸出資料於附錄六。

在不同問題規模下、船舶到達時間呈現不同分配下，以統計上 T 檢定兩母體平均數差來檢定兩種演算法於多目標的求解結果發現兩者並無顯著的差異。

(8)實證研究與參數敏感度分析：

本研究利用基隆港過去的指派資料，並與模式求解出指派結果做一比較，發現模式所求出的解優於船席會議指派的結果。

參數敏感度方面，經由多次測試模式在不同參數下的反應，發現船席服務的船舶數與服務時間、使用成本成反向變動，此合乎一般的邏輯性，更可證明本模式正確性，具有應用價值。

(9)應用方面：

以現有的資訊安排預定到達的船舶進行指派，於多目標的考量下，可以有效、迅速的求解出船席指派順序，供港埠當局、航商作為決策之參考。

6.2 建議

- (1) 本研究並未限定每個船席所可以處理的貨櫃量，後續的研究可以把違反車輛容量限制的假設加到目標式中進行探討，即限定每個船席有其一定處理貨櫃量的限制加入到模式中進行探討。
- (2) 實務中，會有船席閒置的情況產生，可以把此一情況加到模式中進行探討，例如將船席的閒置時間列入考量，使船席閒置時間減少。
- (3) 在成本方面，節線的成本，即船席服務船舶的成本，本研究是以假設的方式進行探討，後續的研究可以針對此一方面，估計出較正確的參數。
- (4) 對於求解過程中轉換模糊多目標中，各個目標的上下界是經由演算法測試多次後所找出，在此方面，後續可以用問卷調查或是經驗值得出目標值期望水準(下界)及可接受水準(上界)。
- (5) 本研究以車輛繞徑的觀點構建模式，將此一問題視為靜態車輛繞徑問題，後續的研究可以將需求點數目(船舶數)視為動態的問題進行探討，或者是將車輛於需求點的作業時間視為不確定性，例如船席的服務時間假設為常態分配等進行探討，使能更與現實世界相符合。
- (6) 本模式並未考慮類似機門指派的「緩衝時間」，後續的研究可以將此一因素納入考慮，並進行探討。
- (7) 實證研究中，模糊隸屬函數的上下界係採用模式所求出的上界與下界，若是能以問卷(或是其他方式)求出實務中的上下界，再進行比較，較能突顯模式與實務真正差異性。

参考文献

1. Alan S. Perelson and Gerard Weisbuch. Immunology for physicists. Preprint for Review of Modern Physics, June 1995.
2. Ali Haghani and Min-Ching Chen (1998), "Optimizing Gate Assignments at Airport Terminals", Transportation Research-A, Vol.32, No.6, 1998, pp.437-453.
3. Barbarosoglu, G. and Ozgur, D. (1999), "A tabu search algorithm for the vehicle routing problem," Computer & Operations Research, 26, pp.255-270.
4. Bodin, L., B. Golden, A. Assad & M. Ball (1983), "Routing and Scheduling of Vehicle and Crew: The State of Art," Special Issue of Computers and Operations Research, Vol.10, No.2, pp.63-211
5. Brandao, J., Mercer, A. (1997), "A tabu search algorithm for the multi-trip vehicle routing and scheduling problem," European Journal of Operational Research, 100, pp.180-191.
6. Brown, G. G., K. J. Cormican, S. Lawphongpanich and D. B. Widdis (1997), "Optimizing submarine berthing with a persistence incentive," Naval Research Logistics, 44, 301-318.
7. Dasgupta, D. "Artificial Immune Systems and Their Applications", Springer-Verlag, 1999.
8. Dasgupta, D., "Artificial Neural Networks and Artificial Immune Systems : Similarities and Differences," Proceeding of the IEEE Transaction on Systems, Man and Cybernetics, Vol. 1, pp.873-878, 1997.
9. Dasgupta, D., and Atttoh-Okine, N., "Immunity-Based Systems : A Survey," Proceeding of the IEEE Transaction on Systems, Man and Cybernetics, Vol. 1, pp.369-374, 1997.
10. de Castro, L. N., and Timmis, J., "Artificial Immune Systems : A New Computational Intelligence Approach," Springer, 2002.
11. de Castro, L. N., and von Zuben, F. J., "Artificial Immune Systems : Part I-Basic Theory and Applications," Technical Report TR-DCA 01/99, 1999.
12. de Castro, L. N., and von Zuben, F. J., "Artificial Immune Systems : Part II-A Survey of Applications," Technical Report DCA-RT 02/00, 2000.
13. de Castro, L. N., and von Zuben, F. J., "The Clonal Selection Algorithm with Engineering Application," Proceedings of the Genetic and Evolutionary Computation Conference, Vol. 1, pp.36-37, 2000.
14. Dote, Y., "Soft Computing (Immune Networks) in Artificial Intelligence," Proceeding of the IEEE Transaction on Systems, Man and Cybernetics, Vol. 2, pp.1382-1387, 1998.
15. Forrest, S., & Perelson, A., "Genetic algorithms and the immune system.", In H. Schwefel, & R. Maenner (Eds.), Parallel Problem Solving from Nature, Berlin. Springer-Verlag (Lecture Notes in Computer Science), 1991
16. Fukuda, T., Mori, K. & Tsukiyama, M., Parallel search for multi-modal function optimization with diversity and learning of immune algorithm, Artificial Immune Systems and Their Application, Springer-Verlag, pp. 210-220, 1999.
17. Fukuda, T., Mori, M., Tsukiyama, M., "Immunity-Based Management System for a Semiconductor Production Line", Artificial Immune Systems and Their Applications, pp: 278~288, Springer-Verlag, 1999
18. Gendreau, M., Hertz, A., Laporte, G. (1994), "A tabu search heuristic for the vehicle routing problem," Management Science, Vol. 40, No. 10, pp.1276-1290.
19. Geoffrey W. Hoffmann. A neural network model based on the analogy with the immune system. Journal of Theoretical Biology, 122:33-67, 1986.

20. Glenn W. Rowe. The Theoretical Models in Biology. Oxford University Press, first edition, 1994.
21. Glover, F. & M. Laguna (1997), Tabu Search, Kluwer Academic Publishers, Massachusetts.
22. Glover, F. (1995), Tabu Search Fundamentals and Uses, Work Report, Graduate School of Business, University of Colorado.
23. Glover, F.(1986), "Future Paths for Integer Programming and Links to Artificial Intelligence," Computers & Operations Research, Vol.13, pp.533-549.
24. Han Shenglian, Ni Meng, and Ge Wancheng, Problems in GA and necessity of importing immune function, Intelligent Control and Automation, 2000. Proceedings of the 3rd World Congress on, Volume 1, pp.542-544, 2000.
25. Imai, A., E. Nishimura and S. Papadimitriou (2001), "The dynamic berth allocation problem for a container port," Transportation Research Part B, 35, 401-417.
26. Imai, A., E. Nishimura and S. Papadimitriou (2003), "Berth allocation with service priority," Transportation Research Part B, 37, 437-457.
27. Imai, A., K. I. Nagaiwa and C. W. Tat(1997), "Efficient planning of berth allocation for container terminals in Asia," Journal of Advanced Transportation, 31, 75-94.
28. Ishida, Y., Hirayama, H., Fujita, H., Ishiguro, A. and Mori, K. "Immunity-Based Systems—Intelligent Systems by Artificial Immune Systems", Corona Pub. Co. Japan(in Japanese),1998.
29. Ivan Roitt, Jonathan Brostoff, and David Male (1998) "Immunology,5/e", Original English edition Copyright Mosby International Ltd.
30. Jiao Licheng and Wang Lei, A novel genetic algorithm based on immunity, Systems, Man and Cybernetics, Part A, IEEE Transactions on, Volume 30 Issue 5, pp.552-561, Sept. 2000.
31. John E. Hunt and Denise E. Cooke, Learning using an artificial immune system, Journal of Network and Computer Applications, pp. 189-212, 1996.
32. Jones, D.F., Mirrazavi, S.K. and Tamiz, M.(2002), "Multi-objective meta-heuristics: An overview of the current state-of-the-art ", European Journal of operational Research, Vol. 137, pp.1-9.
33. Kim, K. H. and K. C. Moon(2003), "Berth scheduling by simulated annealing," Transportation Research Part B, 37, 541-560.
34. Lai, K. K. and K. Shih (1992), "A study of container berth allocation," Journal of Advanced Transportation, 26, 45-60.
35. Lei Wang and Licheng Jiao, A novel genetic algorithm based on immunity, Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on, Volume 5, pp.385-388, 2000.
36. Lim, A.(1998), "The berth planning problem," Operations Research Letters, 22, 105-110.
37. Mori, K., Tsukiyama, M., and Fukuda, T., "Adaptive Scheduling System Inspired by Immune System," Proceeding of the IEEE Transaction on Systems, Man and Cybernetics, Vol. 4, pp.3833-3837, 1998.
38. Mori, M., Tsukiyama, M., Fukuda, T., "Immune Algorithm with searching Diversity and Its Application to Resource Allocation Problem", Trans. IEE Japan, 113-C(10):872~878, 1993
39. N. K. Jerne (1984) "Idiotypic networks and other preconceived ideas", Immunological Rev., Vol.79, 5-24.
40. Nishimura, E., A. Imai and S. Papadimitriou(2001), "Berth allocation planning in

