

# 國立交通大學

交通運輸研究所

碩士論文



高雄港貨櫃起重機司機員  
排班作業之研究

A Study on Scheduling of Container Crane  
Operators in the Port of Kaohsiung

研究生：林羅耀

指導教授：黃承傳 教授

中華民國九十七年六月

高雄港貨櫃起重機司機員排班作業之研究

A Study on Scheduling of Container Crane Operators in the Port of  
Kaohsiung

研究生：林邏耀

Student : Luo-Yau Lin

指導教授：黃承傳

Advisor : Dr. Cheng-Chwan Hwang

國立交通大學



A Thesis

Submitted to Institute of Traffic and Transportation

College of Management

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of Master Engineering

in

Traffic and Transportation

June 2008

Taipei, Taiwan, Republic of China

中華民國九十七年六月

# 高雄港貨櫃起重機司機員排班作業之研究

研究生:林邏耀

指導教授:黃承傳

國立交通大學交通運輸研究所

## 摘要

橋式起重機為貨櫃碼頭之主要裝卸機具，而操作起重機裝卸作業人員則為貨櫃碼頭貨物裝卸流程中最主要的作業人員，其排班方式由於船舶靠港時間、裝卸量及其他不確定因素的影響，使得司機員之指派作業有別於一般企業的人員排班。我國國際港埠之貨櫃碼頭目前對於司機員之排班大都仍採取人工經驗法則排班，此種方式除了比較耗費時間之外，所排出的班表亦有若干缺點及待改進之處。

本研究首先針對貨櫃碼頭中司機人員之排班問題深入了解其作業現況及目前排班方式之缺失及問題，並且依據現況人工排班之經驗法則與公司排班規則，利用數學規劃方法進行模式之構建與求解，再依據所構建之模式進行電腦程式之設計，除可代替人工作業，節省排班時間外，並可求解出不同目標式與司機員人數之排班結果

本研究並以國內 E 航運公司為個案，依據其實際排班相關規則，50 名司機員及一週船期之排班資料作為測試案例，應用套裝軟體 ILOG OPL Studio 求解。研究結果顯示，數學規劃排班模式除可代替現行的人工指派方式，提升排班效率外，亦可大幅改善目前勞逸不均之情形。而偏好休假之模式更可在以兼顧業主及勞工意願之原則下完成排班，提供另一種排班的參考方式。

關鍵字: 排班、橋式起重機、整數規劃、最適化

# A Study on Scheduling of Container Crane Operators in the Port of Kaohsiung

Student : Luo-Yau Lin

Advisor : Dr. Cheng-Chwan Hwang

Institute Traffic and Transportation  
National Chiao Tung University

## ABSTRACT

Gantry crane is the main equipment of loading and unloading containers in seaport terminals, and the quay crane operators are the main workforce of container terminals. The characteristics of gantry crane operators scheduling are different from those of other industries due to uncertain mooring time of ships and inconsistent loading and unloading amount of containers. The Container Terminal of most international ports in Taiwan at present use manual scheduling based on some rule of thumb, it will be in addition to waste time on this method, the result of scheduling also have some defects needs to improve.

In this search, we first study the characteristics of gantry crane operators scheduling, and problems of present scheduling operators, then in accordance with the scheduling of existing manual rule of thumb and company's rules, use Mathematics Programming to formulate a scheduling model. According to the model, then we design the computer program to solve it. It can not only replace manual operation and save scheduling time, but also resolve the result of different number of operators.

The model is applied to a case study of a shipping company. Using real world data of 50 operators and the Shipping Schedule of a week, obtained from the studied company and solved by the commercial optimization software OPL. The results show that the model can either replace current manual scheduling to increase scheduling efficiency, and improve the labors aniso-average. Preferences of rest model which consider both sides of proprietor and labors to provides many useful information for manpower planning and management for shipping company.

Keywords : Scheduling, Gantry Crane, Integer Programming, Optimization

## 致謝

漫長的學生生涯終於在鳳凰花開來臨的季節，即將寫下句點。看著就要修訂完成的論文，心中百感交集。幾年前的我，還沒意識到自己有一天將會寫著論文致謝詞，此時的我實在五味雜陳，想到這一路上走來，有太多要感謝的人，有太多言語難以言喻。

本碩士論文得以完成，首先感謝我的指導老師黃承傳教授，從老師身上學習到修養品格，以及論文上悉心的指導，提供明確的方向及建議。再來是感謝口試委員吳榮貴教授及梁金樹教授百忙之中撥空給予寶貴的意見，使得論文更加完善。在程式的部分，必須特別感謝我的高中同學昱龍，即使在繁忙中仍耐心地不停抽空幫忙，以及工工所博班一平學長出借軟體以及教導修正程式參數的部分。還有最後初稿以及口試部分這段時間裡，所有同學們互相鼓勵加油，這些都是最珍貴的片段，感謝大家這一路的扶持。

最後必須感謝我的家人們，從大學時期至現在研究所畢業這段期間，不斷地給我支持，在我受挫折時賦予我更多去面對的力量。還有我的好朋友昱龍及士錚一路的陪伴，總是在我會低潮時給我最多的包容與關懷，若沒有你們，或許早已經半途而廢了。

這個句點，劃下了另一個開始，邁出這一步之後有更多的挑戰，我期望自己能夠帶著感恩的心以及你們的陪伴，創造更璀璨的未來。

林邏耀 謹誌

記于 2008.6 台北交大

# 目錄

摘要 .....	I
英文摘要 .....	II
致謝 .....	III
目錄 .....	IV
圖目錄 .....	VII
表目錄 .....	VIII
第一章 緒論 .....	1
1.1 研究動機 .....	1
1.2 研究目的 .....	1
1.3 研究範圍與限制 .....	2
1.4 研究內容與方法 .....	3
1.5 研究流程 .....	4
第二章 文獻回顧 .....	6
2.1 人員排班問題 .....	6
2.1.1 人員排班問題定義 .....	6
2.1.2 人員排班問題型態 .....	7
2.2 人員排班之解法 .....	8
2.2.1. 最佳化演算法 (Optimal Solution Algorithm) .....	9
2.2.2. 啟發式演算法 (Heuristic Algorithm) .....	11
2.2.3. 限制規劃法 (Constraint Programming) .....	12
2.2.4. 人員排班各種解法之比較 .....	13

