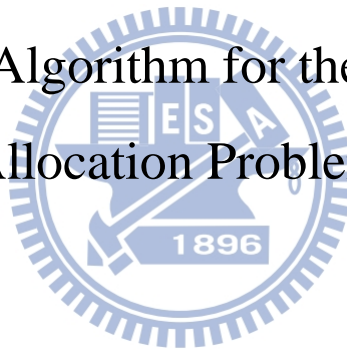


國立交通大學
運輸科技與管理學系碩士班

碩士論文

運用螞蟻演算法求解動態船席指派問題

An Ant System Algorithm for the Dynamic Berth
Allocation Problem



研究生：陳儀安

指導教授：黃寬丞 副教授

中華民國 一 百 零 一 年 七 月

運用螞蟻演算法求解動態船席指派問題

An Ant System Algorithm for the Dynamic Berth Allocation Problem

研究生：陳儀安

Student : Yi-An Chen

指導教授：黃寬丞

Advisor : Kuan-Cheng Huang

國立交通大學

運輸科技暨管理學系

碩士論文

A Thesis

Submitted to Department of Transportation Technology and Management

College of Management

National Chiao Tung University

In partial Fulfillment of the Requirements

for the Degree of

Master

In

Transportation Technology and Management

July 2012

Hsinchu, Taiwan, Republic of China

中華民國 一百零一年七月

運用螞蟻演算法求解動態船席指派問題

學生：陳儀安

指導老師：黃寬丞 副教授

國立交通大學運輸科技與管理學系

摘要

本研究之核心為船席指派問題(Berth Allocation Problem, BAP)，旨在給予船舶相應的指派船席。我們可事先得到船舶抵達時間的動態資訊，動態 BAP 的目標式為最小化總服務時間，隱含了所有船舶的等待時間以及處理時間。另外，船舶的處理時間與服務船席具相關性，進而會影響船席的指派決策。由於 BAP 為一排序問題，而螞蟻演算法(Ant Colony Optimization, ACO)在排序問題上的搜尋能力以及求解過程相較於其他巨集啟發式演算法更有優勢。因此，本研究採用螞蟻演算法作為求解方法。首先，透過螞蟻演算法產生船舶的指派順序，再藉由啟發式指派法則，根據船舶的抵達時間以及與船席相關的處理時間，進一步將船舶指派到相應船席進行服務。由數值測試可知，啟發式演算法與文獻中採用拉式鬆弛法之研究相較，能有效提升求解品質。

關鍵字：船席指派問題、螞蟻演算法

An Ant System Algorithm for the Dynamic Berth Allocation Problem

Student : Yi-An Chen

Advisor : Kuan-Cheng Huang

Department of Transportation Technology and Management
National Chiao Tung University

Abstract

The focus of this study is the Berth Allocation Problem (BAP), which determines the assignment of the berths to the calling ships. Given the dynamic information of the ship arrival times, the objective of the dynamic BAP is to minimize the total service times, defined as the sum of the waiting times and handling times, for all calling ships. In particular, the handling time is assumed to be berth-dependent and thus affected by the berth assignment decision. Owing to the nature of the sequencing decision problem associated with the BAP, this study chooses the Ant Colony Optimization (ACO), which has some inherited advantages over other meta-heuristics due to its sequential framework for the searching process and the solution building procedure. This study designs an ant-based algorithm to generate the ship assignment sequence, by which a greedy heuristic assigns a berth to a ship and determines the berthing window by considering its arrival time and the berth-dependent handling time. In the numerical experiment, the developed algorithm is compared with a solution algorithm based on Lagrangian Relaxation in the literature. It is found that the developed ant-based algorithm is promising with respect to the solution quality for the dynamic BAP.

Keywords: Berth Allocation Problem, Ant Colony Optimization

誌謝

回首交通大學為期兩年的研究生涯，深深覺得要感謝的人相當多，多虧你(妳)們的協助與叮嚀，才能讓我如期完成碩士學位，永遠對你(妳)們心存感謝。

從起初，偌大的校園帶來的疏離感，到如今對於交通大學有著濃厚的歸屬感，且離情依依，謝謝交通大學能給予我們一個優秀的學術環境不斷學習，讓我感受到學校對於學生的用心與照顧。也感謝運管大家庭的幫助，教授群們總是不遺餘力地傾囊教授，除了課程上的指導，也給予我們許多討論的機會，彼此激盪腦力，能對於知識的理解更為深入。

首先，我要非常感謝我的指導老師-黃寬丞老師，由於我的大學科系屬於財金領域，跨入運輸物流的世界，非常幸運有老師不耐其煩地相濡以沫，給了我許多有別於過去看待事情的思考面向。從老師身上，學習面對問題，該用什麼樣的角度做切入。另外，也謝謝老師在研究之路上的鼓勵與打氣，每每在迷惘的時候能讓學生找到方向繼續前進。也給了我許多機會擔任助教工作，協助同學，並有機會多和外籍生接觸。更重要的是，我在老師身上學習到面對壓力的正確心態，以積極取代抱怨，讓自己常保正面思考，才是處理問題的不二法門。

我也要感謝研究室的同學們的陪伴，謝謝婉葶學姊、家誼學姊、筱薇學姊的幫助，指引我在研究與課業上不斷前進。謝謝季佑、逸銘的相互打氣，才能在苦悶的日子得以宣洩壓力，也要謝謝彥蓉、小季和其他可愛的同學們，要是沒有你們的笑聲，很難想像日子會多麼乏味啊~另外，我要感謝薪軒，在我程式有疑惑的時候，總會耐心地與我討論，探討邏輯是否足夠縝密。此外，我還要謝謝均宜，不斷常常給我意見，也給我打氣，我才得以更有信心地迎向嶄新的每一天。

最後，我要感謝我的家人，你們就是我的避風港，能讓我無後顧之憂地好好讀書，我也會牢記你們的叮嚀。未來，就要換我成為你們的後盾了!!

陳儀安 謹誌於

國立交通大學運輸科技暨管理學系 KC lab

中華民國 101 年 07 月 04 日

目錄

中文摘要	i
英文摘要	ii
誌謝	iii
目錄	iv
表目錄	vi
圖目錄	vii
第一章 緒論	1
1.1 研究背景與動機	1
1.2 研究範圍與目的	1
1.3 研究方法與流程	4
1.3.1 研究方法	4
1.3.2 研究流程	4
第二章 文獻回顧	6
2.1 動態且離散型船席指派問題	6
2.1.1 動態且離散型船席指派問題之文獻回顧	6
2.1.2 動態且離散型船席指派問題之數學模式	8
2.2 螞蟻演算法之概述	11
2.2.1 螞蟻演算法簡介	11
2.2.2 螞蟻演算法在船席指派問題之應用	11
第三章 求解演算法	13
3.1 螞蟻演算法之一般求解過程	13
3.2 求解演算法之設計	15
3.2.1 以螞蟻演算法產生指派順序	15
3.2.2 由指派順序進行船席指派	16
3.2.3 演算法流程圖與總結	18

第四章 數值測試·····	21
4.1 測試題目設計·····	21
4.2 螞蟻演算法迭代效果分析·····	22
4.3 演算法數值測試結果·····	24
第五章 結論與建議·····	28
參考文獻·····	29
簡歷·····	32



表目錄

表 4-1 數值測試 - 問題 1(25 艘船)的抵達時間資訊	21
表 4-2 數值測試 - 問題 1(25 艘船)的處理時間資訊	22
表 4-3 數值測試 - 螞蟻演算法之參數值	23
表 4-4 數值測試 - 問題 1(25 艘船)的模擬結果	23
表 4-5 數值測試 - 本研究(GAR1 與 GAR2)之數值比較	25
表 4-6 數值測試 - 本研究(GAR2)與 Imai(SUBG)之數值比較	26
表 4-7 數值測試 - 本研究(GAR1、GAR2)之求解效能比較	27



圖目錄

圖 1-1	動態且離散型船席指派問題之示意圖	2
圖 1-2	船席運作時間軸	2
圖 1-3	動態且連續型船席指派問題之示意圖	3
圖 1-4	研究流程圖	5圖
2-1	BAP目標式示意圖	9
圖 2-2	船舶指派順序到船席之示意圖	10
圖 3-1	求解演算法之簡要示意圖	13
圖 3-2	螞蟻演算法 - 節點選取示意圖	15
圖 3-3	GAR1 示意圖	17
圖 3-4	GAR2 示意圖	17
圖 3-5	船席指派問題之流程示意圖	19
圖 4-1	船舶處理時間之線性關係圖	22
圖 4-2	數值測試 - 問題 1(25 艘船)的模擬分析圖	24



第一章 緒論

1.1 研究背景與動機

綜觀全球的國際貿易活動，可以發現由於海洋占地球的覆蓋面積高達 70%，因而先前的國際貿易往往皆以海洋運輸為大宗。即便在幾十年前航空運輸的問世，其一舉縮短了相當幅度的運輸時間，但受限於空運載量少、運費高的要件，往往只有保值期受限或高單價的貨物，才會選擇使用空運當作國際貿易的運輸方式。因此，遑論過去、現在或是將來，海運在國際貿易的地位仍是不可撼動的。進而分析海洋運輸的領域，船舶視為運輸工具，港埠則是起訖點，同樣也是裝卸貨物的場域，對於國際貿易活動為主的國家來說，海洋運輸必然扮演著不可或缺的角色，要能有出色的經濟發展，其港口的競爭力自然成為不得不關注的焦點。

儘管全球的總體經濟有所起伏，長遠來說，海運的運量，尤其是在貨櫃數量上的走勢仍是持續向上攀升，全世界的港口往往都需解決船隻停泊的擁擠問題。此外，面臨油價高漲的趨勢，對於海運成本更是錙銖必較，如何控管整體海運成本自然是為相當重要的議題。以需求面來說，航運業者做航線規畫時，必然希望自身船隻都具有優先使用權，能在抵達港口時就接受服務；針對供給面而言，港口的營運者卻又面對資源有限的問題。因此，在有限的資源下，港口營運者如何進行船隻靠港的服務作業，遂成為一重要課題。

貨櫃碼頭的作業可區分為三種功能系統，一為碼頭船席的分配排班、二為從船舶卸下貨櫃、三則為把貨櫃裝上船舶。其中，第一項碼頭船席的分配排班乃為整體海運的成本結構中最為重要的環節。不僅須兼顧船舶在海上的等候服務的等待時間，也要評估船舶在不同船席服務時，會有不同的處理時間。其原因來自於船公司與相對貨櫃場的有合作契約，解決搬運過程的處理時間會有所差異。然而，船席為一重要且稀少的空間資源，建造費用往往也所費不貲，不可能一味增加船席供給來提供船商服務，有效地加以運用船席資源才是較為可行的辦法。有別於過去貨櫃港埠經營的人工作業，目前較為先進的貨櫃港埠經營模式多採電腦輔助作業，以解決如此大型且繁複的船席指派問題。因此，在面對眾多需要靠港的船舶且船席有限的情況下，如何提供「船舶最適化的調派服務資訊」，使得整體的港埠營運更有效率乃為本研究所欲探討的核心課題。