- the public berth system by genetic algorithms,” *European Journal of Operational Research*, 131, 282-292.
41. Nishimura, E., 1998. Public use effects on the performance of container berth. Master Thesis, Kobe University of Mercantile Marine, Japan (in Japanese).
 42. Renaud, J., Laporte, G., F. Boctor, F.F.(1996), “A tabu search heuristic for the multi-depot vehicle routing problem,” *Computer & Operations Research*, Vol. 23, No. 3, pp.229-235.
 43. S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri. Self-Nonself Discrimination in a Computer. In *Proceedings of IEEE Symposium on Research in Security and Privacy*, pages 202-212, Oakland, CA, 16-18 May 1994.
 44. S. Bandara and S.C Wirasinghe, “Airport Gate Position Estimation Under Uncertainty”, *TRR* 1199, 1988, pp.41-48.
 45. S.C Wirasinghe and S. Bandara, “Airport Gate Position Estimation For Minimum Total Costs-Approximate Closed Form Solution”, *Transportation Research-B*, Vol.24B, No.4, 1990, pp.287-297.
 46. Steven A. Frank. *The Design of Natural and Artificial Adaptive Systems*. Academic Press, New York, M. R. Rose and G. V. Lauder edition, 1996.
 47. Gioia Tauro, “Modelli ed Algoritmi per un Terminale Marittimo Container.”
 48. 卓裕仁，以巨集啟發式方法求解多車種與週期性車輛路線問題之研究，國立交通大學運輸工程與管理系碩士論文，2001
 49. 廖子銘，類免疫演算法於多目標最佳化問題之研究與應用，大同大學機械工程研究所碩士論文，2001
 50. 張世峰，即時訂貨資訊下物流配送作業規劃之研究，淡江大學運輸科學研究所碩士論文，2002
 51. 張企宏，貨櫃碼頭分派問題之研究，元智大學工業工程與管理學系碩士論文，2004
 52. 張家銘，機門指派最佳化之研究，國立中央大學土木工程研究所碩士論文，1995
 53. 張東珍，機場機門指派模擬模式之研究，國立交通大學交通運輸研究所碩士論文，1993
 54. 張立偉，災後工程緊急搶修作業排程之研究，淡江大學運輸科學研究所碩士論文，2001
 55. 徐玉琦，基隆港貨櫃裝卸作業費率之研究-作業基礎成本制度(ABC)之應用，國立交通大學交通運輸研究所碩士論文，1998
 56. 朱橋榮，最小化機場機門數量之研究，國立中央大學土木工程研究所碩士論文，2001
 57. 李育達，基隆港貨櫃船席調配電腦化之分析，國立海洋大學航運技術研究所碩士論文，1998
 58. 杜世文，多目標與模糊時窗貨物配送啟發式解法之研究，國立交通大學交通運輸工程研究所碩士論文，1991
 59. 林明俊，隨機環境下多車種車派車問題之研究，中原大學工業工程研究所碩士論文，1998
 60. 林秀璘，應用模糊多目標規劃法於公車營運計畫之研究，國立成功大學交通管理(科學)學研究所碩士論文，1992
 61. 楊博文，配合大眾捷運系統之接運公車營運計劃整合之研究--分析數學法與多目標規劃法之應用，國立成功大學交通管理(科學)研究所碩士論文，1990

62. 沈進成，公車系統營運計劃最佳化模式之研究，國立成功大學交通管理(科學)研究所碩士論文，1988
63. 洪子盛，以模糊多目標規劃法求解消防站配置問題之最佳化，國立成功大學工業管理科學系博士論文，2003
64. 游守田，停機坪即時性調度之研究，國立成功大學交通管理科學研究所碩士論文，2004
65. 遊建堂，應用塔布搜尋法於多目標最佳化問題，淡江大學電機工程學系碩士在職專班碩士論文，2004
66. 游文松，公路客運人員與車輛排班之研究，中華大學科技管理研究所碩士論文，2004
67. 王丘明，港埠管理，2003，華泰
68. 王文貞，圖書配送車輛排程問題之研究，成功大學交通管理研究所碩士論文，1997
69. 羅意茹，應用免疫演算法於可控退化率存貨問題之研究，嘉義大學運輸與物流工程研究所碩士論文，2004
70. 許志義，多目標決策增訂版，2003
71. 謝東緯，大專學生參與專題研究計畫成果報告之「船席調派問題之研究-以基隆港為例」，國立成功大學交通管理學系，1997
72. 謝致遠，「靜態機門指派與即時營運擾動間相關性之研究，國立中央大學土木工程學系碩士論文，1998
73. 鄧宏安，整合 CFDRC 及類免疫演算法於散熱片之最佳化設計，大同大學機械工程研究所碩士論文，2003
74. 陳勝男，禁忌搜尋法應用於車輛路線問題之研究，大葉大學工業工程研究所碩士論文，1996
75. 陳春益、李宇欣、盧華安，民國八十六年，「時空網路應用於機門指派問題之研究」，運輸學刊，第 10 卷，第 3 期，頁 1-20
76. 陳朝文，基隆港貨櫃船舶優先靠泊制度之研究，國立海洋大學航運管理研究所碩士論文，2002
77. 霍俊明，機門指派最佳化模式暨變數產生法之研究，國立中央大學土木工程學系碩士論文，1983
78. 黃明居，知識庫系統應用於基隆港貨櫃船席調配作業之研究，國立海洋大學航運技術研究所碩士論文，1990
79. 黃木才，貨櫃運輸公司車輛排程問題之研究-模糊多目標遺傳演算法之應用，國立交通大學交通運輸研究所碩士論文，1995
80. 黃金智，隨機型車輛途程問題解法之研究，大葉大學工業工程研究所碩士論文，1999

附錄一、小型範例

範例一：(船席數,船舶數)=(2,4)

假設在某個港口中有兩個船席，而預定到達的船舶數量有四艘，相關的資訊如表一所示。

表一 範例一相關資訊表

船舶資訊表					
船舶編號	預計到達時間	船舶吃水			
1	1	7			
2	14	12			
3	21	9			
4	33	13			

船席水深與船舶作業時間表					
船席編號	船席水深	船舶編號			
		1	2	3	4
		作業時間			
1	12	7	3	12	4
2	13	13	8	7	6

船席使用成本表				
船席編號	船舶編號			
	1	2	3	4
	作業成本			
1	60	40	50	30
2	50	30	60	45

範例二：(船席數,船舶數)=(2,8)

假設在某個港口中有兩個船席，而預定到達的船舶數量有八艘，相關的資訊如表二所示。

表二 範例二相關資訊表

船舶資訊表		
船舶編號	預計到達時間	船舶吃水
1	5	7
2	37	6.4
3	7	6
4	30	6
5	1	7
6	23	8
7	45	6.4
8	16	8

船席水深與船舶作業時間表									
船席編號	船席水深	船舶編號							
		1	2	3	4	5	6	7	8
		作業時間							
1	12	10	14	28	2	30	15	19	28
2	8	15	10	18	6	20	20	25	39

船席使用成本表								
船席編號	船舶編號							
	1	2	3	4	5	6	7	8
	作業成本							
1	13	15	20	35	34	8	40	6
2	20	10	10	39	21	17	28	15

範例三：(船席數,船舶數)=(3,5)

假設在某個港口中有三個船席，而預定到達的船舶數量有五艘，相關的資訊如表三所示。

表三 範例三相關資訊表

船舶資訊表						
船舶編號	預計到達時間	船舶吃水				
1	5	9.1				
2	32	10.6				
3	73	8.5				
4	66	7.3				
5	18	7				

船席水深與船舶作業時間表						
船席編號	船席水深	船舶編號				
		1	2	3	4	5
		作業時間				
1	12	45	48	14	20	35
2	10	50	55	25	30	50
3	8	35	40	8	10	20