2.3 限制式規劃.....	14
2.4 運輸業的排班規劃.....	16
2.4.1 海運業 .....	17
2.4.2 陸運業 .....	17
2.4.3 空運業 .....	18
2.5 小結.....	21
第三章 現況分析與檢討 .....	22
3.1 現況之作業環境.....	22
3.2 現況之作業流程.....	22
3.3 橋式機人員指派及班表規劃考量之現況.....	23
3.4 值勤班次時間分配.....	24
3.5 實務上作業指派方式及原則.....	25
3.6 現況排班結果與檢討.....	29
3.7 小結.....	33
第四章 模式構建 .....	34
4.1 限制式規劃與整數規劃.....	34
4.2 限制式規劃之求解.....	34
4.2.1 一致性技巧 .....	34
4.2.2 樹狀搜尋演算法 .....	35
4.3 最佳化問題之限制式規劃求解概念.....	36
4.4 最佳化求解軟體 .....	38
4.4.1 CPLEX .....	38
4.4.2 ILOG OPL .....	39
4.4.3 AMPL CPLEX.....	39



4.4.5 小結 .....	39
4.5 排班邏輯修正 .....	40
4.6 休假方式之重新擬定 .....	40
4.7 數學模式建構 .....	41
第五章 實證分析 .....	45
5.1 測試及輸入資料背景 .....	45
5.2 作業司機員人數變動之敏感度分析 .....	55
5.3 連休排班模式 .....	57
5.4 小結 .....	60
第六章 結論與建議 .....	61
6.1 結論 .....	61
6.2 建議 .....	62
參考文獻 .....	63
附錄一、C 語言程式 .....	66





## 圖目錄

圖 1-1 研究流程圖 .....	5
圖 3-1 作業人員的排班規劃 .....	24
圖 4-1 一致性技巧之流程圖 .....	35
圖 4-2 標準搜尋程序流程圖 .....	37
圖 4-3 二分搜尋法之流程圖 .....	38



## 表目錄

表 2.1 解算法之比較表 .....	13
表 2.2 限制式規劃之發展情形 .....	15
表 3.1 貨櫃碼頭資料 .....	22
表 3.2 E 航運公司橋式起重機司機員工作班次 .....	25
表 3.3 實務作業指派第一階段 .....	25
表 3.4 實務作業指派第二階段 .....	26
表 3.5 實務作業指派第三階段 .....	27
表 3.6 船舶到港裝卸貨櫃之資料 .....	30
表 3.7 E 航運公司之司機員班表 (2007 年 12 月 2 日至 12 月 8 日) .....	32
表 3.8 各司機員值勤班次數(範例) .....	33
表 4.1 司機員休假表 .....	41
表 5.1 一周司機員人力需求表 .....	46
表 5.3 司機員一周總值勤班次數 .....	52
表 5.4 數學規劃及原始班表之比較 .....	53
表 5.5 每位司機員一周總值勤時間(小時) .....	54
表 5.6 45 位司機員人數之一週值勤班次數 .....	55
表 5.7 40 位司機員人數之一週值勤班次數 .....	56
表 5.8 35 位司機員人數之一週值勤班次數 .....	56
表 5.9 司機員人數之敏感度分析 .....	57

# 第一章 緒論

## 1.1 研究動機

在全球國際貿易發達的現代，大多國家發展國際貿易以促進經濟成長。目前全球大部份的貨物必須透過海運運輸，而國際貿易運輸這種先進的現代化運輸方式立即在全球運輸業中得到了廣泛應用，進而貨櫃化運輸成為當前海運運輸業市場的主流。

在國際港埠貨櫃裝卸搬運的過程中，基於效率及安全上的考量，除了需要足夠的機具以及港埠硬體設施外，完善的排班規劃以及足夠的作業人數也是不可或缺的一環。我國國際港埠在裝卸作業民營化後，機具及硬體設施受制於港務局及法令問題，不易於短期之內變動，而排班及人員數部分則可由承攬公司自行規劃。

目前國內裝卸承攬公司對於橋式機作業人員之排班部分多採用人工排班，此種以經驗法則來處理班表指派的方式，不僅浪費時間與精力，且排出的班表仍有許多缺點及可改進之處。人工排班常見的問題之一係由於船舶裝卸量不同、船舶進港靠泊時間不固定及工作負荷量之因素，造成同一天中值勤的班次需間隔八、九個小時以上，使得作業人員自我時間安排上非常的麻煩。此外，如果人員數不足，每週之工作時數常會超過勞基法規定四十八小時，且時常一天內連續值勤多個班次，工作之安全度因高度專注且連續執勤而大幅減低。

為了提高人員作業效率並且盡量避免人員連續值勤過多之班次，必須有一套合理的排班系統。因此引發本研究針對橋式機作業人員之排班問題進行深入探討之動機，期能有助於改善目前之排班作業。

## 1.2 研究目的

本文主要針對高雄港之橋式機作業人員排班問題進行探討，其主要目的如下：

1. 了解作業現況及檢討目前排班之缺失或問題。
2. 利用數學規劃方法建構模式，並且求解出最適的排班結果。

3. 針對目前排班方式的缺失提出改善的方法及建議。

### 1.3 研究範圍與限制

高雄港港灣自然條件優良，港域廣闊、腹地廣大，全年氣候條件溫和，臨海附近有狹長沙洲之天然屏障。由於地理條件優越，港灣形勢天成，自建港以來，即在海運運輸上，扮演重要之角色，目前為我國最大之國際港埠，並位居世界貨櫃進出之第五大貨櫃港。高雄港港區面積為 17,678 公頃，其中陸域面積 1,442 公頃，佔全港面積之 8.2%，水域面 16,236 公頃，佔全港面積之 91.8%，港區配置以碼頭作業區為主，其次為工業區，其餘則為港務行政、漁港、造船廠、台電、中油等用地。現有碼頭 118 座，全長 26,595 公尺，繫船浮筒 22 組，同時可供 155 艘船靠泊。貨櫃碼頭總共有 23 座，每年裝卸總作業能量達 1265 萬 TEU。

高雄港貨櫃碼頭之營運方式除了少數公營之外，主要係以長期出租方式，分別由多家航商租用。由於不同的航運公司及各裝卸承攬公司司機作業人員排班方式不盡相同，因此在排班方面所面臨的問題也有所差異。考慮此因素，本研究選擇以國內較大的海運公司-E海運（以下簡稱E公司）所租用的79號碼頭、80號碼頭、81號碼頭三座連續碼頭為研究範圍。（因公司要求保密，故公司名稱不便透露）