1.2 研究範圍與目的

一般將上述把貨櫃碼頭分配給船舶的問題稱為「船席指派問題」(Berth Allocation Problem, BAP)。它為一種組合性的最適化問題，希望能在有限的船席

資源下，讓需要接受服務的船舶指派的越為緊密，所佔用的時間越短越好。船隻停泊的分派可用圖 1-1 的時間和空間關係圖來表示。圖 1-1 的橫軸為時間軸，縱軸則為碼頭長度。碼頭可以被切割成數個船席，而每一個船席則具有相同的長度，此種 BAP 型態可以被視為是一個「離散型」的問題，不同船席之間的指派服務，彼此獨立，並不會直接影響到其他船席的服務品質與行政效能。而每一個船席，同時只能服務一艘船舶。圖 1-1 中，每一個 Ship 的矩形方塊即代表這艘船在該船席下所耗費的服務時間(隱含等候服務的等待時間及解決裝卸貨的處理時間)，若把所有的矩形方塊進行加總，即為此期間內所有船舶在各船席間共耗費的整體服務時間。

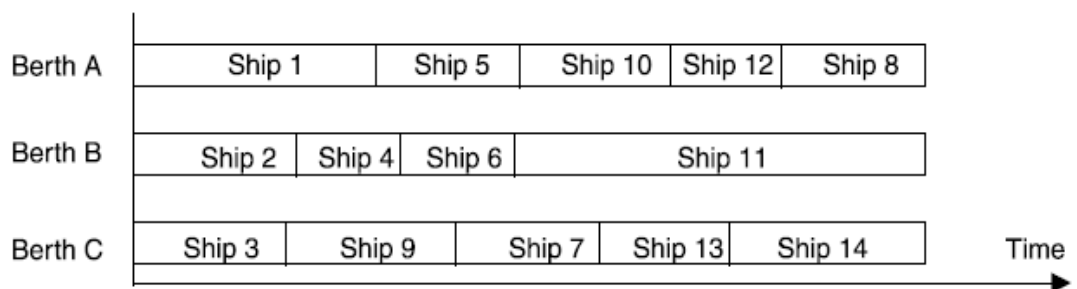


圖 1-1 動態且離散之船席指派問題示意圖 (Imai *et al.*, 2003)

針對上述提及的服務時間，可用圖 1-2 再做進一步的說明。對於每一艘船舶來說，我們考量兩項的作業時間，此為船席指派問題最為攸關的部分。如下圖所示：

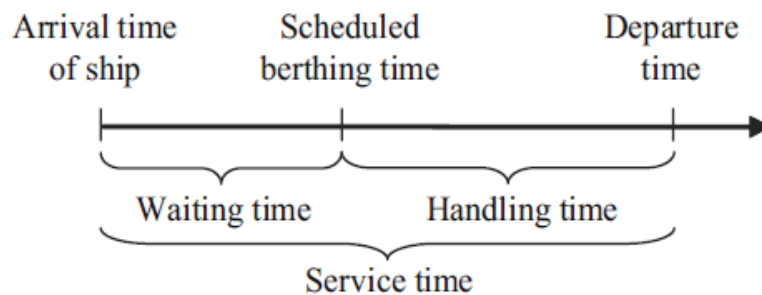


圖 1-2 船席運作時間軸 (Cheong and Tan, 2008)

首先，等待時間乃為船舶從到達後到能開始裝卸貨的時間。此外，處理時間為船舶在指定船席所要裝卸貨的時間，其兩者的總和即為船舶的服務時間。然而，由於考量每艘船的長度不盡相同，文獻上也有模型將碼頭視為一個「連續」性的資源，而船隻可以停靠在碼頭的任何一處。如圖 1-3 所示，縱軸為時間， t_i^B 為船舶 i 的抵達時間。橫軸為碼頭長度，是整體船席所能提供的位置大小， P_i 為船舶 i

的停泊位置。不同船舶所需佔用的停泊空間也因其船身長度的停靠位置而有所差異。面對此種連續型的相鄰船舶而言，若時間軸和空間軸兩者皆重疊，代表同一時間重複指派到相同泊位，自然不被允許。若兩者皆沒有重疊，則可能代表不夠善用船席資源，仍有改善其指派決策的空間。因此，針對兩相鄰船舶，時間軸和空間軸至少有一項重疊，為一較具效率並可行的法則。

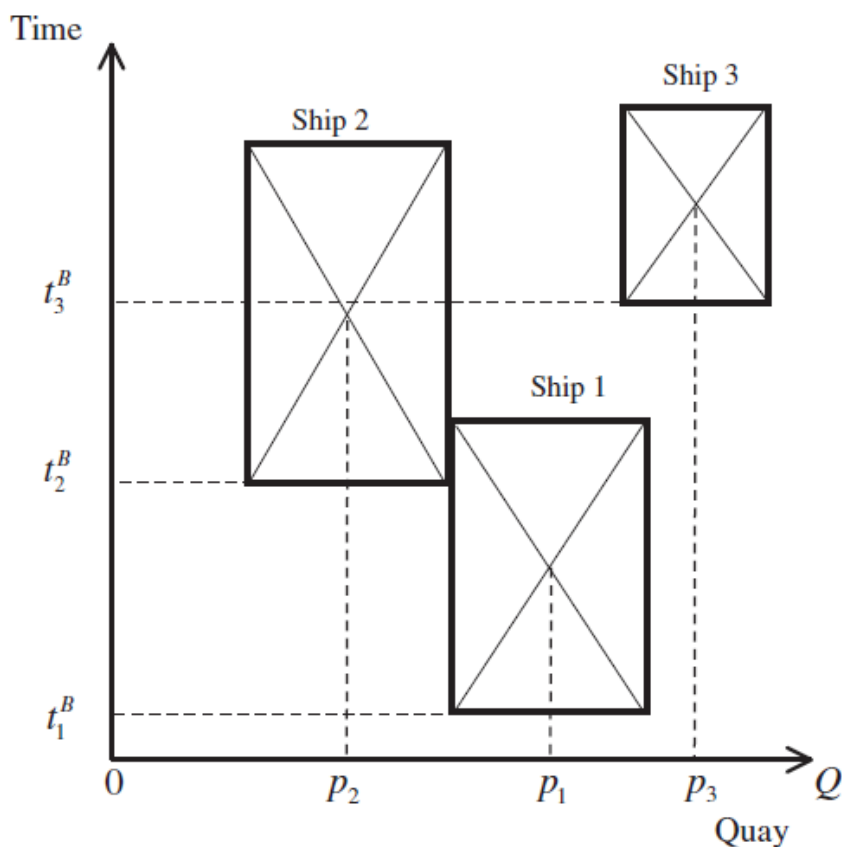


圖 1-3 動態且連續之船席指派問題示意圖 (Imai *et al.*, 2005)

因此，在離散情況下，BAP 可以被視為彼此不相關的機器排程問題；在連續的情況下，BAP 可被視為是一個裝箱問題。

另外，假設所有船隻在進行船席指派前就已全部抵達港口，該類 BAP 一般稱之為靜態模型(Static)，由於不同船舶沒有抵達順序之分，難度較易；反之，若是僅知道所有船隻預定抵達時間，而必須在抵達前安排好調度作業，則 BAP 可視為一個動態模型(Dynamic)。此種問題型態固然較難，但的確較符合實務面操作。而我們想求解的問題即是在一定的時間內，如何做最妥適的安排使得所有的船隻的等待時間及處理時間能夠最小化，若能盡可能地消化各船隻的服務時間，便能舒緩船席排班作業，不但能讓更多的船隻減少其作業成本，更能加速後端工作的流暢性，以使其整體港埠效率能更為提升。

1.3 研究方法與流程

1.3.1 研究方法

而諸如此類船席排班的問題，其複雜度隨著問題規模的上升是以階層 (Factorial) 的方式增加，在考量產業經營所面臨的實際情況時，稍具實務意涵的問題都會造成極大的求解挑戰性。因此，發展出一套有效的啟發式解法，通常具有相當的必要性。

針對排序相關問題，相較於其他各種巨集式啟發式演算法 (Meta-heuristics)，例如基因演算法 (Genetic Algorithm)、模擬退火演算法 (Simulated Annealing)、禁忌搜尋法 (Tabu Search) 等，螞蟻演算法 (Ant Colony Optimization, ACO) 搜尋與建立解答的過程，基本上就是一個循序的架構。基於此種結構上的特性，螞蟻演算法也被廣泛應用在各種排序相關的問題上，如旅行推銷員問題 (Travelling salesman problem, TSP)。在此，我們嘗試將此演算法套用在船席指派的問題上。雖然許多問題不僅僅牽涉到排序決策，但是若其中排序的決策予以確定，或者附加順序上的條件或限制，則整個問題的複雜度就大幅降低，而可以輕易加以求解。

然而，螞蟻演算法固然在架構上很適合求解排序問題，但是其搜尋空間仍然非常可觀，要發展出一個可以兼顧求解時間和求解品質的演算法，仍然是一個相當困難的課題。因此本研究嘗試運用螞蟻的導引機制，以有效限縮螞蟻搜尋的空間及縮短對應的求解時間，並以求提升船席指派問題的求解品質。

1.3.2 研究流程

本研究的研究流程請參見圖 1-4。以下為本研究之主要進行步驟，說明如下：

相關船席指派問題及螞蟻演算法的文獻回顧

1. 建構螞蟻演算法延伸到船席指派上的求解過程
2. 建構動態且離散的船席指派模式
3. 進行數值測試分析
4. 結論與後續相關進度

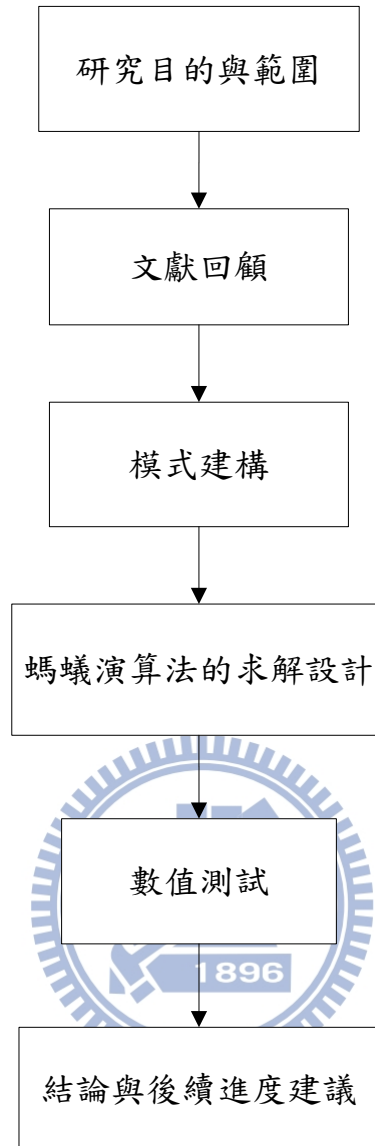


圖 1-4 研究流程圖

第二章 文獻回顧

根據上一章的分類，依照(1)船舶是否在指派前已入港，以及(2)是否考慮固定船席長度等上述兩點，可將船席指派問題大致分為(1) 靜態且離散(SDBAP)、(2) 靜態且連續(SCBAP)、(3)動態且離散(DDBAP) 以及(4)動態且連續(DCBAP)共四種類型。考量實務面的運作，希望能在船舶抵達前，已規劃完畢船席分派的工作，因此本研究選用動態的模型。