船席使用成本表						
船席編號	船舶編號					
	1	2	3	4	5	
	作業成本					
1	36	81	45	66	50	
2	15	60	70	80	80	
3	50	30	20	30	100	

範例四：(船席數,船舶數)=(3,7)

假設在某個港口中有三個船席，而預定到達的船舶數量有七艘，相關的資訊如表四所示。

表四 範例四相關資訊表

船舶資訊表								
船舶編號	預計到達時間	船舶吃水						
1	89	9						
2	98	13						
3	236	10						
4	227	9						
5	173	9						
6	275	8						
7	95	12						

船席水深與船舶作業時間表								
船席編號	船席水深	船舶編號						
		1	2	3	4	5	6	7
		作業時間						
1	13	638	223	584	304	516	249	316
2	14	329	321	444	737	709	567	498
3	10	793	376	332	754	413	781	328

船席使用成本表							
船席編號	船舶編號						
	1	2	3	4	5	6	7
	作業成本						
1	88	198	52	75	66	157	97
2	203	59	211	200	157	118	179
3	160	241	139	80	233	52	85

附錄二、大型例題(5 , 50)相關資訊

船舶編號	預計到達時間	船舶吃水	船舶編號	預計到達時間	船舶吃水
1	222	10	26	552	8
2	543	9	27	123	14
3	451	8	28	152	10
4	369	9	29	402	13
5	412	9	30	376	13
6	288	8	31	528	9
7	451	9	32	237	12
8	78	9	33	157	8
9	113	9	34	433	11
10	480	8	35	128	10
11	52	11	36	368	11
12	472	9	37	386	9
13	528	8	38	435	14
14	370	11	39	107	8
15	528	12	40	535	9
16	172	13	41	79	11
17	218	13	42	285	10
18	292	8	43	251	10
19	592	11	44	386	12
20	211	11	45	347	14
21	593	11	46	118	10
22	215	11	47	308	13
23	409	12	48	435	10
24	149	11	49	59	9
25	83	13	50	122	11

		船舶編號									
		1	2	3	4	5	6	7	8	9	10
船席編號	船席水深	作業時間									
1	13	421	783	466	527	733	439	560	609	289	294
2	14	555	221	560	687	651	484	244	334	363	717
3	13	740	212	467	719	462	772	452	727	747	435
4	13	402	395	406	780	362	394	742	258	672	395
5	11	454	335	465	562	720	551	276	426	526	257
		11	12	13	14	15	16	17	18	19	20
1	13	379	267	736	533	510	296	602	346	762	578
2	14	431	205	279	243	400	242	791	560	459	765
3	13	521	781	465	647	699	678	456	726	234	770
4	13	497	484	356	655	258	637	336	420	430	645
5	11	607	408	568	772	225	340	701	339	563	558
		21	22	23	24	25	26	27	28	29	30
1	13	453	726	322	638	448	412	749	400	742	214
2	14	317	587	716	412	350	349	434	270	448	277
3	13	711	306	740	281	291	254	431	648	729	409
4	13	279	701	753	792	323	238	637	497	789	360
5	11	368	217	766	735	561	781	433	592	221	712
		31	32	33	34	35	36	37	38	39	40
1	13	674	315	244	639	283	693	515	209	575	796
2	14	381	666	237	601	512	525	789	231	425	399
3	13	761	476	302	528	596	484	282	385	789	743
4	13	459	731	372	399	741	277	449	636	514	482
5	11	349	487	604	206	731	721	792	299	657	795
		41	42	43	44	45	46	47	48	49	50
1	13	343	426	611	678	513	357	302	232	700	753
2	14	439	610	538	529	751	326	598	217	468	490
3	13	259	682	266	622	461	427	335	505	277	551
4	13	568	657	732	533	224	504	462	670	612	509
5	11	420	446	727	390	538	735	708	736	681	582

		船舶編號									
		1	2	3	4	5	6	7	8	9	10
船席編號	船席水深	作業成本									
1	13	117	111	106	122	200	163	138	91	167	164
2	14	162	103	146	114	149	54	147	120	193	102
3	13	107	58	146	99	137	111	70	181	148	139
4	13	146	81	114	74	170	182	150	139	83	163
5	11	79	167	114	154	139	96	165	146	140	140
		11	12	13	14	15	16	17	18	19	20
1	13	81	123	191	153	175	144	156	72	59	64
2	14	63	68	119	196	168	54	138	123	105	152
3	13	155	190	96	96	92	161	162	126	169	194
4	13	116	131	79	145	186	104	147	197	115	68
5	11	109	165	128	172	93	109	90	64	142	113
		21	22	23	24	25	26	27	28	29	30
1	13	143	163	71	134	82	187	64	116	200	85
2	14	87	108	186	143	192	86	63	66	113	149
3	13	155	126	92	179	97	106	115	90	54	133
4	13	88	158	173	82	55	148	153	77	148	133
5	11	94	66	54	158	70	105	55	56	81	58
		31	32	33	34	35	36	37	38	39	40
1	13	101	121	128	97	168	91	82	113	120	161
2	14	190	63	118	84	140	177	98	61	97	109
3	13	95	106	173	63	90	196	53	106	116	105
4	13	174	184	101	148	96	83	135	152	61	126
5	11	94	85	174	56	74	181	56	142	71	181
		41	42	43	44	45	46	47	48	49	50
1	13	192	188	176	89	175	192	124	138	114	58
2	14	111	81	191	180	149	100	86	175	75	83
3	13	72	171	91	158	165	94	159	149	195	200
4	13	101	58	192	150	116	116	128	120	82	98
5	11	80	163	146	115	190	112	164	171	114	159

附錄三、類免疫演算法求解求解在港時間參數實驗設計

抗體數目實驗設計求解結果

15	25	40	60	15	25	40	60
104012.5	103378.9	105660.9	98023.68	102460.9	106601.3	102574.9	100933.9
105499.9	108545.7	102023.1	104002.9	103317.9	106127.3	103196.7	101339.1
103827.5	107708.9	103854.5	99731.52	105009.3	101747.1	103820.7	100355.9
104241.7	102761.3	103635.1	102976.9	109066.3	100784.5	105373.7	100336.1
109698.9	106316.9	102691.9	104449.1	109925.5	104904.9	103748.9	104182.7
103123.7	103755.5	100721.9	101511.1	102933.9	104013.9	101007.1	96285.31
107213.7	106038.7	104143.7	100376.1	104284.9	107962.3	103615.7	105955.9
110172.5	101200.3	106105.3	103841.7	106824.5	107299.3	105714.7	104150.5
106358.1	103812.6	100513.7	100179.3	112284.5	101419.5	105385.7	104334.1
102452.3	100468.9	102632.3	105591.9	106287.9	100467.3	99995.78	102183.3
108780.5	109101.9	105034.4	100863.1	107839.5	104163.9	106678.1	102792.3
105407.7	103494.5	101744.3	103063.7	107250.8	101979.3	101313.7	100060.9
107118.4	104177.3	102857.9	105293.5	107800.7	107809.7	103003.1	101009.9
104449.3	104980.9	107656.9	104355.7	106541.5	105836.3	100754.7	98243.89
103285.3	107080.7	104307.3	102086.4	104626.5	101537.1	105077.9	100075.7

單因數變異數分析

摘要				
組	個數	總和	平均	變異數
欄 1	30	3182097	106069.9	6655362
欄 2	30	3135477	104515.9	6771125
欄 3	30	3104845	103494.8	3888410
欄 4	30	3058586	101952.9	5743773

ANOVA						
變源	SS	自由度	MS	F	P-值	臨界值
組間	2.7E+08	3	89962231	15.6058	1.36E-08	2.682809
組內	6.69E+08	116	5764667			
總和	9.39E+08	119				

輕鏈與重鏈重組率實驗設計求解結果

0.5	0.7	0.8	0.5	0.5	0.7	0.8	0.5
0.5	0.4	0.2	0.8	0.5	0.4	0.2	0.8
102666.3	105115.9	103405.5	96787.31	105679.1	104949.7	104353.5	105239.5
107185.7	103814.9	103822.3	104377.5	108038.7	104071.1	102222.1	99547.89
107820.7	103253.3	104354.7	106486.7	105062.1	107377.5	105267.7	102508.9
106528.9	106218.9	104617.9	103227.9	104328.5	107010.5	105057.3	101485.9
106149.7	106085.9	109439.7	103211.9	105192.3	111267.4	102785.3	103652.1
104225.7	107060.1	104671.7	101944.5	103196.3	103863.9	104262.3	101764.7
106537.9	110046.7	102379.1	104111.5	104285.7	103776.9	105575.7	105850.3
103374.7	109058.7	108831.9	108025.3	100549.5	105891.3	107992.9	105088
103582.5	98768.89	105589.7	106798	108705.9	105494.9	103475.3	101065.9
99233.68	108398.1	102173.7	106645.9	104738.5	104698.5	104122.1	101603.3
100667.1	100700.1	102574.3	108005.5	106943.9	106352.3	105239.7	102189.5
106936.3	106174.9	102529.4	105325.9	109560.5	101130.7	103383.9	108370.1
103359.5	106821.7	107748.7	105600.3	105801.9	106754.7	104511.3	101987.3
104678.8	105780.1	107030.7	103044.1	106315.7	105026.5	106449.4	105447.4
111374.1	100903.5	106472.7	103331.7	106031.3	105959.1	107311.3	106435.3