在人員排班方面，由於該公司已事先將人員休假規劃成一循環性，所有休假人員均已排定，因此，本研究並不對此問題加以探討，只針對人員值勤排班問題做規劃。在作業人員方面，貨櫃碼頭之工作人員包含理貨人員、拖車司機、橋式機司機員及地勤人員等。理貨人員負責船舶裝貨或卸貨時，對貨物的件數進行清點，並對貨物的交接作出證明。拖車司機負責駕駛拖車，協助貨櫃之裝卸與提領。橋式機司機員與地勤人員為橋式機之主要作業人員。司機員為橋式機駕駛，處理船舶與岸上貨櫃之抓吊作業。地勤人員

為貨櫃裝卸作業中，在船上及岸上協助指揮橋式機司機員及拖車司機，以使每個裝卸動作能夠有效及安全的完成。本研究之研究對象只針對橋式機操作司機員，而不包含其他的作業人員。

此外，人員指派作業需考量船舶進/出港時間、裝卸量、橋式機數目、碼頭數及人員熟練度等，由於橋式機數目及碼頭數短期間不易改變，且船舶進/離港時間、裝卸量、人員熟練度為不易掌握之因素，因此本研究假設橋式機數目、碼頭數及人員熟練度為固定，船舶進/出港時間及裝卸量為已知條件下，針對高雄港長榮海運三座碼頭之橋式機作業人員排班問題作一探討。

## 1.4 研究內容與方法

本論文之研究方法主要先以現場訪查及資料收集方式，充分了解該公司作業現況、人員排班特性及排班規則。在了解實際作業及現況後，收集原始之班表做為投入模式求解的基本資料集。其次，為了有效率提高班表指派之完整性且使人員盡量避免連續值勤過多之班次，並在考慮所有法規限制條件下，本研究以數學規劃方法構建人員排班模式，並以C 語言程式撰寫輸入限制式入檔產生器，ILOG CPLEX 10.0 數學規劃軟體求解。

### 1. 問題界定

國內對於港埠作業之橋式機作業人員工作特性及排班問題並沒有過完整探討，也由於沒有一套有效的分析工具，因此在實務上對於員工指派作業產生之缺點，也無法解決，因此本論文著眼於此一問題之研究，希望希望能夠有所貢獻。

### 2. 文獻回顧

蒐集有關人員排班問題之相關文獻，並加以分類整理歸納，包括不同的人員排班問

題的定義與相關應用等，以便瞭解橋式機作業人員排班問題之特性。

### 3. 現況分析與檢討

藉由訪談及相關資料蒐集瞭解 E公司在高雄港所租用三座貨櫃碼頭目前的硬體設施、橋式機數量及目前人員排班之作業現況方法與其所面臨的問題。

### 4. 數學模式構建

本研究將人員排班問題定式為一最佳化整數規劃問題，在司機員排班模式上，其目標函數為求得總值勤班次數平均化，並依據該公司班表編排之規定進行限制式之建構。排班指派模式裡，其目標函數為實際作業人員數最小化與總值勤班次數平均化，並依據該公司之排班規定及班表有待改善之處等進行限制式之建構。

### 5. 模式求解與測試

此階段依據蒐集之資料測試司機員及地勤人員排班模式之可行性，並觀察是否與實務單位模式所產生之班表相同。在司機員方面，並測試司機員之經濟人數，以瞭解組數改變對總值勤班次數之影響。