另外，處理連續型 BAP，雖然對於船舶調派較有彈性，但處理難度頗高，不易求解；反觀離散型 BAP，依舊相當常見。由其在某些海港碼頭中，船席空間的型態並非為直線型，船席自然也仍屬於離散型態。再者，離散型 BAP 對於處理問題的把握度也較高。有鑑於此，本研究將先著眼於解決離散型的船席指派問題，並採用螞蟻演算法做為此排序問題的啟發式解法。有關動態且離散的船席指派問題(DDBAP)，Theofanis *et al.* (2009)提供了較具代表性的文獻，有關的文獻回顧會在 2.1.1 小節做說明。並在 2.1.2 小節針對 DDBAP 的特性做詳述。2.2.1 小節則是針對螞蟻演算法做概述，並整理相關文獻。至於 2.2.2 小節，將會針對過去 DDBAP 曾採用螞蟻演算法的研究作介紹。

2.1 動態且離散型船席指派問題

2.1.1 動態且離散型船席指派問題之文獻回顧

綜觀 DDBAP 的發展，可略分為基本型與延伸型兩類。基本型研究在 DDBAP 當中扮演了承先啟後的角色，能將此類問題以數學模式的方式來建構，而後續學者則在日後持續對問題的本質做延伸，不斷進行模式優化，或是納入更多的求解演算法，以提升對 DDBAP 的處理效能；反觀延伸型研究之領域，則是以基本型研究為基礎，納入更多實務性要素以及管理意涵，諸如船席型態、尖峰時段及服務水準等，盼能涵蓋更多 DDBAP 的討論範圍。

首先，在基本型研究的部分，Imai *et al.* (2001)為了一改過去研究採先來先服務的處理架構，對於 DDBAP 視為一個 MIP (Mixed Integer Programming)問題。採用拉式鬆弛的啟發式解法來處理，目標式為最小化所有船舶的等待時間和處理裝卸貨時間。之後，Nishmura *et al.* (2001)參考 Imai *et al.* (2001)，在目標式相同的情況下，針對共用(Public)船席系統來討論。其假設在考慮港埠限制的情況下，能否在單一船席服務超過一艘以上的船舶。其啟發式的演算法採用基因演算法。另外，Imai *et al.* (2003)考慮有關權重的服務優先權限制，進一步修正 DDBAP 的數學規劃，目標式為最小化所有船舶的加權服務時間，在幾個數值測試的例子中，

探討包含不同權重的優先權所產生的影響，並採用基因演算法進行求解。而近年來，仍有許多研究投身於起初的 DDBAP 架構。一再嘗試對於基本的 DDBAP，優化其數學模式建構及求解的演算法。舉例來說，Monaco and Samara (2007)再針對 Imai *et al.* (2001)，強化其數學模式之限制式，使其更為嚴謹。但仍舊是以精確解的拉式鬆弛法處理同樣的最小化船舶服務時間的 DDBAP 問題。Golias *et al.* (2010)則針對 Imai *et al.* (2001)，目標式改為最小化所有船舶的等待時間及加權的處理時間，設計 lamda-optimal 的演算法進行求解。此外，Buharkal *et al.* (2011)比較了三種主要 DDBAP 模式架構的處理效能，包含了 Imai *et al.* (2001)、Cordeau *et al.* (2005)以及 Christensen and Holst (2008)等精確解之研究。

此外，在延伸型 DDBAP 的研究領域，許多 DDBAP 也納入相關的管理意涵。Cordeau *et al.* (2005)的目標式如 Imai *et al.* (2003)同樣為最小化所有船舶的加權服務時間。另外在 DDBAP 加入了具時窗限制的多場站車輛路線問題(Multi-Depot Vehicle Routing Problem with Time Window, MDVRPTW)之條件，並採禁忌搜尋 (Tabu search, TS)演算法作為求解工具。此外，Imai *et al.* (2008)將問題背景設定在船舶服務的尖峰期，在此情境下，若某船舶的等待時間超過限制，將會被指派到外部碼頭以接受服務。

近年來，也有多篇研究開始將真實情況納入考量，目標式值也不僅僅是傳統形式(諸如像等候時間及處理時間)，也會考慮一些實務性因素。舉例來說，此外，Imai *et al.* (2007)引進具有凹型船席(Indented Berth)的 DDBAP 模式，基於大型貨櫃船的等待成本較高，應擁有較高的優先順序。因此，在其 DDBAP 的架構中，大型貨櫃船將盡可能地先接受服務。而其求解方式，採用基因演算法。Imai *et al.* (2007)在船席調派的問題上，額外考慮兩個構面。除了最小化所有船舶服務時間外，也要檢視其服務水準，透過船舶駛離時間以及船席的使用績效兩者來進行評估。Hansen *et al.* (2008)在 DDBAP 的架構上，除了考慮等待時間和裝卸貨時間極小化，也納入了提早和延遲完工等要素做為目標式的一部分。透過雙目標式的模式建構，採用變動鄰域搜尋法(Variable eighborhood Search, VNS)進行求解。最後，Golias *et al.* (2009)根據船舶的優先權，採用多目標式的 DDBAP 以解決船舶服務差異化的問題，並透過基因演算法進行求解。

有關 DDBAP 之數學模式，根據實際作業情況的差異，許多學者欲涵蓋更多層面的目標範圍，或考量相關的限制。然而，典型的 DDBAP 仍是各種延伸型船席指派問題的基礎，直至今日，仍舊有學者不斷致力於基本效能的提升。此外由上述的文獻回顧可發現有 Nishmura *et al.* (2001)、Imai *et al.* (2007)及 Golias *et al.* (2009)等多篇研究採用基因演算法求解。因此，本研究希望能針對基本型的 DDBAP，改採螞蟻演算法，試圖能在問題的本質做進一步突破，以檢視其成效之優劣。另外，也希望能藉由 DDBAP，讓螞蟻演算法的應用範圍涵蓋得更為全面。

2.1.2 動態且離散型船席指派問題之數學模式

在此，本研究以 Imai *et al.* (2001) 為例，對於 DDBAP 模式的特性做說明。首先，是問題的情境假設。

1. 每一船席每次只能服務單一船舶。
2. 不考慮吃水深度以及其他技術問題。
3. 因考量船舶有其相應的貨櫃場，船舶的處理時間與指派船席具相關性。
4. 船舶抵達前，已排定其相應的船席指派資訊。
5. 船席服務不因其他原因而中斷。

其 DDBAP 數學模式如下，基本上此 DDBAP 可以被視為一個三維的混合整數規劃問題。

$$\text{Minimize } \sum_{i \in B} \sum_{j \in V} \sum_{k \in U} \{(T - k + 1)C_{ij} + S_i - A_j\}x_{jik} + \sum_{i \in B} \sum_{j \in W_i} \sum_{k \in U} (T - k + 1)y_{ijk} \quad (1)$$

subject to

$$\sum_{i \in B} \sum_{k \in U} x_{ijk} = 1 \quad \forall j \in V, \quad (2)$$

$$\sum_{j \in V} x_{ijk} \leq 1 \quad \forall i \in B, k \in U, \quad (3)$$

$$\sum_{l \in V} \sum_{m \in P_k} (C_{il}x_{ilm} + y_{ijm}) + y_{ijk} - (A_j - S_i)x_{ijk} \geq 0 \quad \forall i \in B, j \in W_i, k \in U \quad (4)$$

$$x_{ijk} \in \{0, 1\} \quad \forall i \in B, j \in V, k \in U \quad (5)$$

$$y_{ijk} \geq 0 \quad \forall i \in B, j \in V, k \in U, \quad (6)$$

符號說明

- i : 代表船席， $i = 1, \dots, I$ ，而 $i \in B$ 為船席的集合。
- j : 代表船舶， $j = 1, \dots, T$ ，而 $j \in V$ 為船舶的集合。
- k : 代表服務次序， $k = 1, \dots, T$ ，而 $k \in U$ 為服務次序的集合。
- A_j : 船舶 j 的抵達時間。
- C_{ij} : 為船舶 j 在船席 i 所需裝卸貨的時間。
- S_i : 為船席 i 變為閒置的時間點。
- P_k : 為 U 的子集合，定義為 $P_k = \{p \mid p < k \in U\}$ 。
- W_i : 為 V 的子集合，定義為 $W_i = \{q \mid A_q < S_i, q \in V\}$ 。

決策變數:

- x_{ijk} : 為二元變數，為 1 時代表代表船舶 j 在 i 船席的第 k 次序接受服務。
- y_{ijk} : 代表在船席 i ，從次序 $k-1$ 的船舶駛離，直至次序 k 的船舶 j 抵達的閒置時間。

以下則針對數學模式的意涵作說明:目標式(1)欲求得最小化所有船舶的等待時間以及裝卸貨時間。限制式(2)在確保每一艘船可以在某船席被排定順序。限制式(3)著重在每船席同時最多提供一艘船舶進行服務。限制式(4)則確定船舶為抵達後才接受服務。限制式(5)則表示,若決策變數 $x_{ijk} = 1$,代表船舶 j 在船席 i 以第 k 個次序接受服務,否則為0。限制式(6)則表示 y_{ijk} 為船席 i 在次序 k 的船舶 j 抵達前的閒置時間,必然大於或等於0。另外,我們於下一段再針對目標式(1)以及限制式(4)另做詳盡說明。

首先,對於目標式(1)的前半段,利用下圖 2-1 做說明。若沒有考量船席有閒置時間,亦即 y_{ijk} 皆為0的情況下。計算當 $x_{ijk} = 1$ 時,也就是ship j 被指派到berth i ,第 k 個位置上。 $(T - k + 1) C_{ij}$ 當中的 C_{ij} 可視為船舶 j 自己的處理時間。而 $(T - k) C_{ij}$ 則是對之後船舶所需等候的等待時間。 $S_i - A_j$ 則是所有船舶都須等候船席開始提供服務的等待時間,但須扣除其到達的時間。至於船席的閒置時間 y_{ijk} 對於目標式的影響,則類似 C_{ij} 之效果,同樣乘上 $(T - k + 1)$ 。