單因數變異數分析

摘要				
組	個數	總和	平均	變異數
欄 1	30	3158752	105291.7	6954235
欄 2	30	3161827	105394.2	7385013
欄 3	30	3147652	104921.7	4008446
欄 4	30	3119161	103972	7067865

ANOVA						
變源	SS	自由度	MS	F	P-值	臨界值
組間	37777155	3	12592385	1.981839	0.120548	2.682809
組內	7.37E+08	116	6353890			
總和	7.75E+08	119				

迭代數實驗設計求解結果

100	300	500	700	900	100	300	500	700	900
104426.7	102601.1	108422.7	105323.9	97509.47	103676.7	103413.5	99779.68	100268.9	96902.15
100644.1	107875.7	102443.4	102505.9	106174.1	108196.7	104107.1	100808.9	99805.73	97572.1
106681.5	102906.5	100728.7	103618.3	103668.3	106733.9	103042.3	102458.1	109022.3	97728.1
103888.5	103623.9	103522.3	100759.9	102079.7	106804.9	104444.9	105481.9	99820.73	99808.26
106261.3	108759.7	106396.4	105903.5	95234.15	106170.3	103504.1	96815.15	100195.1	103638.5
104553.1	103608.7	101023.7	105294.9	106819.3	111299.3	100376.1	97789.31	100783.7	101572.7
106776.1	106706.3	99900.52	102587.8	102000.3	101082.7	103508.3	96309.31	100734.1	100132.5
106907.1	98144.1	102312.3	102499.1	95234.52	106085.7	103732.1	104609.1	103101.5	98022.31
107345.4	102825.7	101320.7	97887.73	102044.7	107135.7	104797.1	104908.1	100688.1	100073.5
106377.5	101151.3	103447.4	103888.1	95688.1	105597.7	104151.5	100773.3	103847.3	104258.9
102240.3	107206.1	105622.1	100239.9	98992.36	102174.5	100708.3	106171.1	97379.52	102914.4
106646.3	103042.9	99375.1	103152.9	97430.1	99678.94	103729.1	103298.7	102354.3	102904.3
108433.9	103944.5	102329.1	101972.7	101920.7	106160.9	101710.1	104815.3	102563.3	96609.68
100611.3	106021.6	102089.3	96151.47	96910.68	105472.1	104380.4	96741.31	102174.3	100820.1
107504.3	104934.9	97844.68	100752.9	102044.9	105077.9	100041.9	106858.4	97410.47	102285.3

單因數變異數分析

摘要	個數	總和	平均	變異數
欄 1	30	3160646	105354.9	6959293
欄 2	30	3109000	103633.3	5246271
欄 3	30	3064396	102146.5	10351338
欄 4	30	3052689	101756.3	7533734
欄 5	30	3008994	100299.8	10204661

ANOVA						
變源	SS	自由度	MS	F	P-值	臨界值
組間	4.46E+08	4	1.11E+08	13.82734	1.41E-09	2.434065
組內	1.17E+09	145	8059059			
總和	1.61E+09	149				

記憶區抗體數目取出比例實驗設計求解結果

1:1	1:3	4:1	1:1	1:3	4:1
99793.891	100171	104076.3	101417.891	102454.9	102505.3
97716.523	102611.5	103345.7	95776.102	98582.73	103281.1
104101.313	96753.15	99596.1	102642.938	101648.3	100013.1
100453.102	99667.1	102088.1	102212.938	99318.89	102275.9
100156.727	95742.94	97442.52	101072.68	101325.3	103216.4
108939.102	103258.9	104124.5	103683.313	103146.5	101942.9
100965.313	100851.3	100602.1	100883.891	96731.1	98988.1
96501.313	96746.89	98050.89	98318.891	98934.47	104010.3
98794.148	102542.3	102892.4	98042.313	99766.94	103120.3
99637.313	94688.47	101532.5	99834.727	101402.9	95713.31
101592.891	95209.89	102087.3	102182.359	101451.3	99191.52
100963.313	101308.9	98360.52	96780.68	96220.89	101787.7
100056.102	96862.73	102484.7	98669.891	99393.73	100911.9
98727.523	100556.1	98904.52	101530.102	97872.47	99866.73
98980.727	100558.7	103627.3	95682.102	103803.3	102359.3

單因數變異數分析

摘要	個數	總和	平均	變異數
組				
欄 1	30	3006110	100203.7	7367422.579
欄 2	30	2989584	99652.79	6765448.179
欄 3	30	3038399	101280	4862491.883

ANOVA						
變源	SS	自由度	MS	F	P-值	臨界值
組間	41096070	2	20548035	3.245218659	0.043723	3.101296
組內	5.51E+08	87	6331788			
總和	5.92E+08	89				

附錄四、禁忌搜尋法求解求解在港時間參數實驗設計

禁忌名單參數實驗設計求解結果

5	7	14	5	7	14
98287.26	105196.1	106871.9	99223.47	103399.7	106519.1
96984.52	100257.1	105694.9	99003.47	107069.7	103359
97655.47	101862.5	104575.5	98114.68	104148.5	106940.7
97881.94	105733.5	105733.7	95722.47	105966.7	107053.9
103327.7	102817.3	108228.7	101490.5	103141.3	105486.5
100290.9	102670.3	107867.3	96159.26	103573.1	107770.5
100326	104232.5	102740.5	97538.94	103700.7	105190.3
101884.9	108359.5	108062.7	100663.9	103700.5	104992.8
100870.3	102875.7	104666.9	101098.7	105885.9	105494.5
100601.3	105067.7	105205.5	97008.89	104020.7	107286.3
98986.52	108520.1	109191.9	97375.1	108689.5	105490.7
101393.3	102036.7	104064	96259.73	105584.4	104127.5
95407.52	104670.3	104927.7	98087.31	103753.9	102765.1
93069.73	107159.1	109238.9	101515.3	104782.5	106578.7
98896.1	107309.9	103886.3	102975.1	104451.5	106379.7

單因數變異數分析

摘要				
組	個數	總和	平均	變異數
欄 1	30	2968100	98936.68	5865197
欄 2	30	3140637	104687.9	4235363
欄 3	30	3176392	105879.7	3076991

ANOVA						
變源	SS	自由度	MS	F	P-值	臨界值
組間	8.27E+08	2	4.14E+08	94.14064	1.73E-22	3.101296
組內	3.82E+08	87	4392517			
總和	1.21E+09	89				

鄰域搜尋次數實驗設計求解結果

50	100	500	800	1000	50	100	500	800	1000
105796.7	105573.1	104545.3	104581.6	108086.9	105321.1	104989.5	103950.3	104946.3	101460.5
103444.7	105933.3	105518.9	106328.7	101770.3	103543.3	103969.9	103244.3	103510.7	102447.3
103240.7	103793.1	105238.5	107296.5	105502.5	105562.7	106582.1	103690.9	99230.15	105250.1
104379.1	107458.7	107669.9	103592.1	106368.9	106777.3	104766.9	106979.3	106374.7	103204.9
103707.9	102803.7	102396.9	105219.9	108331.1	107327.7	107796.3	105072.9	105878.4	104660
101757.7	103258.7	103826.5	105309.9	105186.1	107452.9	101822.9	105765.5	106715.1	102771.3
100379.9	105979.9	105935.3	102056.9	105282.3	102911.5	105434	103595.9	107140.5	105522.1
104850.1	104230.9	104298.3	102458.3	100771.9	107758.3	105863.1	106418.1	103709.3	104019.9
104740.3	107854.7	102382.5	105353.5	106076.5	106233.3	106559.9	107065.1	105284.7	106983.5
107984.5	103573.7	104052	104584.3	104404.5	105232.9	105876.5	100127.9	103701.3	102165.7
105539.3	103622.3	104499.1	102999	107049.3	105820.8	106955.1	102329.7	105211.1	104849.3
101614.5	106827.3	105815.1	102291.5	106340.3	105472.5	105860.7	108276.9	108332.7	105602.3
108463.7	103650.9	106710.5	105941.3	105608.1	102448.7	104801.1	102822.9	107419.3	103707.7
106263	104061.9	104966.5	102027.7	106690.9	108679.5	104026.7	106966.5	106127.5	102138.1
105323.5	106767.3	105510.7	104605.7	104702.5	103985.7	104827.5	101681.7	105965.7	106667.1