### 6. 實證分析

利用求解出來之值勤班表，進行求解品質與求解效率之分析，並與實際上人工排班的情形做比較，進而探討本研究模式之可行性與優缺點，作為模式調整之依據。

### 7. 結論與建議

綜合本論文之研究測試之結果提出主要結論與未來研究方向之建議。

## 1.5 研究流程

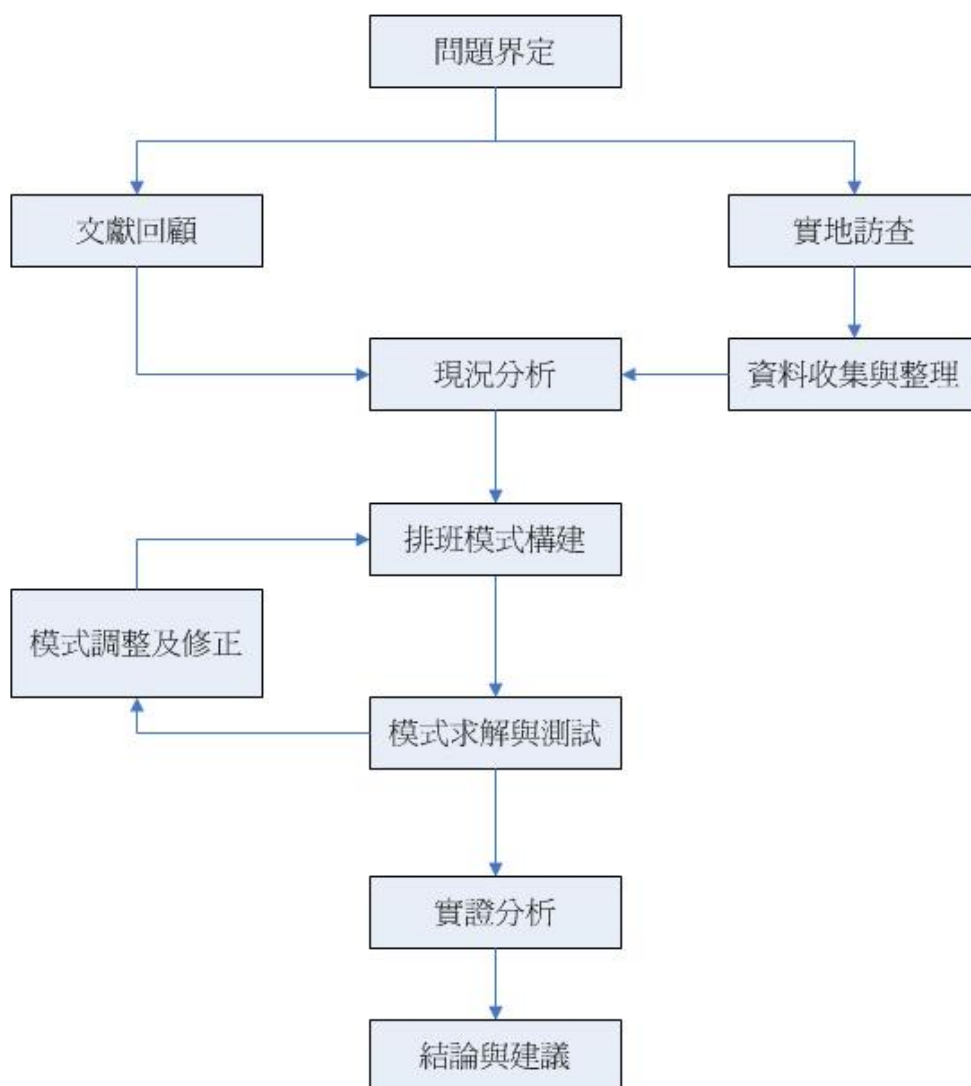


圖1-1 研究流程圖



## 第二章 文獻回顧

### 2.1 人員排班問題

約自1970年起，許多學者開始著手研究討論人力資源規劃(Work Force Planning)與人員排班(Staff Scheduling)的相關問題。投入這些研究的共通點都是為了讓公司企業的雇主在以滿足雇用人員數的需求量、符合各種排班需求要求及法令規定勞工時數下，以排班規劃方法來達到僱用人數最少，亦即達到成本最小、效用最高之目的。但由於經濟水準的提高以及近年來追求生活品質的觀念抬頭，員工對生活的品質價值越加重視，進而要求工作及休假時間上的安排，尤其是例假日是否可以放假、連續假日的需求及排班的公平性等。

#### 2.1.1 人員排班問題定義

人員排班問題是真實世界中常面臨到的一個問題。大至政府單位、公司組織，小至一般團體、班級間的人力工作安排，簡單的來說就是在適當的「時間」安排適當的「人力」於適當的「工作」上。

Lau 【29】針對人力排班做了以下的定義：人力排班為滿足管理者、勞方、政府等各單位的目標與政策下，將人力資源適當的安排於所需的作業項目。意即在企業營運中，將員工妥善安排至各項工作以提供服務，已成為一項不可或缺的管理活動。

Abboud et al. 【14】認為真實世界中的人力指派問題中尚有很多要素需要考量，例如個別勞工的工作能力、偏好及滿意程度等均影響指派作業的好壞。一般而言，實際人力排班情形並不易解決，故進而發展出兩種方法，第一種是發展數學分析模式來獲取最佳解；第二種作法為採用啟發式解法來得到近似最佳解。



## 2.1.2 人員排班問題型態

不同的排班問題有各種不同解題型態，人員排班的分類型態以下列說明：

1. 休假及值班的排班問題可分為下列三種排班方式：

### (1). 值勤班次排班問題( Shift Scheduling Problem )

此問題僅規劃一天內的班表，意即只要決定人員在這一天內那個時段必須工作。最簡單的方式是指派非重疊性班次(Nonoverlapping shifts)，如早、中、晚三班。但是缺點為當需求隨著時間而變化很大時，非重疊班次雖然能夠滿足尖峰時段的需求，但在非尖峰時段會產生很大的人力浪費。為了改善這種情形，於是就有重疊性排班(Overlapping shift)的產生。如此一來，班次型態就更多，增加了問題的複雜度。

### (2). 休假排班問題( Days-off Scheduling Problem )

意即以休假為優先安排，有別於上述以排班優先的方式。休假班表的規劃長度通常為一個星期至一個月，且每一天人員的需求已知。人員每一星期的工作天數必小於企業單位一星期營運的天數時，因此必須安排人員的休假日。而當休假日一旦決定，即可知道人員之工作日。休假的型態有很多種，例如一星期中有一天休假或兩天休假，後者又可分為連續或不連續假，至於休假是何種型態，則視營運單位法規而定。

### (3). 休假暨值勤排班問題( Tour Scheduling Problem )

此排班問題的規劃是為綜合上述兩種方式。通常規劃的長度週期亦為一個星期至一個月。所指派的班表不只包含指明何日為休假日，也包含在值勤日的工作時段。換言之，執勤班次排班問題及休假排班問題是為此排班問題的分支。此規劃問題使得指派班次的型態更加龐大，問題求解更加複雜。

目前大多企業公司，包括該裝卸承攬公司皆是採用上述第三種排班規劃方式，在上

個周期班次中期至末期，就先排定下一值班及休假的班表。而指派班表時間週期及長度端看公司規模大小、排入班表的員工數多寡以及內部調和因素而決定。

2.人員排班可分為下列兩種排班型態：

(1). 週期性排班(Cyclical Scheduling)

週期性排班為制定固定班型，每4-6 週或7-12 週循環，優點為較為公平，因為每個人都會循環輪班，所以都會輪到好班或較差的班。另外由於事先人員知其工作及休假時間，因此人員易於安排個人休閒及社交活動長短程之計劃。

(2). 非週期性排班(Non-cyclical Scheduling)

排班人員依照公司所給的人員數，排定該單位之班表，可依人員數改變而修改排班表。其優點為排班較為彈性，若公司有突如其來的行銷活動或有意安排某些人員進行進修課程活動時，亦可讓排班人員事先得知；也可讓人員事先提出申請來安排較長的假期等優點。



本論文之岸肩起重機司機員排班模式的問題複雜度相當高，因為船舶到港時間不一致、裝卸貨櫃量大小、工作人員數多寡及人員可能臨時休假等因素，需要極佳的排班彈性才可產生完善的班表，因此本模式在排班方式為休假執勤排班問題，而排班型態是屬於非週期性排班。

## 2.2人員排班之解法

關於人員排班問題，在國內外均有廣泛的討論，相關文獻頗豐，其在學術領域被定義為組合最佳化問題。在過去國內外學者的研究下，已發展出多種求解概念與方法，基本上求解方式可簡單分為兩派，一為最佳化演算法(Optimal Solution Algorithm)，另一為

啟發式演算法(Heuristic Algorithm)。

### 2.2.1. 最佳化演算法 (Optimal Solution Algorithm)

最佳化演算法是在求解條件與限制式已知條件下，對於問題的目標求取其可行解空間內最佳解。然而此法最主要面臨的瓶頸為演算時間較長，對於大型人員排班問題往往需要耗費過多的求解時間。