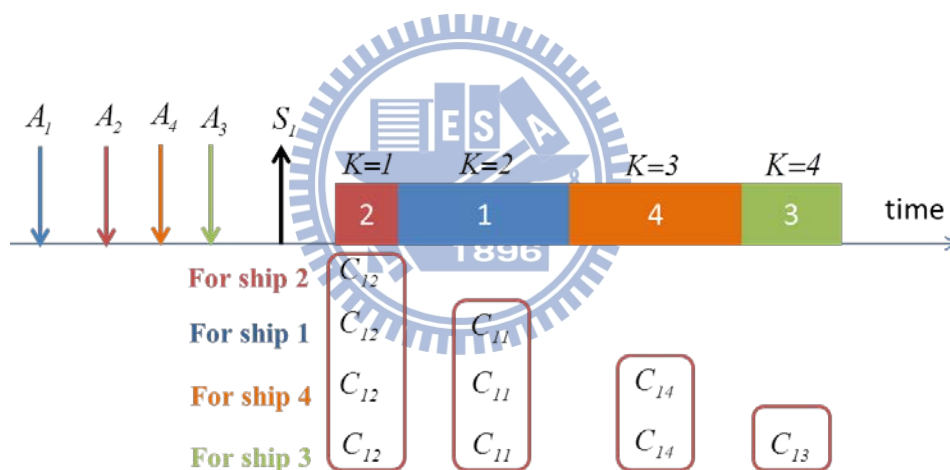


圖 2-1 BAP 目標式示意圖 (Imai *et al.*, 2001)

另外,在此要繼續針對限制式(4)作說明。當 x_{ijk} 為1,限制式(4)可寫為下式(8)。數學式(8)即意味當 x_{ijk} 為1時,對任一Ship j 而言,左式為上一次序結束後的完工時間加上次序 k 的船席閒置時間之和不能小於Ship j 的抵達時間,才能接受服務。另外,當 x_{ijk} 為0,限制式(4)可寫為下式(9)。數學式(9)代表當 x_{ijk} 為0時,對任一Ship j 而言,左式必然會大於或等於0。

$$S_i + \sum_{l \in V} \sum_{m \in P_k} (C_{il} x_{ilm} + y_{ijm}) + y_{ijk} \geq A_j \quad \forall i \in B, j \in W_i, k \in U \quad (8)$$

$$\sum_{l \in V} \sum_{m \in P_k} (C_{il} x_{ilm} + y_{ijm}) + y_{ijk} \geq 0 \quad \forall i \in B, j \in W_i, k \in U \quad (9)$$

再透過圖 2-3 來說明限制式(4)，其船舶指派到船席的情況。細線代表船舶從抵達到接受服務的等待時間；粗線代表船舶在裝卸貨的處理時間；虛線代表船舶等待下一艘指派船舶的閒置時間。因此，如圖 2-3 可看出Ship 1、Ship 2 及 Ship 3 有船舶等候服務的等待時間，而Ship 4 和Ship 5 則是有船席等候服務的閒置時間 (y_{i42} , y_{i53})。

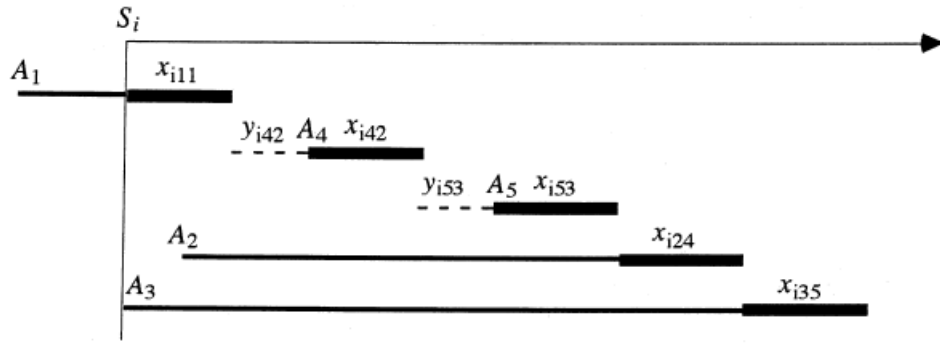


圖 2-2 船舶指派順序到船席之示意圖 (Imai *et al.*, 2001)

Imai *et al.* (2001)之 DDBAP，為一混合整數規劃問題(MIP)，對於原問題採用拉式鬆弛法(Lagrangian relaxation)，透過梯度最佳解(subgradient optimization)來處理。因此，可將原問題轉變為以下類型：

$$\begin{aligned} \text{Minimize } & \sum_{i \in B} \sum_{j \in V} \sum_{k \in U} \{(T - k + I)C_{ij} + S_i - A_j\}x_{jik} + \sum_{i \in B} \sum_{j \in W_i} \sum_{k \in U} (T - k + I)y_{ijk} \\ & - \sum_{i \in B} \sum_{j \in W_i} \sum_{k \in U} \lambda_{ijk} \left\{ \sum_{l \in V} \sum_{m \in P_k} (C_{il}x_{ilm} + y_{ijm}) + y_{ijk} - (A_j - S_i)x_{jik} \right\} \end{aligned} \quad (10)$$

subject to

$$\sum_{i \in B} \sum_{k \in U} x_{ijk} = 1 \quad \forall j \in V, \quad (11)$$

$$\sum_{j \in V} x_{ijk} \leq 1 \quad \forall i \in B, k \in U, \quad (12)$$

$$x_{ijk} \in \{0, 1\} \quad \forall i \in B, j \in V, k \in U \quad (13)$$

$$y_{ijk} \geq 0 \quad \forall i \in B, j \in V, k \in U, \quad (14)$$

其中， λ_{ijk} 為船席*i*，船舶*j*在第*k*個次序下的拉式鬆弛法乘數，為一非負值。經過處理由於 y_{ijk} 已不存在限制式，可再將問題改寫成下述問題：

$$\text{Minimize } \sum_{i \in B} \sum_{j \in V} \sum_{k \in U} \{(T - k + I)C_{ij} + S_i - A_j\}x_{jik} - \sum_{i \in B} \sum_{j \in W_i} \sum_{k \in U} \lambda_{ijk} \left\{ \sum_{l \in V} \sum_{m \in P_k} C_{il}x_{ilm} \right\} \quad (15)$$

subject to (11)-(13)

如此一來，使得船席 i 以及次序 k 轉為一組合問題，提供船舶 j 作指派。原先的三維整數問題便可縮減至二維的指派問題(Assignment problem)來加以處理。透過限制式(4)的數學放鬆處理，可由 Total Unimodularity 的性質，把模式變為一 LP 問題。也就是能讓式(13)加以放鬆，仍然可以保有整數解。

而本研究乃希望同樣的問題情境下能針對此種核心的 DDBAP，透過我們所提及的螞蟻演算法以及自行開發的指派規則來加以解題。也將問題所給定抵達時間做為排序時的相關資訊，希望能進一步突破原有文獻的處理效能，更能進一步推廣到其他更多實務面的船席指派問題上。

2.2 螞蟻演算法之概述

螞蟻演算法在處理排序相關問題的能力顯著，過去在船席指派的問題上卻鮮少有研究把螞蟻演算法投入此領域做延伸，因此本研究希望透過此演算法應用於船席指派問題上，以期得到更卓越的成效與思維。

2.2.1 螞蟻演算法簡介

螞蟻演算法(Ant Colony Optimization, ACO)是由模仿自然界中螞蟻覓食的行為而來，包含了一系列陸續發展的演算法。第一個相關的演算法稱為螞蟻系統演算法(Ant System Algorithm, ASA)，係由 Dorigo 的博士論文(Dorigo, 1992)中所提出，之後並正式發表(Dorigo *et al.*, 1996)。基本上，螞蟻在搜尋路徑的選擇上是隨機的，但是螞蟻在搜尋時，會沿路留下叫做費洛蒙(Pheromone)的揮發性化學物質，進而影響後續其他螞蟻的路徑選擇，因為濃度越高的路徑則後續螞蟻越有可能跟隨。由於費洛蒙的揮發量是隨著時間遞增，同一地方若未再有其他螞蟻經過則濃度會慢慢減低。另外，在較長的路徑上，搜尋的時間較長，費洛蒙被蒸發的較多，造成累積的速度較慢；相對地，在較短的路徑上，費洛蒙蒸發的較少，濃度較高。如此，漸漸地大部份的螞蟻都會選擇較短的路徑。

螞蟻演算法在組合性的最適化問題上成效卓著，也包含了許多排序問題。目前螞蟻演算法的發展已有相當廣泛的應用，除了之前提及之 TSP 外(諸如 Dorigo and Gambardella, 1997、Stutzle and Hoos, 1997)，有關螞蟻演算法一般的介紹，各種衍生的演算法，以及應用的領域，可參考以下文獻：Dorigo and Stutzle (2004)、Dorigo *et al.* (2006)、Mullen *et al.* (2009)。

2.2.2 螞蟻演算法在船席指派問題之應用

檢視過去螞蟻演算法投入船席指派問題的研究相當稀少，主要僅有兩篇研究針對船席指派問題，採用螞蟻演算法進行求解。分別是 Tong *et al.* (1999) 以及 Cheong and Tan (2008)。Tong *et al.* (1999) 嘗試將螞蟻演算法應用至連續型的 BAP。

但其也點出其研究並不能多有效強化其求解性能，但希望透過螞蟻優化的方式來延伸到動態的連續型 BAP。目標式想盡量減少所需服務船舶的碼頭長度，並非減少所以船舶的服務時間。而受到一些空間和時間的限制，實驗結果發現，演算法沒辦法對於真實世界的問題提出好的指引，問題型態與本研究不同。

Cheong and Tan (2008)提出了一種多目標的群聚螞蟻演算法(MOMCAA)。其中，每一隻螞蟻僅負責單一船席如何指派的規劃順序。其架構如下，先以螞蟻演算法導引出一組船舶的指派順序後，再針對每艘船舶以另一螞蟻演算法產生出相應之船席，為避免兩螞蟻演算法的隨機性產生干擾，本研究認為一旦導引出船舶的指派順序，即能透過一套具系統性的啟發式指派法則來產生船舶指派決策。因此，我們認為本研究與上述研究在問題情境、數學模式或是求解演算法設計仍然有所差異。因此，我們可知本研究無論是在問題特性、數學模式建構或是求解演算法都與上述兩研究有所差異。因此，本研究希望在減少繁瑣的數學設計情況下，透過一簡單的指派順序，即能真正地克服 DDBAP 的問題。



第三章 求解演算法

螞蟻演算法的基本架構可參見 3.1 節。由於船席指派問題為 NP-hard 之問題，面臨船舶數量較多時，問題的搜索空間將變得相當龐大。為了達到縮小搜索空間又兼顧求解品質的前提，本研究先將求解演算法區分為兩個環節。首先，透過螞蟻演算法產生船舶服務的指派順序。再者，在該組指派順序下，透過本研究自行開發的貪心指派法則(Greedy Assignment Rule, GAR)產生船席指派決策。最後，透過 GAR 的求解結果，得到該組次序的目標式值。我們將結果進一步回饋給螞蟻演算法，藉由式(17)與式(18)更新費洛蒙值，以供下一代所使用，進而產生出更佳的指派順序與船席指派決策，求解流程圖請參見圖 3-1。