單因數變異數分析				
摘要				
組	個數	總和	平均	變異數
欄 1	30	3152014	105067.1	4380270
欄 2	30	3155522	105184.1	2399426
欄 3	30	3141354	104711.8	3669695
欄 4	30	3144194	104806.5	3903171
欄 5	30	3143622	104787.4	3940760

ANOVA						
變源	SS	自由度	MS	F	P-值	臨界值
組間	4944644	4	1236161	0.337872	0.852069	2.434065
組內	5.31E+08	145	3658664			
總和	5.35E+08	149				

附錄五、建議船席指派位置、順序與多目標隸屬函數值

<table><tr><td>17</td><td>24</td><td>36</td><td>20</td><td>22</td><td>3</td><td>34</td><td>29</td><td>44</td><td>0</td><td>0</td><td>2</td><td>46</td><td>13</td><td>0</td><td>0</td><td>0</td></tr><tr><td>45</td><td>41</td><td>27</td><td>25</td><td>0</td><td>1</td><td>12</td><td>23</td><td>15</td><td>9</td><td>48</td><td>0</td><td>38</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>39</td><td>16</td><td>30</td><td>28</td><td>0</td><td>0</td><td>47</td><td>42</td><td>0</td><td>0</td><td>21</td><td>0</td><td>0</td><td>11</td><td>0</td><td>0</td><td>0</td></tr><tr><td>6</td><td>33</td><td>32</td><td>5</td><td>18</td><td>7</td><td>43</td><td>31</td><td>50</td><td>0</td><td>0</td><td>0</td><td>0</td><td>37</td><td>0</td><td>0</td><td>0</td></tr><tr><td>14</td><td>4</td><td>8</td><td>49</td><td>35</td><td>19</td><td>10</td><td>0</td><td>0</td><td>26</td><td>0</td><td>0</td><td>40</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>																	17	24	36	20	22	3	34	29	44	0	0	2	46	13	0	0	0	45	41	27	25	0	1	12	23	15	9	48	0	38	0	0	0	0	39	16	30	28	0	0	47	42	0	0	21	0	0	11	0	0	0	6	33	32	5	18	7	43	31	50	0	0	0	0	37	0	0	0	14	4	8	49	35	19	10	0	0	26	0	0	40	0	0	0	0
17	24	36	20	22	3	34	29	44	0	0	2	46	13	0	0	0																																																																																					
45	41	27	25	0	1	12	23	15	9	48	0	38	0	0	0	0																																																																																					
39	16	30	28	0	0	47	42	0	0	21	0	0	11	0	0	0																																																																																					
6	33	32	5	18	7	43	31	50	0	0	0	0	37	0	0	0																																																																																					
14	4	8	49	35	19	10	0	0	26	0	0	40	0	0	0	0																																																																																					
迭代數		最大λ		迭代數		最大λ		迭代數		最大λ		迭代數		最大λ																																																																																							
1		0.733		31		0.918		61		0.918		91		0.918																																																																																							
2		0.733		32		0.918		62		0.918		92		0.918																																																																																							
3		0.766		33		0.918		63		0.918		93		0.918																																																																																							
4		0.779		34		0.918		64		0.918		94		0.918																																																																																							
5		0.79		35		0.918		65		0.918		95		0.918																																																																																							
6		0.79		36		0.918		66		0.918		96		0.918																																																																																							
7		0.79		37		0.918		67		0.918		97		0.918																																																																																							
8		0.79		38		0.918		68		0.918		98		0.918																																																																																							
9		0.793		39		0.918		69		0.918		99		0.918																																																																																							
10		0.794		40		0.918		70		0.918		100		0.918																																																																																							
11		0.832		41		0.918		71		0.918		101		0.918																																																																																							
12		0.832		42		0.918		72		0.918		102		0.918																																																																																							
13		0.832		43		0.918		73		0.918		103		0.918																																																																																							
14		0.84		44		0.918		74		0.918		104		0.918																																																																																							
15		0.867		45		0.918		75		0.918		105		0.918																																																																																							
16		0.867		46		0.918		76		0.918		106		0.918																																																																																							
17		0.867		47		0.918		77		0.918		107		0.918																																																																																							
18		0.891		48		0.918		78		0.918		108		0.918																																																																																							
19		0.891		49		0.918		79		0.918		109		0.918																																																																																							
20		0.905		50		0.918		80		0.918		110		0.918																																																																																							
21		0.905		51		0.918		81		0.918		111		0.918																																																																																							
22		0.905		52		0.918		82		0.918		112		0.918																																																																																							
23		0.918		53		0.918		83		0.918		113		0.918																																																																																							
24		0.918		54		0.918		84		0.918		114		0.918																																																																																							
25		0.918		55		0.918		85		0.918		115		0.918																																																																																							
26		0.918		56		0.918		86		0.918		116		0.918																																																																																							
27		0.918		57		0.918		87		0.918		117		0.918																																																																																							
28		0.918		58		0.918		88		0.918		118		0.918																																																																																							
29		0.918		59		0.918		89		0.918		119		0.918																																																																																							
30		0.918		60		0.918		90		0.918		120		0.918																																																																																							

迭代數	最大 λ	迭代數	最大 λ	迭代數	最大 λ	迭代數	最大 λ
121	0.918	151	0.939	181	0.939	211	0.939
122	0.918	152	0.939	182	0.939	212	0.939
123	0.918	153	0.939	183	0.939	213	0.939
124	0.918	154	0.939	184	0.939	214	0.939
125	0.918	155	0.939	185	0.939	215	0.939
126	0.918	156	0.939	186	0.939	216	0.939
127	0.918	157	0.939	187	0.939	217	0.939
128	0.918	158	0.939	188	0.939	218	0.939
129	0.918	159	0.939	189	0.939	219	0.939
130	0.918	160	0.939	190	0.939	220	0.939
131	0.918	161	0.939	191	0.939	221	0.939
132	0.918	162	0.939	192	0.939	222	0.939
133	0.918	163	0.939	193	0.939	223	0.939
134	0.918	164	0.939	194	0.939	224	0.939
135	0.918	165	0.939	195	0.939	225	0.939
136	0.918	166	0.939	196	0.939	226	0.939
137	0.918	167	0.939	197	0.939	227	0.939
138	0.918	168	0.939	198	0.939	228	0.939
139	0.918	169	0.939	199	0.939	229	0.939
140	0.918	170	0.939	200	0.939	230	0.939
141	0.918	171	0.939	201	0.939	231	0.939
142	0.918	172	0.939	202	0.939	232	0.939
143	0.918	173	0.939	203	0.939	233	0.939
144	0.918	174	0.939	204	0.939	234	0.939
145	0.918	175	0.939	205	0.939	235	0.939
146	0.939	176	0.939	206	0.939	236	0.939
147	0.939	177	0.939	207	0.939	237	0.939
148	0.939	178	0.939	208	0.939	238	0.939
149	0.939	179	0.939	209	0.939	239	0.939
150	0.939	180	0.939	210	0.939	240	0.939

迭代數	最大 λ	迭代數	最大 λ	迭代數	最大 λ	迭代數	最大 λ
241	0.939	271	0.939	301	0.939	331	0.939
242	0.939	272	0.939	302	0.939	332	0.939
243	0.939	273	0.939	303	0.939	333	0.939
244	0.939	274	0.939	304	0.939	334	0.939
245	0.939	275	0.939	305	0.939	335	0.939
246	0.939	276	0.939	306	0.939	336	0.939
247	0.939	277	0.939	307	0.939	337	0.939
248	0.939	278	0.939	308	0.939	338	0.939
249	0.939	279	0.939	309	0.939	339	0.939
250	0.939	280	0.939	310	0.939	340	0.939
251	0.939	281	0.939	311	0.939	341	0.939
252	0.939	282	0.939	312	0.939	342	0.939
253	0.939	283	0.939	313	0.939	343	0.939
254	0.939	284	0.939	314	0.939	344	0.939
255	0.939	285	0.939	315	0.939	345	0.939
256	0.939	286	0.939	316	0.939	346	0.939
257	0.939	287	0.939	317	0.939	347	0.939
258	0.939	288	0.939	318	0.939	348	0.939
259	0.939	289	0.939	319	0.939	349	0.939
260	0.939	290	0.939	320	0.939	350	0.939
261	0.939	291	0.939	321	0.939	351	0.939
262	0.939	292	0.939	322	0.939	352	0.939
263	0.939	293	0.939	323	0.939	353	0.939
264	0.939	294	0.939	324	0.939	354	0.939
265	0.939	295	0.939	325	0.939	355	0.939
266	0.939	296	0.939	326	0.939	356	0.939
267	0.939	297	0.939	327	0.939	357	0.939
268	0.939	298	0.939	328	0.939	358	0.939
269	0.939	299	0.939	329	0.939	359	0.939
270	0.939	300	0.939	330	0.939	360	0.939