基於最佳演算法無法在特定時間內求得可行解的原因，一般對於實際或大型人員排班問題的處理會選擇設計適合的啟發式演算法以尋求一可行解，雖然不盡然為最佳解，但綜合時間上的效用，縮短時間所換取的求解效率通常更符合實務上所需。企業所面臨的人力需求隨著時間不斷改變，因此對執勤人員的需求亦隨著時間而有所不同，因此可將整個排班期分為數個時段，根據顧客到達率、服務時間及特定的服務水準，得知各時段人員需求數，來進行排班工作。在滿足人員需求水準及法規的限制下，以最低成本為目標，決定值勤人數及其值勤時間。最常應用的方法便是 Dantzig【24】年提出的數學規劃式，表示如下：

$$\begin{aligned} \text{Min} \quad & \sum_j C_j X_j \\ \text{s.t.} \quad & \sum_j a_{ij} X_j \geq d_i \quad i=1,2,\dots \\ & X_j \geq 0 \\ & X \in \text{integer} \end{aligned}$$

$i$  : 時段區間指標 (Interval Index),  $1 \leq i \leq I$

$j$  : 值勤班次指標 (Shift Index), 表類型  $j$  值勤班次

$X_j$  : 值勤班次  $j$  之班次數, 為決策變數

$C_j$  : 值勤班次  $j$  之單位成本

$d_i$  : 值勤  $i$  之人員需求

$a_{ij} : \begin{cases} 1, \text{若時段 } i \text{ 為班次 } j \text{ 之工作時段} \\ 0, \text{其他} \end{cases}$

而此模式在1997年由Chu【22】重新提出修正數學模式表示如下:

$$\min \sum_{j \in M} c_j x_j$$

$$\text{ST} \quad \sum_{j \in M} a_{ij} x_j = 1 \quad \forall i \in N$$

$$x_j = 0,1 \quad \forall j \in M$$

$c_j$ : 值勤班次  $j$  的單位成本

$M$ : 值勤的班次集合

$N$ : 時段的集合

$a_{ij}$ : 若是 1 表示時段  $i$  為班次  $j$  之工作時段, 否則為 0

$x_j$ : 若是 1 表示執勤班次  $j$  有執行, 否則為 0

在處理 0-1 整數規劃的人員排班問題上, 以分支定限法(Branch and Bound Method) 為代表。另外還有分支切面法 (Branch-and-Cut Method) 及變數產生法(column generation)。茲分別說明如下:

### 1. 分支定限法 (Branch and Bound Method)

分支定限法是最常來解決最佳化問題的演算法, 在Hiller and Liebermanru【26】分支定限法是以分割以及解決的觀念。

以Beasley and Cao【18】在1996年所發表的論文為例, 利用樹狀搜尋法 (Tree Search

Algorithm) 也就是利用分支定限法來求解最小成本人員排班問題。若在演算過程中獲得整數可行解，則可知其即為模式最佳解。即使此部分之解無法獲得最佳解，亦可由其目標值來決定下一階段求解之目標值下界。此演算法大多能在合理的時間內得到較簡單問題的的最佳解。

但人員排班問題也被 Bartholdi 【17】證明為NP-Complete，意即要在合理時間內，對於較龐大且複雜的問題找到一個快速求得最佳解的方法，是極不可能的。

## 2. 分支切面法 (Branch-and-Cut Method)

以Hoffman and Padberg 【27】於1993年所提出之分支切面法為例，作者先將複雜的人員排班模式利用簡單的數學方法，來去除不必要的變數以及限制式以簡化模式。而有別於傳統的分支定限法，其在分支定限法中加入一啟發式搜尋法，以快速獲得一整數可行解，並以此方式不斷更新模式之上界。

## 3. 變數產生法 (column generation)

變數產生法(column generation)或是稱為Dantzig - Wolfe Decomposition，主要是利用線性規劃中的對偶理論(Dual theory)來產生變數(column)，以避免浪費不必要的時間去窮舉不可行(或對問題求解沒有貢獻)的變數。此法的優點在於求解過程中一次只考慮部份的變數，並利用主、子問題間的訊息傳遞，逐步得到最佳解，如此將可增進求解效率，並可避免因變數過多而導致無法求解。最早變數產生法最早用於排班問題上，則可追溯至 1969 年由Appelgren 【16】提出應用在船舶排班問題上。

### 2.2.2. 啟發式演算法 (Heuristic Algorithm)

關於人員排班常見的啟發式解法有線上交談法、批次產生法、基因演算法(Genetic Algorithm, GA)、禁制搜尋法(Tabu Search, TA)以及模擬退火法(Simulated Annealing, SA)等人員排班之啟發演算法方法頗多，茲舉例如下：

1. Smith 【31】利用啟發式解法，將人員排班設計成批次模式作業，分為三個階段：首先為總和各單位每週用人狀況，再產生暫時的班表，並顯示各單位人力的多餘或短缺；最後再手動調整，產生最後的班表。
2. 高建元【7】採用線上交談方式協助護理行政人員完成排班工作，其考量項目有：排班作業的起始日期、本月參與排班的人數，每個人員職別的區分，每個人員本月班別，每個人員上月末派班狀況以及類計的休假數紀錄，以及人員預約休假資料與人力需求等。
3. 陳豪雷【5】提出利用多重時空網路定義人旅次的流動，並透過系統最佳化，求得最佳運送旅客的方式下，車隊的排程與班表的設定。整合一旅次需求與班次供給之車隊排程模式，並進一步的發展拉氏啟發式演算法，以幫助業者規劃最佳的車隊排程與班次表。

### 2.2.3. 限制規劃法 (Constraint Programming)

目前限制規劃應用在人員排班上的文獻大多為探討護士排班問題，許多國家的學者利用限制規劃求解護士排班問題，並實際作出決策支援系統，已有提供醫療單位極佳之功效。

Chun et al. 【23】，運用限制規劃方法為香港醫院管理局(Hospital Authority, HA)開發一項護理人員派遣模式(Rostering Engine, RE)，並將此模組放入香港醫院管理局之人員派遣系統中。

Cheng et al. 【20】，運用 "Redundant Modeling" 之模式技巧將香港鄧肇堅醫院急診部之護士人員排班問題模式化成兩個具有相等意義之限制滿足問題模式，且加入相連此兩模式之溝通限制式(Channeling Constraints)，藉由兩個模式之相互運作影響，增加模式之限制式繁衍 (Constraint Propagation)能力，以加快限制規劃方法之求解速度。其測試的問題範圍為「25-28 位護士與 11 種值勤班次型態(shift types)」，由於此系統之運作績效相當良好且具有彈性，使得醫院當局已開始採用此系統來輔助排班人員。