圖 3-1 求解演算法之簡要示意圖

3.1 螞蟻演算法之一般求解過程

螞蟻演算法的設計，乃模擬自然界中螞蟻藉由費洛蒙的遺留(Laying)及跟隨(Following)，達到彼此交換訊息、分享經驗的行為，進而設計具有探索能力及收斂特性的搜尋法，來求解大型的組合性最佳化問題(Combinatorial Optimization Problems)。

為說明螞蟻演算法的基本架構，以下以最單純的螞蟻系統演算法(ASA)針對經典的旅行推銷員問題(Travelling salesman problem, TSP)，來說明其基本架構。將各螞蟻放在任意一點做為起點，在建構路線的過程中，假設某螞蟻在次序 k 位於某一節點 i ，在選擇要走向下一節點 j 時，其機率性的選擇行為會受到以下兩個因素所影響：足跡(Trail level)及吸引力(Attractiveness)。足跡則是前面螞蟻走過的路

徑對後面螞蟻的影響，也就是藉由費洛蒙的遺留及蒸發來影響後續螞蟻選擇路徑的機制；吸引力表示路徑本身對螞蟻的影響，以啟發法權值(Heuristic Value)表示。假設在建構路線的過程中，某螞蟻位於某一節點 i ，在選擇要走向下一節點 j 的機率之可用式(1)來表示，其中 t 指的是在第 t 次的遞迴運算。

$$p_{kij}(t) = \begin{cases} 0 & \text{if } (i, j) \in B_{ki} \\ \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in B_{ki}} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta} & \text{otherwise} \end{cases} \quad (16)$$

- $p_{kij}(t)$: 第 t 次遞迴，螞蟻在第 k 階段從節點 i 選擇節點 j 的機率。
 B_{ki} : 螞蟻在節點 i 時，不能前往的節點。
 $\tau_{ij}(t)$: 在第 t 次遞迴中，路徑 (i, j) 上費洛蒙的量。
 α : 螞蟻選擇路徑時，費洛蒙影響的權重。
 η_{ij} : 啟發法權值，表示路徑 (i, j) 的吸引力
 β : 螞蟻選擇路徑時，啟發法權值影響的權重。
 p : 每一迭代的螞蟻數量。
 t : 迭代標號。

當每隻螞蟻均完成其路徑，則為一回合結束，然後將所有 m 隻螞蟻走過的路徑參考其所對應的解，進行費洛蒙矩陣的更新。由於螞蟻搜尋回報的解有優有劣，因此更新時必須以搜尋結果的品質來決定累積費洛蒙的量，如式(17)和(18)；其中， $f(t)$ 代表螞蟻所對應解的目標式值， Q 為一固定常數， $\Delta\tau$ 代表費洛蒙之增加量。

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}(t) \quad (17)$$

$$\Delta\tau_{ij}(t) = \begin{cases} \frac{Q}{f(t)} & \text{if arc}(i, j) \text{ is used by ant } k \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

- ρ : 每次遞迴間，費洛蒙的蒸發比率，其值介於 0~1。

一般而言，搜尋演算法的中止條件有幾種常見的策略可以選擇。一是限制執行固定的回合數，例如執行 100 回合後停止；或者限制一定的執行時間，例如執行 30 秒後停止。再者，也參考求解品質改善的進度，例如未改善次數超過連續 10 回合後停止。

螞蟻的搜尋空間相當龐大，太強烈的費洛蒙將影響機制，可能會限縮螞蟻探索的可能性，容易收斂在區域解；但太微弱的費洛蒙也會影響機制，將耗費相當長的搜尋時間，甚至會有求解發散的問題。因此，在訂定費洛蒙時需參照問題特性做審慎評估。使得求解過程中，不會在初期就立即收斂，也能有效在迭代當中顯著地提升求解品質。此外，傳統的 TSP 問題與船席指派問題有所差異，如何把

上述的螞蟻演算法套用在船席指派的問題上，遂成為本研究的接續探討的核心課題。

3.2 求解演算法之設計

在解決 DDBAP 的問題上，我們已知所有的船席大小可容納任何一艘的船舶，且在船隻尚未抵達港口前，得以先獲知船席到港時間之資訊，並能有效地提供各船隻如何停泊的相關資訊，包含船隻指派的順序、時間以及相應的船席位置。

有了上述資訊，我們乃欲透過 3.2.1 小節，說明已 ASA 先對 DDBAP 提供一組可行的指派順序。再透過 3.2.2 小節的介紹，利用我們所設計的啟發式指派法則，進行 DDBAP 的船席指派作業與目標式的求解。最後，3.2.3 小節是螞蟻演算法的求解流程圖，並另做其他補充。

3.2.1 以螞蟻演算法產生指派次序

根據 3.1 節的式(16)，假設螞蟻已在上一次序選定船舶 i ，在選取下一次序的船舶 j 時，其機率性的選擇行為會受到 $\tau_{ij}(t)$ 與 η_{ij} 兩者所牽引。並利用輪盤法，得以逐步為該代該隻螞蟻產生一組指派順序。

圖 3-2 為螞蟻演算法指派一組順序的示意圖，船舶編號如圖 3-2 依序分別為 1, 2, 5, 3, 4, 10, 8, 9, 6, 7。可看出每一個 Stage 都有其相應的指派船舶。以 Stage3 以及 Stage4 為例，即是在第 3 次序下選取 Ship 5，並在第 4 次序下選取 Ship 3。

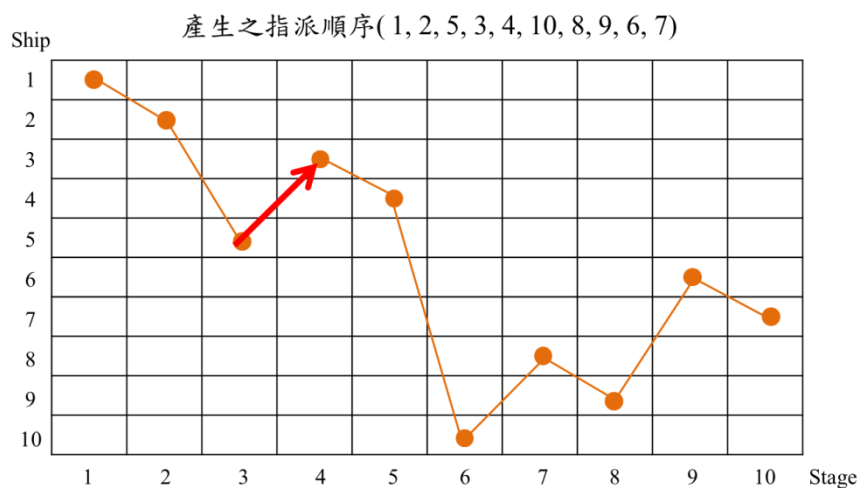


圖 3-2 螞蟻演算法一節點選取示意圖

另外，本節繼續對 $\tau_{ij}(t)$ 與 η_{ij} 做說明。首先， $\tau_{ij}(t)$ 代表在第 t 代中已選擇船舶 i 的情況下，再選取下一節點時，船舶 j 所對應的費洛蒙濃度大小。若船舶 j 的費洛蒙濃度越大，則必然有較高機率會於下一次序選擇若船舶 j 。

另外， η_{ij} 則是代表啟發值大小。透過兩艘船的抵達時間遠近，能決定是否先行指派，此部分與費洛蒙大小無關。若船舶 j 的抵達時間比船舶 i 要延遲許多，選取機會 η_{ij} 較低。但若是若船舶 j 的抵達時間比船舶 i 要來得早，自然會考慮優先選取。關係式如下頁數學式(19)。其中， k_2 乃根據不同問題及規模大小做調整。而為避免 $A_j < A_i$ 的情形，使得 η_{ij} 無限大，因而設立 k_1 ，以合理化數值大小。

$$\eta_{ij} = \frac{1}{k_1 + k_2 * \text{Max}(A_j - A_i, 0)} \quad (19)$$

由上述內容可知，透過 $\tau_{ij}(t)$ 與 η_{ij} 的導引，得以求出一組服務次序。再透過下一小節 3.2.2 由自行設計的指派原則給予這組次序，指派的選用船席，並可得到其相應的目標式值。透過尋解機制的不斷優化，以找到更好的服務次序作為船席指派之用。

3.2.2 由指派順序進行船席指派

過去，螞蟻演算法在處理 TSP 的問題上，一旦產生出次序，即有相應的解及其目標值。然而本研究中，除了給定其指派順序後，還需要設計出在該次序下，如何產生相應的船席指派順序。以下我們提出了兩版本的貪心指派原則(GAR, Greedy Assignment Rule)GAR1 與 GAR2 供求解之用。

指派次序的每一船舶，以 GAR1 來說，指派原理乃是依照所有船席中處理時間及等待時間之總和最小者，進行船席指派；等待指派完一次序後，將會得到該船舶已指派下所耗用的等待時間、處理時間以及該船席所預計的完工時間。會立即更新資訊，做為下一艘指派船舶之參考。等待該組指派順序已全數服務完畢後，將會加總所有的船隻等待時間以及處理時間，此一目標式值即為我們做為評判此組順序好壞的依據。

以圖 3-3 為例，倘若 Ship 3 為目前第 4 艘需要指派到船席進行服務的船隻。共有 3 個船席的相關停泊資訊可供參考。Ship 3 的抵達時間為 18，Berth 1、Berth 2 以及 Berth 3 可以開始提供服務的時間分別為 20、26 及 17。若選擇前往 Berth 1，服務時間為 14(等待時間 2，處理時間 12)；若選擇前往 Berth 2，服務時間為 23(等待時間 8，處理時間 15)；若選擇前往 Berth 3，服務時間為 18(等待時間 0，處理時間 18)；因此，比較不同船席的服務時間下，我們會擇其服務時間最短者，並非是最早可提供服務的船席。也就是說，我們會選擇 Berth 1 作為 ship 4 的指派港口，並且將船席 1 的完工時間進一步更新，以作為之後第五艘船 ship 4 之用。