迭代數	最大 λ	迭代數	最大 λ	迭代數	最大 λ	迭代數	最大 λ
361	0.939	391	0.939	421	0.939	451	0.981
362	0.939	392	0.939	422	0.939	452	0.981
363	0.939	393	0.939	423	0.939	453	0.981
364	0.939	394	0.939	424	0.939	454	0.981
365	0.939	395	0.939	425	0.939	455	0.981
366	0.939	396	0.939	426	0.939	456	0.981
367	0.939	397	0.939	427	0.939	457	0.981
368	0.939	398	0.939	428	0.939	458	0.981
369	0.939	399	0.939	429	0.939	459	0.981
370	0.939	400	0.939	430	0.939	460	0.981
371	0.939	401	0.939	431	0.939	461	0.981
372	0.939	402	0.939	432	0.939	462	0.981
373	0.939	403	0.939	433	0.939	463	0.981
374	0.939	404	0.939	434	0.939	464	0.981
375	0.939	405	0.939	435	0.939	465	0.981
376	0.939	406	0.939	436	0.939	466	0.981
377	0.939	407	0.939	437	0.963	467	0.981
378	0.939	408	0.939	438	0.963	468	0.981
379	0.939	409	0.939	439	0.963	469	0.981
380	0.939	410	0.939	440	0.963	470	0.981
381	0.939	411	0.939	441	0.963	471	0.981
382	0.939	412	0.939	442	0.963	472	0.981
383	0.939	413	0.939	443	0.963	473	0.981
384	0.939	414	0.939	444	0.981	474	0.981
385	0.939	415	0.939	445	0.981	475	0.981
386	0.939	416	0.939	446	0.981	476	0.981
387	0.939	417	0.939	447	0.981	477	0.981
388	0.939	418	0.939	448	0.981	478	0.981
389	0.939	419	0.939	449	0.981	479	0.981
390	0.939	420	0.939	450	0.981	480	0.981

附錄六、船舶進出港流程與基隆港船席指派作業

一、船舶進出港流程

港埠管理當局為維持船舶進出港船舶、人、貨之安全，且使港埠有限資源能充分運用，及維護港埠本身之安全，船舶進出港均採一定之流程予以一定之流程，予以管理，使船舶能安全在港內裝卸，進出港均能井然有序，達到船舶均能有效率地使用港埠資源。

船舶進出港作業大致分為「船舶進、出港預報」、「船舶進出港檢查」、「船席調派」、「靠泊碼頭」、「裝卸作業」等。各項作業及相關作業條件分別敘述如下：

1. 船舶進港預報

船舶進港，應於 24 小時前得由船長以傳真或電報向港務當局報告或是由船公司(代理行)轉報預訂到達時間(ETA)、吃水深度、貨物種類、旅客人數，以便指定船席，並由船公司(代理行)依照相關法令規定，向相關機關辦理申報進港手續，其預報時間進港時間與實際進港時間，前後均不得超過 48 小時，逾時應再申辦，如因緊急事故申請進港，得依港口管制辦法之規定，辦理進港手續。

2. 船舶進港

船公司(代理行)向商港管理機關(港務局航政組)申請，經審核簽證後，由申報人分送各港區檢查處、港警所外事組，進港船舶並按照港務局指定船席及許可信號通知，同時懸掛中華民國國旗、船籍國旗及船舶電台呼號旗。在船舶到港時，由引水人引領進港，靠泊指定之碼頭，屆時，港務局將派拖船及纜工協助完成靠泊作業。

3. 船舶進出港檢查

船舶檢查，包括檢疫、旅客、貨物及行李等項，由檢疫機關、海關，及證照檢驗與其他有關機關分別辦理。一般進出港檢查，應於啟碇或停泊後二小時檢查完畢。

4. 船舶出港預報

船舶出港應該於 12 小時前，由船公司(代理行)向港務局預報，並依照相關法令，向有關相關機關申報出港手續，其預報時間出港時間與實際出港時間，前後均不得超過 48 小時，逾時應再申辦，如因緊急事故申請進港，得依港口管制辦法之規定，辦理出港手續。

5. 船舶出港

船公司(代理行)依商港主管機關規定，據實填妥船舶出港預報表，載明出港時間、吃水、船長、貨物種類、數量、旅客人數、及到達次一港與目的港名稱，連同船員名單、手冊暨各項航政文書，向航政主管申報，並將出港報告單分送港區檢查處、港警所外事組及港務組後，始准出港。

6. 船席調派

船席調派主要是在充分運用船席，使船席的空閒時間減至最少，並使泊於船席之船舶皆能迅速的完成貨物裝卸工作，而不發生中斷之閒置狀況。以下為決定船席的因素：

(1) 船舶長度

配合船席長度，同時保持船與船之間之相當距離，靠泊時才不致影響本身的安全，甚至造成鄰近船隻及本身船舶之安全。

(2) 船舶吃水與碼頭深度之配合：

各碼頭的水深，深淺不一，需慎重處理來港船舶的吃水，配合碼頭水深，以避免發生無法靠泊或擱淺等海事事故。

(3) 內陸運輸：

如基隆港西岸有鐵路連接內陸，東岸碼頭則無此設施，故須利用鐵路內陸運輸之船舶在船席調派時則須事先考慮其需要。

(4) 倉儲能量：

船席決定前，則需要考慮倉棧進儲作業的需要，才不會造成倉位無法配合作業，影響裝卸率。

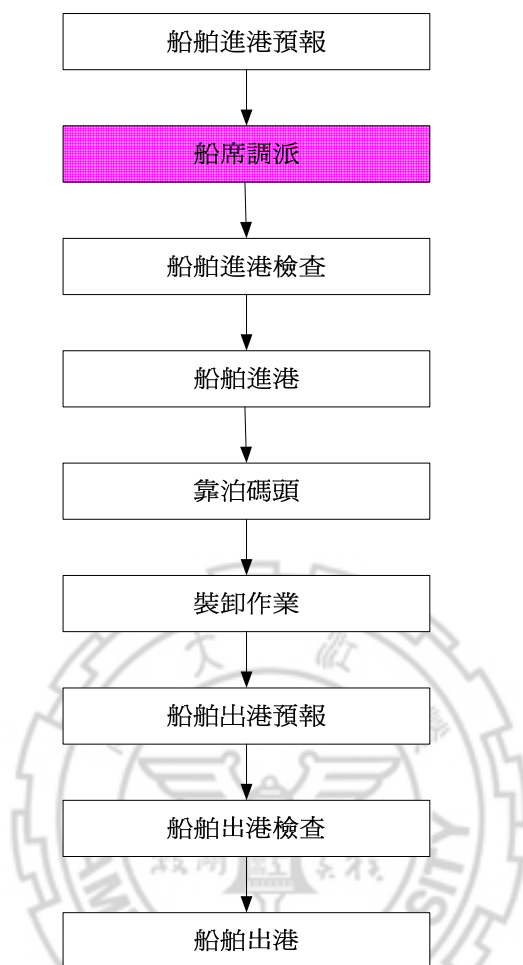
(5) 裝卸設備：

來港作業船隻，須配合船上裝卸設備特性及裝卸機具作業能力，如貨櫃碼頭貨櫃起重機可用狀況，使調配後船舶不會造成裝卸上的困擾。

(6) 天候因素：

如某些船席受氣候影響無法裝卸，須改泊其他船席始能作業。

船舶進出港流程如圖一所示：



圖一、船舶進出港流程圖

資料來源：王丘明，2003

二、船席指派作業(以基隆港為例)

基隆港現行船席調派是以會議的方式進行，參與會議的小組有船席調派小組、航商與代理公司。

船席調派小組的組成由港務長、棧埠處處長、港務組長、船交中心主任、船舶所所長、繫船課長、倉儲課長、搬運課長八人（或由其代理人）及繫船課作業人員組成，港務長為召集人，港務組長為協助召集人，繫船課長為執行秘書。