#### 2.2.4. 人員排班各種解法之比較

有關人員指派的問題，在一般運輸系統中一直為重要的研究問題。其中有些文獻著重於營運路線班次的排程，有些著重於班次確定後之車輛及人員的排班排程。文獻上通常將人員排班的問題以 0-1 整數規劃問題來描述。目前有關人員排班的問題，多是利用能在合理的時間內找到較優可行解的啟發式演算法的解法為主。近年來由於電腦的技術不斷進步，使得一些不具即時性的問題，也可考慮以最佳化的演算法來求解。以下將針對人員排班的種類與排班問題作一介紹。

人員排班問題主要是在滿足人員的需求水準及法令的限制下，以最低成本為目標，決定值勤人員及其值勤時間。

表 2.1 解算法之比較表

演算法	可求解問題規模	求解彈性
最佳化演算法	可求解問題規模較小，若問題規模較大求解則需花費的時間越長	一旦模式訂定，若改變問題條件，模式修改不易
啟發式演算法	可求解問題規模較大，在合理的時間內可求出近似最佳解	一旦模式訂定，若改變問題條件，模式修改不易
限制規劃法	求解模式可大可小，可在合理的時間內求出多組可行解提供排班之參考	限制規劃模式的組合較容易修改變動

## 2.3 限制式規劃

因本研究之人員規劃方法亦為限制式規劃中之混和整數規劃，因此，本節整理出關於限制式規劃之發展背景以及相關文獻，以利往後模式構建。

限制式規劃之發展亦可追溯自 1970 年起，滿足限制式問題由人工智慧領域及人力資源相關之學者開創及定義。然而早期受限於電腦普及性及硬體配備等級較低，實務上規劃問題之求解仍然愈到相當大的瓶頸及困難。

至 1980 年代起，電腦革命量產後帶動限制式滿足問題的研究風潮，當時邏輯語言大量使用於求解滿足限制式問題。早期的限制式規劃問題主要利用 (Local Programming) 的概念求解，由於當時受限於演算法的缺點，限制式規劃之發展並未成熟。

直到 1990 年開始，有人著手試圖利用一般程式語言 (如 C++ 撰寫成 ILOG Solver) 取代原有的邏輯語言，使用限制式規劃的門檻也較以往低，故普及性大大提升。另外，許多研討會開始定期以限制式規劃為主題進行研究相關討論，亦使得限制式規劃之研究領域迅速擴大且普及化。

ILOG OPL STUDIO 【13】針對限制式規劃是由滿足限制式問題 (Constraint Satisfaction Problem, CSP) 所衍生而成，因此在瞭解限制式規劃之前，必須先充分了解滿足限制式問題之意義及內涵。

陳柏榮與王國琛【8】於滿足限制式問題即為考慮當一個問題包含變數  $X_1 \dots X_n$ 。此類型問題即稱為滿足限制式問題。而此求解方式在滿足所有限制式之條件下，是否存在可行解。總體而言，問題之變數數目或限制式數目較多時，必須常使用電腦程式語言軟



體來輔助求解。若使用邏輯語言(如 Prolog)則稱為限制式邏輯規劃(Constraint Logic Programming,CLP);而非邏輯概念類型,使用一般程式語言輔助求解之滿足限制式問題時稱為限制式規劃(Constraint Programming,CP)。

表 2.2 限制式規劃之發展情形

時間	問題產生	使用之軟體	發展情形	普遍性
1970	人力資源規劃問題產生	尚無	著手研究 CSP	低
1980	排班問題及演算法研究漸漸受到重視	邏輯語言(Prolog)	<ul style="list-style-type: none"> <li>➢ 演算法萌起</li> <li>➢ 研究人員增加</li> <li>➢ 使用門檻高</li> </ul>	中
1990	各類排班及規劃問題普遍,演算方式趨於成熟	<ul style="list-style-type: none"> <li>➢ 邏輯語言</li> <li>➢ 一般程式語言撰寫之套裝軟體 (C++,ILOG)</li> </ul>	<ul style="list-style-type: none"> <li>➢ 演算法效率高</li> <li>➢ 電腦及相關求解軟體普及</li> <li>➢ 定期研討會,相關研究趨多</li> </ul>	高

利用限制式描述問題是限制式規劃中關鍵的一環。一般而言,限制式之描述方式並不限定某種標準格式,在模式構建中可依循問題之需要類型進而宣告變數撰寫限制式,限制式撰寫方法大致如下:

1. 值域限制式 (Domain Constraint)

值域限制式主要定義模式中所有變數的值域,例如 $Y \in \{1,2,3,4\}$ ,即宣告變數Y之值域為整數且其範圍為1,2,3,4。

## 2. 算數限制式 (Arithmetic constraint)

算數限制式包含等式 (=) 限制式與不等式 (=、≠、≥、≤、>、<) 限制式，其意義與數學規劃模式之限制式相同。運算子 (=、≠、≥、≤、>、<) 左右兩邊算是以數字變數或參數組合。以等式限制式  $W=X+Y+Z$  為例，當左式  $W$  值等於右式  $X+Y+Z$  時，表示變數值滿足限制式。

## 3. 符號限制式 (Symbolic Constraint)

在符號限制式中，所有變數類型為符號形變數，此類型限制式通稱為符號限制式。例如塗色問題 (典型滿足限制式之問題) 中，相鄰區塊顏色必須相異，如區塊 A 之值域範圍為{黃、綠、紅}，區塊 B 值域範圍亦為{黃、綠、紅}，依題目種類需要，當區塊 A 與區塊 B 相鄰時，模式中必須包含限制式  $A \neq B$ 。主類型限制式為符號限制式。

## 4. 邏輯限制式 (Logical Constraint)

邏輯限制式在限制式中使用電腦邏輯語言之邏輯運算子來描述不易以數學符號描述的邏輯性限制式。

## 5. 布林限制式 (Boolean Constraint)

布林限制式是表示限制式當中變數值域都是屬於布林 (Boolean) 的形式，意即所有變數之值只有真或假兩種。

## 6. 全域限制式 (Global Constraint)

全域限制式為事先定義之限制式，如 ILOG Solver 中提供之全域限制式有 alldifferent、sequence、circuit、distribute 等四種。