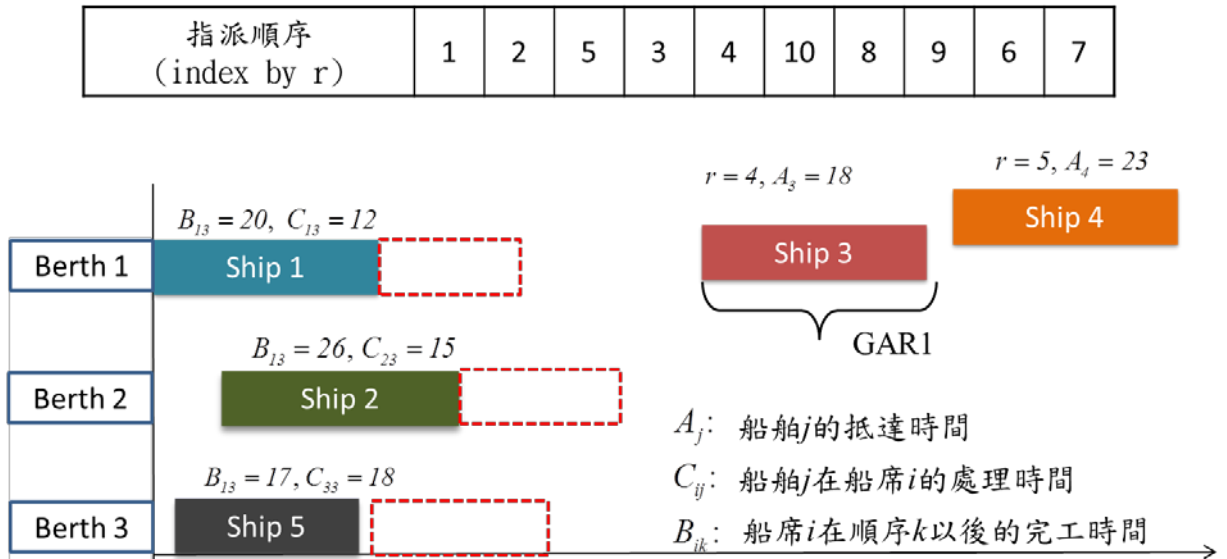


圖 3-3 GAR1 示意圖

而GAR2的方式，則是有別於GAR1。同時針對指派順序中接續兩次序的船舶進行考量。以圖 3-4 作說明，若現在針對第 4 艘船Ship 3 及第 5 艘船Ship 4 進行指派作業，因有 3 個船席，所以將有 3^2 種組合，仍擇其最優者作為選取組合標的。但需注意的是，最後只先確認前一個次序的船席指派作業，進行更新（例如最佳的選擇組合為Ship 3 先選船席 1，而Ship 4 再選船席 3）。GAR2 僅先確認Ship 3 指派到船席 1。第 5 艘船Ship 4 則會與第 6 艘船Ship 10 同時考量後，才進行確認動作）。GAR2 相較於GAR1 更能通盤考量整體決策，解決GAR1 可能存在的盲點，並且在運算時間上的負擔仍可接受。因此，此種指派原則GAR2 可視為有別GAR1 的指派法則。有了GAR1 與GAR2 兩種指派原則，我們便可將螞蟻演算法所給定之船舶指派順序進行船席指派，並得到該組次序下的目標式值。

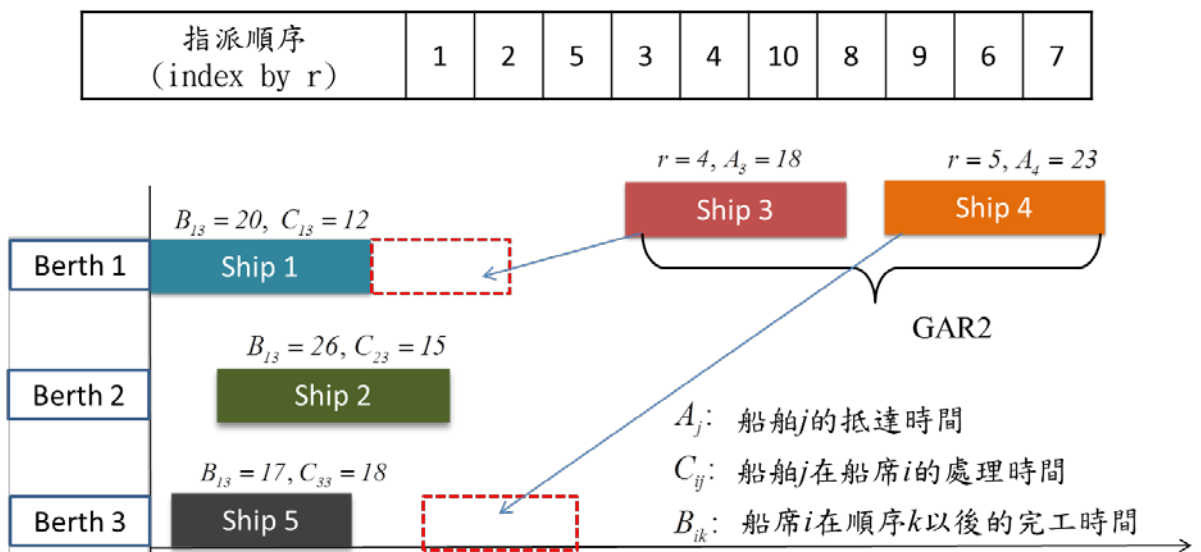


圖 3-4 GAR2 示意圖

在此，我們必須強調螞蟻演算法在產生船舶指派的優先順序時，會將已選擇的船舶列入禁忌清單(Tabu List)。確保往後所指派的船舶必然與先前不同，直至所有等候服務的船舶已全數指派完畢。也就是說，每一個次序都將對應到不同的船舶；再者，GAR 從船舶的指派順序轉化為各船舶之相應船席的過程中，必須從各船席的完工時間之後，始能進行服務。此舉確保單一船席同時不能服務超過一艘船舶。從上述兩點之說明，我們可以了解本研究的求解過程並不會遭遇不可行解。

3.2.3 結論與求解設計流程圖

而 BAP 的處理流程可參見圖 3-5。首先，我們先決定整體螞蟻演算法所欲設定的相關參數值，如初始費洛蒙、螞蟻每隻所帶有的費洛蒙值、啟發值的選用設計、迭代消散比率、費洛蒙與啟發值兩者的相關權重等。再以此為基準，透過 3.2.1 小節產生一組服務順序。進而在 3.2.2 小節指派到不同船席進行服務，並決定該順序所屬的目標式值。



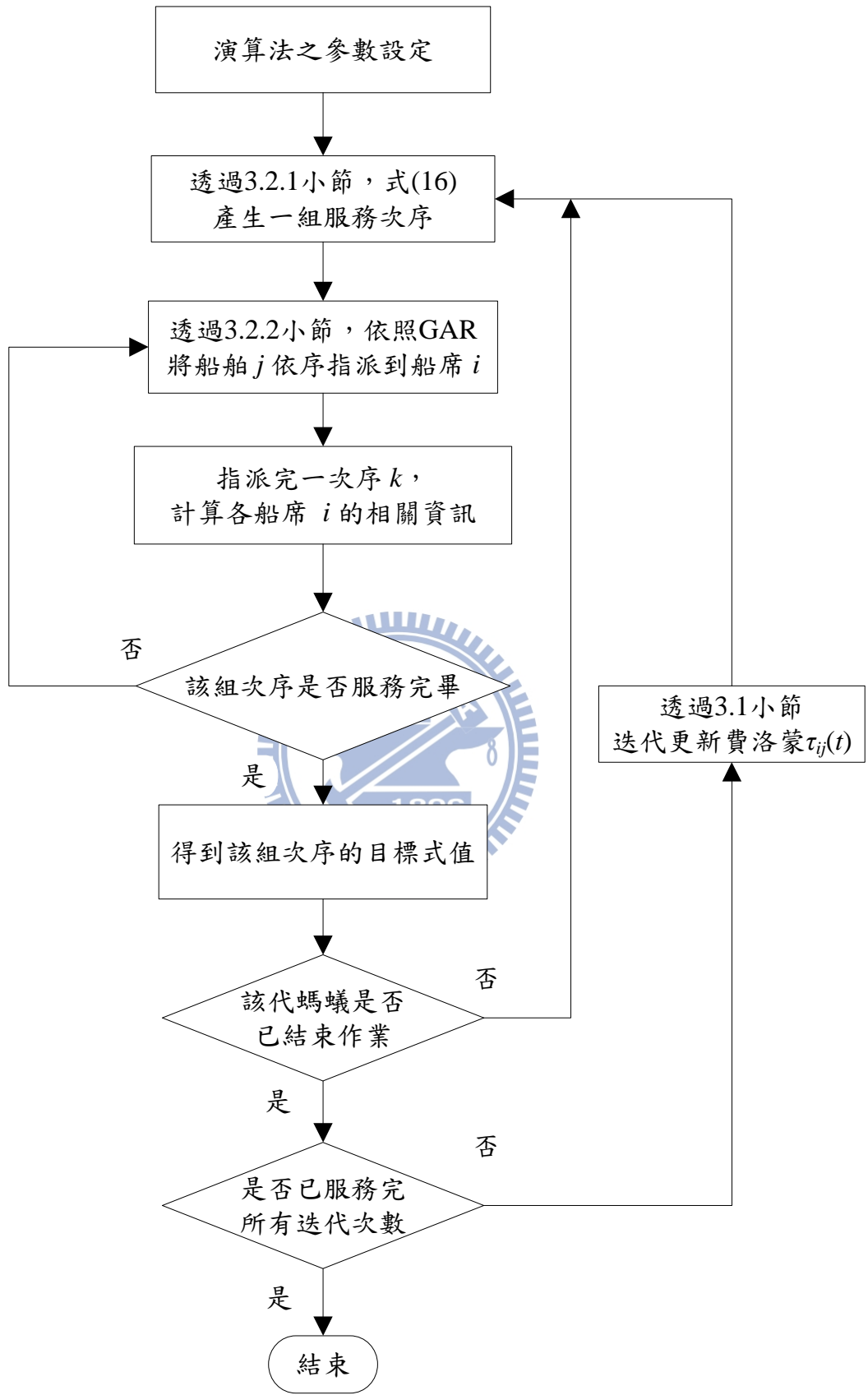


圖 3-5 船席指派問題之流程示意圖

流程圖 3-5 中的該代螞蟻乃為每一迭代所選用的螞蟻數量，等同於試行次數。而在每一迭代的試行過程中，彼此間沒有相互關係。直至該代求解結束時，倘若螞蟻的迭代次數尚未終止，仍有需要進行下一迭代(t)的排序作業時，才會進行更新費洛蒙 $\tau_{ij}(t)$ 的作業。如 3.1 小節的式(18)，若總服務時間耗時長，螞蟻在該指派路徑所殘留下的費洛蒙濃度就會較少。相反地，若是總服務時間相對短，代表螞蟻在該指派路徑所殘留下的費洛蒙濃度就會較高。如此一來，便會影響下一代螞蟻產生次序的可能性。在迭代中，螞蟻得以持續產生服務次序，以相應的目標式值回饋給下一代，以便產生新次序的情況下，我們期待求解的目標式值會越來越小。

有了螞蟻演算法的求解設計和給定次序下的 GAR 法則，我們將在下一章節進行數值測試，以研判此方法的有效性。



第四章 數值測試

4.1 題目設計

在題目設計的部分，本研究依照船舶數量的規模大小區分為 25 艘船、75 以及 150 艘船等三大類群。每一個類群安排 10 個測試問題，並與 2.1.2 小節 Imai *et al.* (2001) 之研究進行結果比較，總共有船席 A 到船席 E 共 5 個船席可供船舶到港停靠，以下將針對抵達時間與處理時間做介紹。