船席調配小組除星期例假外每日定期舉行會議一次，公開決定翌日（或以後）之船席分配，除特殊情況外，參加人員為召集人、或協助召集人、執行秘書及港務棧埠等有關單位派員協調船席調配事宜。

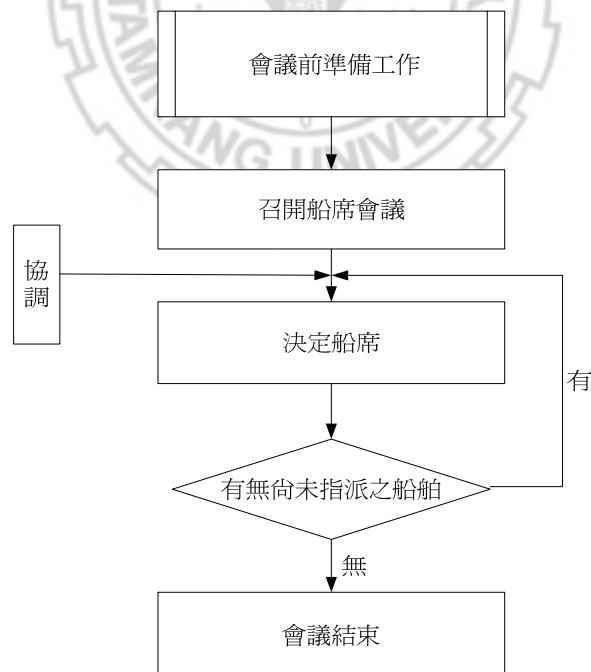
每日上午九點鐘由繫船課課長主持船席會議，是基隆港務局行之多年的制度，會議的主要目的在協調指泊各預定到港之船舶，會議的進程式大致如下：

1. 首先在大黑板上展示後派船舶名單及設施使用現況(主要為船席、裝卸機具)。

2. 由繫船課課長依後派船舶優先順位唱名，被唱名的船商代表考慮其船舶特性及碼頭設施使用現況，隨即決定欲靠泊之船席，並由繫船課人員當場填寫該船舶於大黑板上欲泊船席欄位，如此便完成一艘船舶的船席指泊。
3. 繫船課課長繼續唱名，直到所有後派船舶均完成指泊為止。順位較後的船舶，會觀察整體指泊狀況，若無條件較好之船席可供靠泊，航商通常會提出待定，等隔日船席會議提出重排。
4. 指泊過程中，若順位較後的船舶與順位較前之船舶欲靠泊之船席發生衝突時，順位較後者應於順位較前者選定船席之際，提出協商要求。協商後經一方調整欲靠泊之船席，並經確定無異議後，順位較前之船舶便確定船席。若協商時間過長，繫船課課長通常會視情況與以保留，進而先行決定其他類別船舶之船席，最後才處理有爭議之船席。正常情況下，船席會議在十點鐘左右結束。

會議結束後，繫船課人員依船席會議指派結果，立即更新繫船課辦公室調度看板上之船舶預訂到港時間及預定離港時間，並更新「基隆港港灣圖」，以便充分掌握船舶動態。

每日值班人員如遇有前一艘貨櫃船延開(含)4 小時以上、提前開航 4 小時以上、其他貨櫃船延遲抵港、橋式機故障等原因發生以致船席產生異動，而影響船席調度作業時，值勤人員立即依「基隆港船舶繫泊作業須知」及相關法令，於船席異動原因發生時或於當日下午 4 點舉行另一次小型「船席調派會議」，通知今日與船席異動有關之航商與會，並依 VHF(超短波無線電話)報到時間之先後、指泊應衡量考慮之因素，重新安排船席。圖二為船席調派之流程。



圖二 船席調派之流程

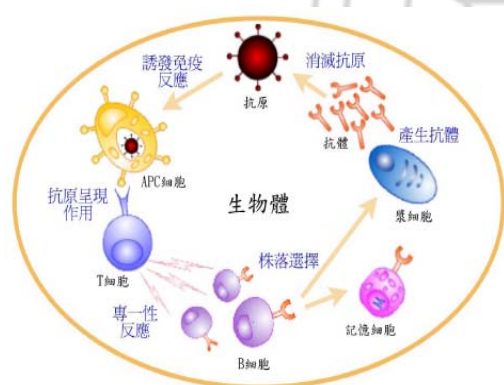
附錄七、生物免疫系統介紹

一、免疫反應(Immune Response)

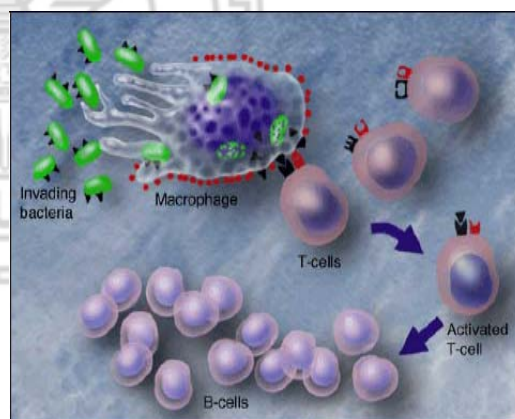
在我們的生活環境中有許許多多各樣的感染性微生物，例如病毒、細菌、原蟲等等，其中任何一種微生物都可引起病理性的傷害，甚至會殺死宿主。在健康人的身上，絕大多數的感染均屬於短暫性，極少留下永久性的傷害，這是因為人體內的免疫系統(immune system)能夠有效地對抗感染性病原體(infectious agents)。

當人體遇到某些病原體的刺激或感染時，體內會自動產生保護性的反應能力，這種反應能力稱為『免疫反應』(immune-response)。這些入侵人體的病原體稱為抗原(antigen)，而保護人體的物質則稱為抗體(antibody)，又稱為免疫球蛋白(immunoglobulin) (Roitt et al.,1998)。

生物免疫反應是由多種細胞及其所分泌的可溶性分子共同參與。雖然白血球是所有免疫反應的主角，但組織中其他細胞也參與反應且傳訊給淋巴細胞，並對T 淋巴細胞與巨噬細胞(macrophage)分泌的細胞激素產生反應。下圖為生物免疫反應架構圖，當抗原剛進入生物體時，會先與毒殺性 T 細胞或吞噬細胞結合，而吞噬細胞有兩種，一種為專門清除特殊抗原的巨噬細胞，另一種為抗原呈獻細胞。抗原呈獻細胞主要的作用在攝取抗原，並在處理完後將此抗原呈獻給輔助型 T 細胞及 B 細胞，藉以加速免疫反應。



圖一 生物免疫反應架構圖



圖二 免疫反應示意圖

免疫系統大致包含兩道防線：

第一道防線：是非特異的天然免疫反應(innate immune response)，包含皮膚、黏液、巨噬細胞(macrophage)等天然屏障。當一個感染性的病毒(抗原)入侵到人體時，體內先天性反應的巨噬細胞或專業的抗原呈現細胞(如樹狀細胞等)便會將病毒(抗原)吞噬，如圖二所示。

第二道防線：適應性免疫反應(specific immune response)，包含細胞性(T細胞)的免疫和體液性(B細胞)的免疫兩大部份，通常他們具有記憶性，是透過淋巴細胞來發揮免疫作用。

承上之免疫反應第一道防線，巨噬細胞將病毒(抗原)吞噬之後，它們會經過酵素的作用將抗原切成碎片，將這些抗原表現在細胞膜外。

這些帶有特殊抗原切片的吞噬細胞會回到淋巴節供 T 細胞與 B 細胞認。一旦 T 細胞與 B 細胞辨認出它們所認識的抗原後便會進一步的被活化並執行分裂及增殖作用，進而產生大量對特定抗原具有特異性的細胞殺性 T 細胞(T-cytotoxic)及促使 B 細胞分泌專一性抗體的漿細胞來清除感染源，此種需要抗原刺激才會發生對特定抗原才有的反應之免疫過程稱之為適應性免疫反應。天然免疫反應與適應性免疫反應的比較如下表一所示。

表一 天然免疫反應與適應性免疫反應之比較

天然免疫反應 (innate immune response)	適應性免疫反應 (specific immune response)
1. 提供體內第一道防線 2. 主要利用吞噬細胞對抗 3. 不具抗原專一性 4. 不具記憶性	1. 在第一道防線被突破後所進行的反應 2. 分為細胞性(T細胞)的免疫與體液性(B細胞)的免疫 3. 具有抗原專一性(specific) 4. 具有記憶性(memory)