## 2.4 運輸業的排班規劃

本論文在橋式起重機司機員排班求解問題所應用的研究規劃方法，近年在國外文獻中已有文獻開始探討。本節將近年來有關各種運輸業之相關文獻作一整理及深入探討。

#### 2.4.1 海運業

Kim and Kim 【28】 探討港口碼頭貨櫃機具司機人員的排班模式，本文將其定式為一限制滿足的問題。由於港口碼頭司機人員的排班需考量員工的工作時間外，還需考量設備是否足夠，再加上有多種類的機器設備，將使得排班有一定之難度。因以上之因素，港口碼頭司機人員與一般人員排班問題相差頗大，值得深入探討。其探討的限制式除了基本的需求滿足限制式、時段衝突限制式，工作時段限制式之外，最特別的探討員工休息時間限制式，除了強調司機人員的偏好外，也給予了基本的休息需求。在實證方面，分為靜態變數指示 (Static Variable-Ordering) 及循環變數指示 (Cyclic variable-Ordering) 兩種方式來做比較，靜態變數指示求解的效率遠比動態變數指示好，說明不同順序的導入規則，將對求解效率造成不同的影響。



#### 2.4.2 陸運業

蘇昭銘與張靖 【4】 對捷運系統站務人員排班，利用整數規劃方式建構一數學模式。其排班過程分為兩階段，第一階段模式主要在決定站務人員之上班日與休假日，故需符合每日人力最小需求、工作時數與排班循環規則等三項條件。該模式除可提供捷運系統營運機構進行一般性排班工作外，尚可處理站務人員指定特定日期進行正常性休假與連續性特別休假兩種特殊情形，賦予員工較大的休假自主權和彈性。第二階段模式主要在決定站務人員上班日之上班時段，故需滿足各時段最小人力需求與值勤班別穩定性兩項基本條件，該階段模式在實務上亦可處理營運機構對班別之特殊規定（如台北捷運公司規定女性站長不得上夜班之規定）。在符合排班循環規則及滿足需求下，令捷運系統站務人員排班集中度、班別穩定性及排班平均化達到最大，以建立輪休及值班表。

Higgin 【25】對鐵路軌道的維修工作與維修人員排班問題，在對原始列車營運班表的變動影響最小為目標下，考慮維修人員需求與維修工作的進行順序及預算限制，並將

問題定式為一多目標整數規劃問題，利用塔布搜尋(Tabu Search)啟發式解法，求解鐵路維修工作與維修人員之配置及排班。

Caprara et al. 【19】針對鐵路運輸問題進行人員排班問題之規劃，模式構建以最小化成本為目標，並以已知的列車班表進行組員排班問題的求解。其分別構建一集合涵蓋問題與最小化成本網路流動問題，再使用拉氏鬆弛法以及自行發展的啟發式演算法產生可行班次，並求解最佳化的人員班表。