關於船舶的抵達時間，為隨機產生，其服從平均值為 300 的指數分配。數值設計如下，首先，不同 Ship j 先產生一 U_j 服從 Uniform(0,1) 的隨機亂數。已知 $\beta=300$ ，再透過下式(20)的轉換，以得到各個 ship j 的預定抵達時間。

$$A_j = -\beta \ln U_j \quad (20)$$

若是抵達時間成均等分配，則整體時間內之抵達情況會較為穩定，未必能分析出求解品質之優劣。但本設計採用指數分配，其為一右偏的分布狀況，因此須同時考量前、後期可能分別存在有船隻擁塞有離群值的兩種情況，求解也會較具挑戰性。表 4-1 為 25 艘船中第 1 個問題所擷取的部分例題資料，內含船舶 1 到船舶 10 的抵達資訊。

Ship j	1	2	3	4	5	6	7	8	9	10
$U_j \sim U(0,1)$	0.94	0.31	0.63	0.06	0.51	0.60	0.46	0.81	0.57	0.60
A_j	19	353	141	824	203	154	236	64	171	152

表 4-1 數值測試 - 問題 1(25 艘船)的抵達時間資訊

而船舶的處理時間為盡可能符合現狀，因此，乃先隨機指定一船舶 j 至特定船席所需服務的時間最為短暫，我們令之為 g_j (船席 1 到船席 5，被選擇的機率皆相同)。其餘船席則因距離的遠近而成線性關係，兩相鄰船席會有差距 α ，而調整其裝卸貨的處理時間。其中每一船舶 j 之最短處理時間為服從 1~200 的均等分配，令之為 h_j 。因此，我們可以將與船席位置相關的處理時間 C_{ij} 設定為 $(1 + \alpha |i - g_j|)h_j$ 。數值設計如下，首先，Ship j 須挑選一處理時間最短的船席。每一 Ship j 會先產生 U_j 服從 Uniform(0,1) 的隨機亂數。若 $U_j \sim U(0, 0.2)$ ，則假定 berth 1 的處理時間最短。若 $U_j \sim U(0.2, 0.4)$ ，則假定 berth 2 的處理時間最短。並以此類推。接續才對於處理時間最短之船席給予其數值，須服從 1~200 的均等分配。最後，針對其他鄰近船席的處理時間再做線性調整。

圖 4-1 可示例各船席的線性關係。若假定 $U_j \sim (0.4, 0.6)$ ，則設定 berth 3 為船席服務時間中最短者。另外，產生其處理時間 $C_3 \sim U(1, 200)$ ，假定得到一值為 10。便可依此，再對其他船席的服務時間之用，在此並假設相鄰船席的處理時間會相差 10%。表 4-2 為 25 艘船中第 1 個問題所擷取的部分例題資料，內含船舶 1 到船舶 10 指派至不同船席所需耗費的處理時間。

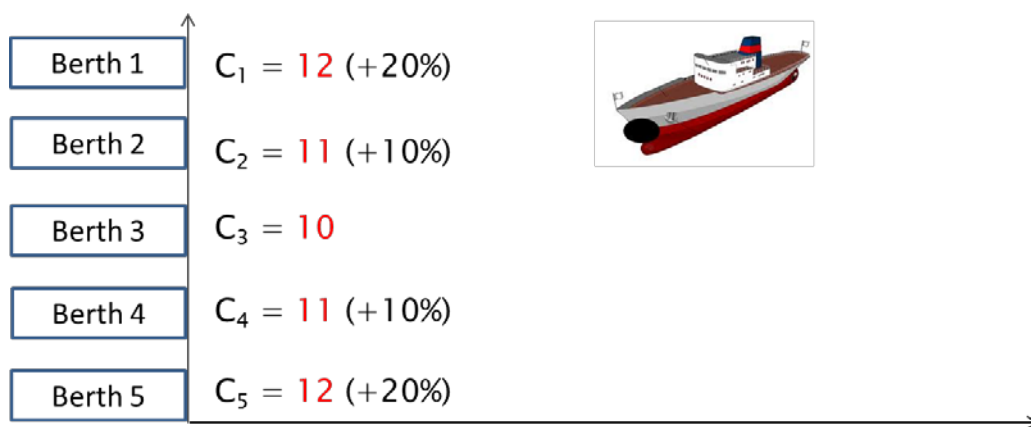


圖 4-1 船席處理時間之線性關係圖

Ship j	1	2	3	4	5	6	7	8	9	10
Best berth	1	3	2	4	3	3	2	5	4	4
Min handling time	135	13	113	50	150	81	200	90	199	17
Berth 1	135	16	125	66	181	98	220	132	265	23
Berth 2	148	15	113	60	164	89	200	120	241	21
Berth 3	163	13	125	54	150	81	220	109	219	19
Berth 4	179	15	137	50	164	89	242	99	199	17
Berth 5	197	16	151	54	181	98	266	90	219	19

表 4-2 數值測試 - 問題 1(25 艘船)的處理時間資訊

4.2 螞蟻演算法迭代效果分析

有了船舶抵達時間與相應到各船席的處理時間資料後。另外，再就題目設計的資料，搭配螞蟻演算法配適出最符合的參數值，如下表 4-3 所示：分別就起始費洛蒙、迭代次數、單代螞蟻隻數、費洛蒙消散比率、費洛蒙影響程度、啟發值影響程度以及每隻螞蟻所帶有的費洛蒙值做假設，以作為例題測試之依據。其中，每一代的螞蟻數，本研究以船舶數/2 作為單代螞蟻隻數。其原因為當船舶數較多時，越難從有限選擇中，嘗試導引出較佳的組合。因此，單代螞蟻隻數的設定會隨著船舶數增加而上升，也提高大規模問題的搜索空間。

費洛蒙的初始值	1
螞蟻演算法的迭代次數	50
每一迭代所選用的螞蟻數量	船舶數/2
ρ (每一迭代中所會消散的費洛蒙比例)	0.1
α (費洛蒙影響選擇機率的權重)	1.25
β (啟發值影響選擇機率的權重)	1
Q (每一隻螞蟻所帶有的費洛蒙大小)	2500(for 25ships) 7500(for 75ships) 15000(for 150ships)

表 4-3 數值測試 - 螞蟻演算法之參數值

表 4-4 為 25 艘船、問題 1 透過 GAR1 試行 10 次後所得之結果，可發現在不同的試行結果中，末代的平均服務時間能顯著低於初代的平均服務時間，而迭代中最小的服務時間也同樣顯著低於首代的最小服務時間。這說明螞蟻演算法在船席指派的問題中，藉由迭代中不同次序所相應的目標式時間，能有效地回饋給系統產生更為有效率的次序以供指派，使得不管是平均的服務時間以及最小的服務時間都顯著地隨著迭代次數增加而有所下降。另外，並可由下頁的圖 4-2，發現數值差距的變化程度，的確能有所改善。

重複試行結果	1	2	3	4	5	6	7	8	9	10
首代最小目標式值	3817	3572	3771	3572	3687	3662	3832	3754	3794	3826
首代平均目標式值	4785	4994	4941	4985	4790	4892	4743	4863	4870	4809
末代平均目標式值	4000	4138	4189	4399	4552	4387	4305	4535	4324	4921
迭代最小目標式值	3437	3515	3516	3556	3535	3516	3516	3516	3371	3538

表 4-4 數值測試 - 問題 1(25 艘船)的模擬結果

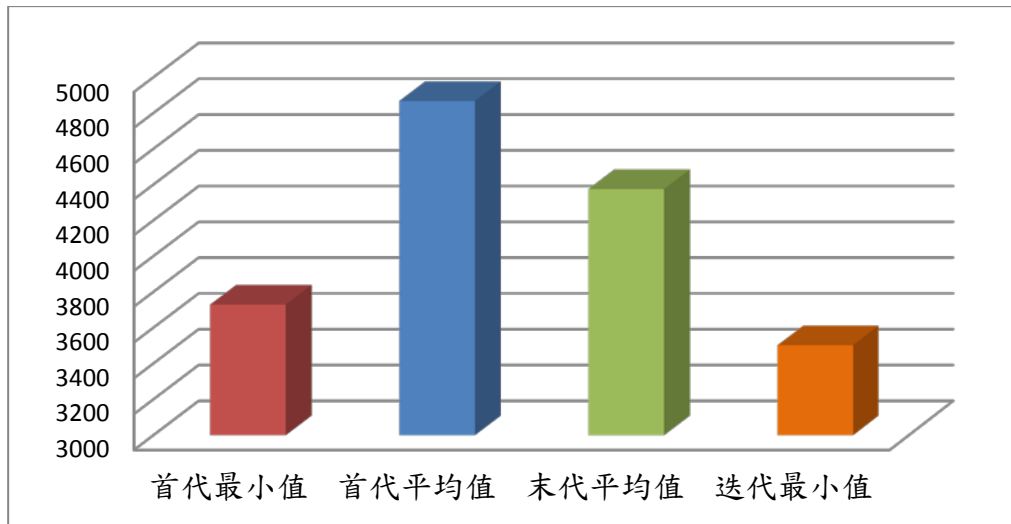


圖 4-2 數值測試 - 問題 1(25 艘船)的模擬分析圖

4.3 演算法數值測試結果

另外，本研究也將此例題資訊與 Imai *et al.* (2001)當中的演算法做比較。如下表所示。GAR1 為本研究在各例題中利用螞蟻演算法所能求解之最佳答案，而 GAR2 同樣也為本研究在各例題中利用螞蟻演算法所能求解之最佳答案。

由下頁的表 4-5 可發現遑論 25 艘船、75 艘船及 150 艘船的問題，一次採用考慮兩指派船席的 GAR2 都要比 GAR1 來得更加優秀。另可由平均值看出在 25 艘船中平均目標式值可改善 1.21%，而在規模更大的題型中，透過 GAR2 可以更有效提升指派效率。75 艘船中平均目標式值可改善 4.79%。150 艘船中平均目標式值可改善 7.83%。顯然 GAR2 的確要比 GAR1 較具求解能力。

另外，SUBG 乃為 Imai *et al.* (2001)在該篇研究中所利用三種不同的 Lagrangian 鬆弛法中挑選一最佳之數據作為比較基準。因此，我們欲對於 SUBG 與指派法則中較好的 GAR2 做數值測試。下頁的表 4-6，即是以 GAR2 與 Imai *et al.* (2001)的 SUBG 做比較。可發現在 25 艘船、75 艘船以及或是 150 艘船等 30 個問題中，除了例題 15、例題 25 及例題 29 外，大致上都有明顯的領先。另可由平均值看出在 25 艘船中平均目標式值可改善 5.59%，而在規模更大的題型中，透過 GAR2 還可更有效提升指派效率。75 艘船中平均目標式值可改善 8.86%。最後，在 150 艘船中，我們發現到目標式值的變異性會由於題目規模的上升而有所增加。然而，平均目標式值的改善仍是更為明顯，可達到 9.14%。顯然 GAR2 比原先 SUBG 的求解效能來得更卓越，在大例題中，更能看見其顯著的改善效果。