二、抗原(antigen, Ag)

簡單地說凡是能夠引起免疫反應的物質便稱為抗原，在抗原表面有抗原決定部位(epitope)為抗體用來辨認的地方，藉由它可與特定的抗體相結合，在抗原表面有抗原決定部位(epitope)為抗體所辨認的地方，藉由它可與特定的抗體相結合。

三、抗體(antibody, Ab)

B 細胞受抗原活化後成為漿細胞，漿細胞會分泌出抗體，也就是所謂『免疫球蛋白 (Immunoglobulin, Ig)』。它是一種 Y 形狀的分子。抗體結構依照分子量不同可分為輕鏈(light chain)與重鏈(heavy chain)兩部分，輕鏈主要由分子量較小的分子鍊所組成，而重鏈主要是由分子量較大的分子鍊所組成。

又依照作用不同又可分為恆定區(constant region)與變異區(variable region)兩部分，而與抗原結合的就是抗體的變異區。一般而言，一個抗體僅能與一個抗原發生專一性結合。而免疫球蛋白基因 V-J 和 V-D-J 的隨機重組及基因重排過程中連接方式的不同是造成抗體具有變異性的主要原因。

四、 T 細胞與 B 細胞

免疫系統中的淋巴球(lymphocyte)是主要的免疫細胞，一旦免疫系統辨識出有抗原入侵，淋巴將會產生大量的抗體(antibody)以消滅抗原。一般淋巴球可以分為 B 淋巴球與 T 淋巴球，即所謂的 B 細胞、T 細胞。

1. T 淋巴細胞

(1)有一群細胞可以幫助 B 細胞分裂、分化及製造抗體。

(2)另一群 T 細胞則與單核吞噬細胞交互作用並幫助其消滅病原。(1)與(2)這兩群細胞稱為輔助型 T 細胞(T-helper cells, T_H)。

(3)第三群 T 細胞負責破壞感染病毒或其他胞內病原體的宿主細胞，此過程稱細胞毒殺作用(cytotoxicity)，而此類細胞稱毒殺性 T 細胞(T-cytotoxic cells, T_c)。

2. B 淋巴細胞

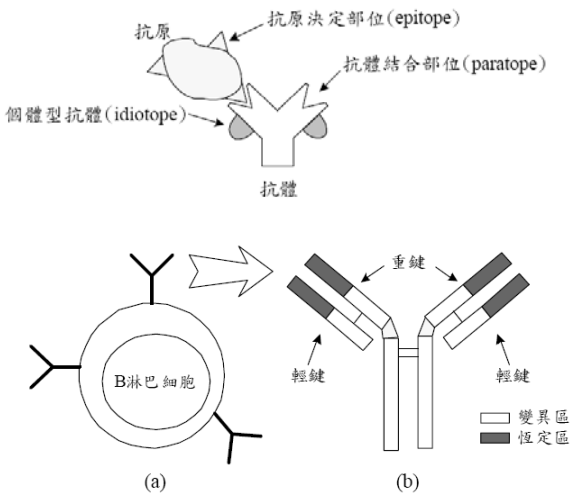
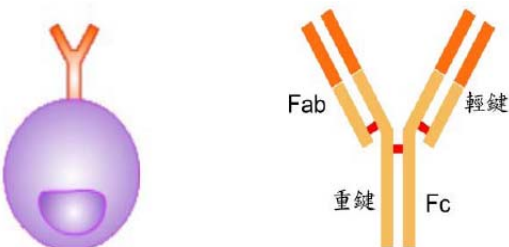
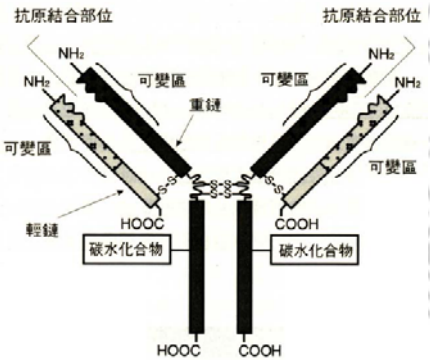
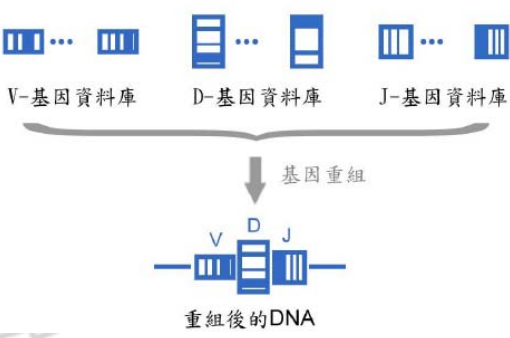
B 淋巴細胞(B 細胞)職責是生產抗體(antibody)。當病毒、細菌、不同種的血液、移植的器官等等出現時，B 細胞便迅速作出回應，製造抗體，擊退不速之客，其中的不速之客指的就是抗原(antigen)，也就是病原體表面的分子或病原體所產生的毒素。

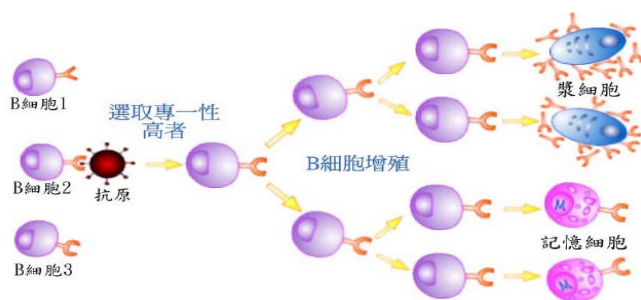
五、 株落選則與細胞記憶

人體可以製造出上千萬種抗體，每一種特定設計對付特定的敵人，就好像鑰匙與鎖一樣。B 細胞在離開骨髓前就已經在其細胞膜上生成具有各式各樣不同的受器(receptor)，這些細胞膜上的接受器是用來辨認特定抗原並與抗原做結合，當受器遇到其專一性的抗原時，便會被活化而形成分泌抗體的漿細胞與記憶性的 B 淋巴球，同時開始繁殖，此過程稱之為株落選擇(clonal selection)。

漿細胞製造大量可溶性受器分子分泌到細胞外，稱為抗體，而抗體是一種能專一辨認特定目標(抗原)的分子。這些抗體分子為大型醣蛋白，存於血液與組織液中。抗體有出色的記憶力和識別力，當同樣的敵人再次出現時，由於這些抗體與原先受器分子有相同的結構，因此可與那些活化 B 細胞的抗原相結合。第一次戰役後保留在血液中的老將會帶領著新兵，以加倍快速和準確的進攻，使敵人措手不及，無招架之力。所以常說曾經患過某種疾病的人，對它有一定的抵抗能力了。上述免疫系統相關專有名詞整理如表二所示。

表二 免疫系統相關名詞示意對照表

 <p>Diagram illustrating the interaction between an antigen and an antibody. The antigen is shown with an 抗原決定部位 (epitope). The antibody is shown with an 抗體結合部位 (paratope). The 個體型抗體 (idiotope) is also indicated. Below, a B lymphocyte is shown with its B淋巴細胞 structure. The antibody structure is detailed with 重鏈 (heavy chain) and 輕鏈 (light chain), and the 變異區 (variable region) and 恆定區 (constant region) are highlighted.</p>	 <p>Diagram illustrating the structure of an antibody. The antibody is shown as a Y-shaped molecule. The Fab (antigen-binding fragment) and Fc (crystallizable fragment) regions are labeled. The 重鏈 (heavy chain) and 輕鏈 (light chain) are also indicated.</p>
<p>抗體與抗原結合示意圖</p>	<p>B 淋巴細胞示意圖</p>
 <p>Detailed diagram of an antibody structure. The 抗原結合部位 (antigen-binding site) is shown. The 可變區 (variable region) and 重鏈 (heavy chain) are labeled. The 輕鏈 (light chain) and 碳水化合物 (carbohydrate) are also indicated. The HOOC (carboxyl) and NH₂ (amino) groups are shown at the ends of the chains.</p>	 <p>Diagram illustrating the gene rearrangement process in a B lymphocyte. The V-基因資料庫 (V-gene database), D-基因資料庫 (D-gene database), and J-基因資料庫 (J-gene database) are shown. The 基因重組 (gene rearrangement) process is indicated, leading to the 重組後的DNA (rearranged DNA).</p>
<p>抗體Y型結構示意圖</p>	<p>免疫球蛋白基因重組之示意圖</p>



細胞株落選擇(clonal selection)示意圖

資料來源：[古自強，2004]、[張慶彬，2004]