# of ships	Problem instances	GAR2	GAR1	Compare
25	1	3331	3371	-1.19%
	2	3456	3477	-0.60%
	3	3002	3050	-1.57%
	4	2995	3065	-2.28%
	5	3661	3707	-1.24%
	6	3962	4004	-1.05%
	7	2569	2570	-0.04%
	8	3360	3349	0.33%
	9	3694	3726	-0.86%
	10	2961	3070	-3.55%
	Average	3299.1	3338.9	-1.21%
75	11	7616	8255	-7.74%
	12	6340	6655	-4.73%
	13	5128	5493	-6.64%
	14	7943	8490	-6.44%
	15	7722	7897	-2.22%
	16	6074	6649	-8.65%
	17	4890	4926	-0.73%
	18	7685	8228	-6.60%
	19	5772	5813	-0.71%
	20	5125	5305	-3.39%
	Average	6429.5	6771.1	-4.79%
150	21	24777	27216	-8.96%
	22	13014	13767	-5.47%
	23	18863	20191	-6.58%
	24	19436	20612	-5.71%
	25	23347	26507	-11.92%
	26	17233	19453	-11.41%
	27	12460	13074	-4.70%
	28	14737	15641	-5.78%
	29	20278	21939	-7.57%
	30	16022	17068	-6.13%
	Average	18016.7	19546.8	-7.83%

表 4-5 數值測試 - 本研究(GAR1 與 GAR2)之數值比較

# of ships	Problem instances	GAR2	SUBG	Compare
25	1	3331	3834	-13.12%
	2	3456	3504	-1.37%
	3	3002	3201	-6.22%
	4	2995	3160	-5.22%
	5	3661	3815	-4.04%
	6	3962	4205	-5.78%
	7	2569	2663	-3.53%
	8	3345	3486	-3.61%
	9	3694	3794	-2.64%
	10	2961	3267	-9.37%
	Average	3297.6	3492.9	-5.49%
75	11	7616	8229	-7.45%
	12	6340	7567	-16.22%
	13	5128	6118	-16.18%
	14	7943	7952	-0.11%
	15	7722	7701	0.27%
	16	6074	6557	-7.37%
	17	4890	5227	-6.45%
	18	7685	8520	-9.80%
	19	5772	6422	-10.12%
	20	5125	6043	-15.19%
	Average	6429.5	7033.6	-8.86%
150	21	24777	25068	-1.16%
	22	13014	15217	-14.48%
	23	18863	21436	-12.00%
	24	19436	21677	-10.34%
	25	23347	22199	5.17%
	26	17233	20095	-14.24%
	27	12460	15622	-20.24%
	28	14737	17291	-14.77%
	29	20614	19957	3.29%
	30	16022	18341	-12.64%
	Average	18050.3	19690.3	-9.14%

表 4-6 數值測試 - 本研究(GAR2)與 SUBG 之數值比較

另外，我們再就各種規模大小的船舶數量（25 艘、75 艘以及 150 艘）下，分別就 10 個例題的求解結果做整體性的品質比較。下表 4-7 為 GAR1 與 GAR2 在各規模的平均求解品質以及平均求解速度之資訊。

# of ships	GAR1		GAR2	
	Objective	CPU time	Objective	CPU time
25	3338.9	1.29 (sec)	3299.1	1.34 (sec)
50	6771.1	4.38 (sec)	6429.5	4.47 (sec)
150	19546.8	217.81(sec)	18099.6	229.77(sec)

表 4-7 數值測試 - 本研究(GAR1、GAR2)之求解效能比較

由表 4-7，我們可以確認各種型態的規模大小以及不同求解方法的運算時間仍然在可接受的範圍內；此外，也可發現 GAR2 相較 GAR1 確實能有效提升求解品質，但所額外花費的運算時間只有微幅上升。因此，求解機制並未因提升品質，而拖垮求解速度，依舊可以確保本研究為可行性之研究。



第五章 結論與建議

本研究歸納的結論如下:

1. 針對 DDBAP，本研究以 Imai *et al.* (2001) 做為比較基礎，成功導入啟發式解法-螞蟻演算法到此問題上。將演算法的求解機制納入船席指派的概念，包含船舶抵達時間的資訊，以及過去的求解經驗等都將一併納入考量。能有效提供給碼頭營運商船舶分派的優先順序以做為船席指派之用。另外，透過本研究自行開發的 GAR1 以及 GAR2，將螞蟻演算法提供的指派順序，轉化為相應的船席位置，碼頭營運者即可得知任一艘船須在何時、何處進行指派作業。
2. 進一步探討本研究的迭代效果分析，可發現在不同的試行結果之下，不僅末代的平均目標式值能顯著地低於首代的平均目標式值，而迭代之中的最小目標式值也同樣低於首代的最小目標式值。此結果說明螞蟻演算法能發揮功效，透過回饋機制，使得求解系統產生更為優化的求解次序，並強化求解品質。
3. 比較 GAR1 以及 GAR2 的求解效能，本研究透過規模大小相異的例題，進行比較。可發現在小範圍的例題中，GAR2 與 GAR1 的求解能力相去不遠。但在範疇更大的例題當中，GAR2 的尋優效果便可顯著地提升。一旦問題的複雜程度越高，更能彰顯 GAR2 的求解功效。
4. 本研究透過螞蟻演算法搭配 GAR2 的求解機制與 Imai *et al.* (2001) 進行比較，可發現無論問題的規模大小，本研究的求解能力都較為卓越。此舉顯示本研究在處理船席指派的問題上，的確有其代表性，也可供其他欲投入螞蟻演算法到船席指派之學者作為參考。

本研究提出的建議如下:

1. 本研究所探討的問題情境仍可再加入更多管理意涵，例如時間窗限制、船舶的權重因子等要件，可繼續研究螞蟻演算法在情境更為複雜下的求解能力。
2. 可進一步把螞蟻演算法的概念導入連續型的船席指派問題，觀察是否能有更好之結果可展現。

參考文獻

- Buhrkal, K., Zuglian, S., Ropke, S., Larsen, J., and Lusby, R. (2011), “Models for the discrete berth allocation problem: A computational comparison,” *Transportation Research Part E*, Vol. 47, pp. 461–473.
- Cordeau, J.-F., Laporte, G., Legato, P. and Moccia, L. (2005), “Models and tabu search heuristics for the berth allocation problem,” *Transportation Science*, Vol. 39, pp. 526-538.
- Cheong, C.Y., and Tan, K.C.(2008), “A multi-objective multi-colony ant algorithm for solving the berth allocation problem,” *Studies in Computational Intelligence (SCI)* 116, 333–350 .
- Christensen, C.G.. and Holst, C.T. (2008), *Berth Allocation in Container Terminals* (in Danish), Master Thesis, Department of Informatics and Mathematical Modeling, Technical University of Denmark, Denmark.
- Dorigo, M. (1992) *Optimization, Learning and Natural Algorithms* (in Italian), Ph.D. Dissertation, Dipartimento di Elettronica, Politecnico di Milano, Italy.
- Dorigo, M., Birattari, M. and Stutzle, T. (2006), “Ant colony optimization,” *IEEE Computational Intelligence Magazine*, Vol. 1, pp. 28-39.
- Dorigo, M. and Stutzle, T. (2004), *Ant Colony Optimization*, MIT Press, Cambridge, MA.
- Dorigo, M. and Gambardella, L.M. (1997), “Ant colony system - a cooperative learning approach to the traveling salesman problem,” *IEEE Transactions on Evolutionary Computation*, Vol. 1, pp. 53-66.
- Dorigo, M., Maniezzo, V. and Coloni, A. (1996), “Ant system: optimization by a colony of cooperating agents,” *IEEE Transactions on Systems, Man, and Cybernetics- Part B*, Vol. 26, pp. 29-41.
- Golias, M. M., Boile, M., and Theofanis, S., (2009), “Berth scheduling by customer service differentiation: A multi-objective approach,” *Transportation Research Part E*, Vol. 45, pp. 878–892.
- Golias, M. M., Boile, M., and Theofanis, S., (2010), “A lambda-optimal based heuristic for the berth scheduling problem,” *Transportation Research Part C*, Vol. 18, pp. 794–806.

- Hansen, P., Oguz, C. and Mladenovic, N. (2008), "Variable neighborhood search for minimum cost berth allocation." *European Journal of Operational Research*, Vol.131, pp. 636-649.
- Imai, A., Nishimura, E. and Papadimitriou, S. (2001), "The dynamic berth allocation problem for a container port," *Transportation Research Part B*, Vol. 35, pp. 401-417.
- Imai, A., Nishimura, E. and Papadimitriou, S. (2003), "Berth allocation with service priority," *Transportation Research Part B*, Vol. 37, pp. 437-457.
- Imai, A., Nishimura, E., Hattori, M. and Papadimitriou, S. (2007), "Berth allocation at indented berths for mega-containerships," *European Journal of Operation Research*, Vol. 179, 579-593
- Imai, A., Zhang, J. Nishimura, E., and Papadimitriou, S. (2007), "The berth allocation problem with service time and delay time objectives," *Maritime Economics and Logistics*, Vol. 9, pp. 269–290..
- Imai, A., Nishimura, E., and Papadimitriou, S. (2008), "Berthing ships at a multi-user container terminal with a limited quay capacity," *Transportation Research Part E*, Vol. 44, pp. 136-151.
- Monaco, M.F. and Samara, M. (2007), "The berth allocation problem: A strong formulation solved by a Lagrangian Approach," *Transportation Science*, Vol. 41, pp. 265-280.
- Mullen, R. J., Monekosso, D., Barman, S. and Remagnino, P. (2009), "A review of ant algorithms," *Expert Systems with Applications*, Vol. 36, pp. 9608-9617.
- Nishimura, E., Imai, A., and Papadimitriou, S. (2001), "Berth allocation planning in the public berth system by genetic algorithms," *European Journal of Operational Research*, Vol. 131, pp. 282-292.
- Stutzle, T. and Hoos, H. H. (1997), "The MAX-MIN Ant System and local search for the traveling salesman problem," in Back, T. *et al.*, (eds.), *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation (ICEC'97)*, IEEE Press, Piscataway, NJ, pp. 309–314
- Theofanis, S., Boile, M., Golias, M. M. (2009), "Container terminal berth planning - Critical review of research approaches and practical challenges," *Transportation Research Record*, Vol. 2100, pp. 22-28.
- Tong, C. J., Lau, H. C., and Lim, A. (1999), "Ant Colony Optimization for theShip

Berthing Problem,” Asian Computer Science Conference , pp. 359-370.



簡歷



中文姓名：陳儀安

英文姓名：Yi-An Chen

出生日期：民國 74 年 11 月 4 日

聯絡地址：桃園市經國路 555 號 5 樓之 1

聯絡電話：0939-672216

E-mail : yian41@gmail.com



簡歷：

民國 101 年 7 月 國立交通大學 運輸科技暨管理學系 碩士班畢業

民國 97 年 6 月 國立中正大學 財務金融學系 畢業

民國 93 年 6 月 國立桃園高級中學 畢業

民國 90 年 6 月 桃園縣立慈文國民中學 畢業

民國 87 年 6 月 桃園縣立北門國民小學 畢業