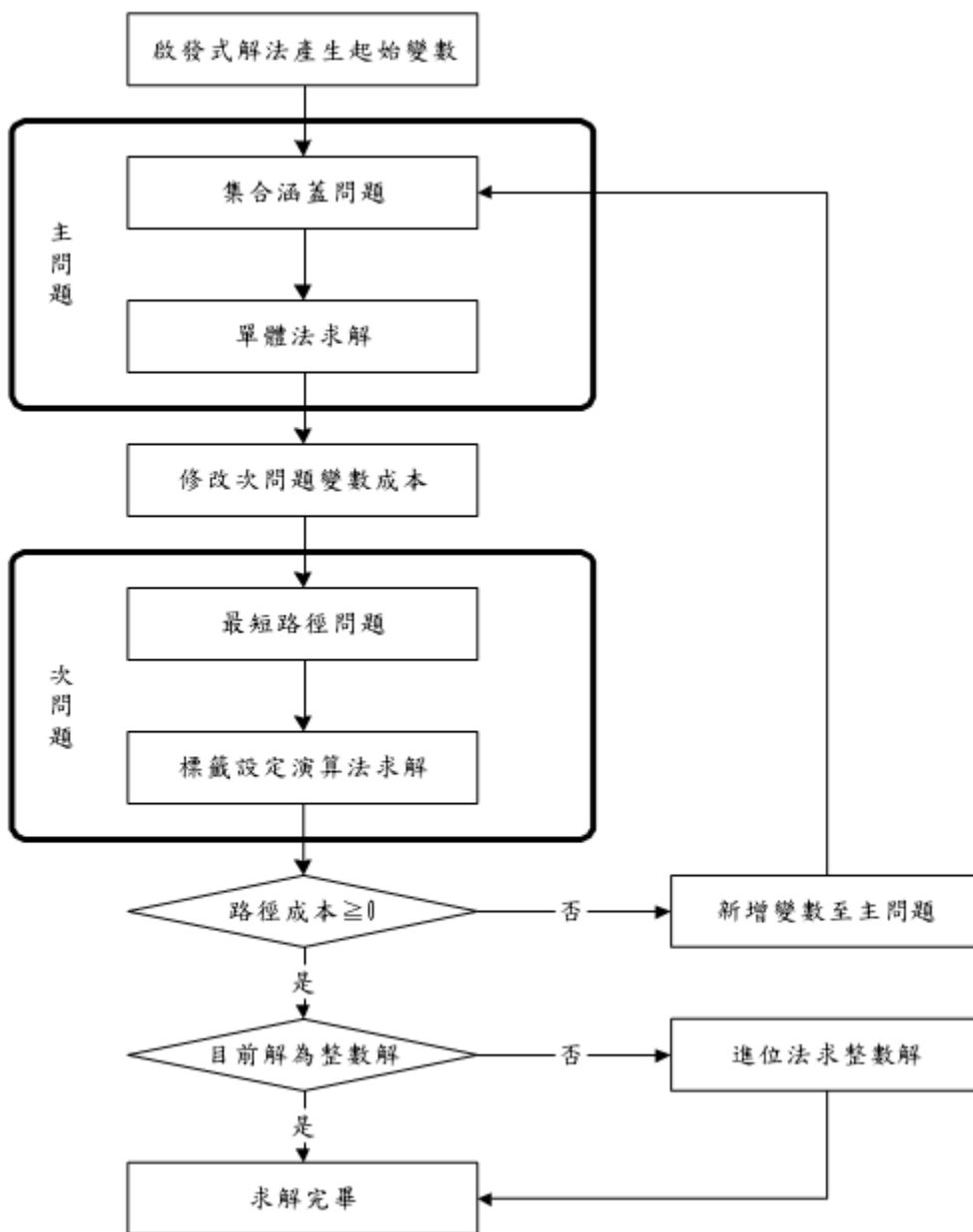
### 2.4.3 空運業

Chew 【21】處理停機坪工作人員之排班問題，在第一階段利用啟發式程序求解最低的總人數需求及第二階段利用整數規劃模式產生循環性 (Cyclic Day Line) 的輪值表 (Rosters) 且設計休假日 (Off-Duty Days) 和工作日 (Working-Days) 以符合輪值班次，而該篇研究著重於第二階段之方法論。其決策變數為，在滿足人力需求下，使循環工作數最少，也就是使工作組數最少。主要是在符合一週內各天中之每一班次的人力需求下，安排員工之休假日及工作日的輪值班次，也就是訂定所有工作人員的輪值表。除了每天上、下午班所需的人力需求外，尚須考量到每一班次的銜接問題。也就是以某一天為結束班型的人員組數數量總和必須等於以該天為起始班型的人員組數數量總和。因此，經作者證明可以較方便的方式推算出結果，不需以傳統方式整數規劃解法進行其解。在決定輪休日及每天上、下午班人員之後，仍須進一步對各工作天的實際輪值班次進行指派，其原則是每一班型以輪值之起始時間不遞增 (Non-Increasing) 為基準，故仍能以較簡易的啟發式程序求得。

顏上堯與林錦翌【6】在「空服員排班組合之最佳化問題」中把問題模式定義為集合涵蓋問題，求解方法則利用變數產生法 (column generation) 求解。此研究中，假設模擬班次表及空服員排班網路為已知進行求解，首先以人工方式產生足夠數量的可行值

勤航班 (pairing) 組合，獲得問題的起始變數。之後，將求解過程分為對主問題與子問題的求解；主問題為集合涵蓋問題，只考慮已產生的部分變數，以單體法求解，找出此部分變數的最佳解，利用求解主問題所得的資訊，加入次問題，修正次問題的參數；次問題則為最短路徑問題 (shortest path problem)，並利用標籤設定演算法 (label setting algorithm) 求解，產生有助於改善主問題目前解的變數，再加入主問題中，重新求解。以此程序逐漸改善目前解，直到無法改善為止。





資料來源: 顏上堯、林錦翌, 「空服員排班組合最佳化之研究」

陳立欣【3】在平均化越來越受重視下，建構一平均化成本函數，並以最小平均化成本為目標，產生公平之班表。將後艙組員派遣問題列式為一集合分割模式並建構一啟發式解法，分為三階段來求解此問題。第一階段為長期勤務指派，將飛行天數大於（或等於）三天的勤務組合，在滿足各種限制以及最小平均化成本的目標下，全數指派給後

艙組員；第二階段為兩日短勤務指派，將所有飛行天數為兩天的勤務組合，同樣在法規限制與最小平均化成本的目標下指派給組員；第三階段為一日短勤務指派，將飛行天數為一天的勤務組合全數指派給組員。另外，設計一計算成本函數，能將每個組員被指派的當月班表行程所產生之平均化成本量化出來。此成本值為假設，並不是航空公司實際成本。最後，在最小成本勤務組合以及組員既定活動已知的情況下，利用啟發式解法求解，將長勤務、短勤務組合指派給組員，並產生合法可行之平均化排班。

## 2.5 小結

綜合上述所整理的文獻可知，針對各種不同性質的人員排班問題，各有不同的求解方法。對於運輸業的排班問題，可以藉由構建模式的概念，以發展一個適用於各企業執勤及運輸業人員排班的數學模式。例如利用數學規劃法以求解人員排班問題、而各種分割法或是啟發式解法等亦經常被使用。大型排班問題所用的排班方法著重於啟發式演算法、中型排班問題則是最佳化演算法及啟發式演算法均有利用、中小型排班問題則是著重於最佳化演算法。此與前述的排班規模、求解方法不謀而合，顯示研究人員排班方法實可朝這幾個方向做考量。



## 第三章 現況分析與檢討

### 3.1 現況之作業環境

本研究範圍之 E 公司所租用的碼頭是第五貨櫃中心七十九號、八十號及八十一號碼頭。第五貨櫃中心七十九、八十、八十一號三座碼頭基地相連成 L 型，全長八一五公尺，水深十五公尺，總面積達四十萬一千八百六十六平方公尺，可同時泊靠三艘大型貨櫃輪進行裝卸作業。七十九號碼頭長度為三百五十五公尺，寬度約為三十八公尺，水深十五公尺，設置橋式起重機(Gantry Crane, GC)四具。八十號碼頭長度為三百四十公尺，寬度三十二公尺，水深十四公尺，橋式起重機原本為兩具，至 2005 年底又添增一具，總共為三具。八十一號碼頭長度為一百二十公尺，寬度三十二公尺，水深十四公尺，橋式起重機配置兩部，如表 3.1。其中八十號及八十一號碼頭為連號碼頭，當大型船舶停靠且只有一座碼頭需要作業時，橋式起重機在兩座碼頭之間可自由移動，以便支援，加快裝卸速度及提升效率。

以高雄港 E 公司租用的碼頭來說，機具之數量係依碼頭大小來設置。如七十九號碼頭長度較長，可容納較為大型的船隻停靠，該碼頭就設置了四部機具，八十號碼頭因長度夠長，亦常停靠大型船舶，該公司共設置三部機具，八十一號碼頭則僅設置兩部機具。

表 3.1 貨櫃碼頭資料

碼頭編號	長度	寬度	設計水深	碼頭類別	裝卸機具(固定式)	使用單位
七十九號碼頭	355	38.46	15	貨櫃	GC 4 台	出租
八十號碼頭	340	32	14	貨櫃	GC 3 台	出租
八十一號碼頭	120	32	14	貨櫃	GC 2 台	出租

資料來源:港務局提供、本研究整理

### 3.2 現況之作業流程



貨櫃動態的實體流程是始自貨櫃(重櫃)卸船後、進場、拆櫃、空櫃交還、空櫃領出、重櫃進場、至重櫃裝船後形成一作業週期(Cycle)。而貨櫃碼頭即扮演一個作業環境的主體，提供起重機具裝卸作業人員操作，以便進行從船舶至陸地間貨櫃的運送。

橋式起重機通稱為橋吊或岸橋，係豎立在碼頭前緣，固定在軌道上運行，藉著吊臂延伸至海面，用來裝、卸貨櫃的一種機具，是為貨櫃碼頭上最醒目的設備。岸橋式起重機是在港口中裝卸貨櫃之主要機具，除了具備碼頭與陸地間貨櫃轉運功能外，亦是貨櫃碼頭裝卸效率重要指標之一，其作業效率之發揮亦對整個港口發展影響甚鉅。

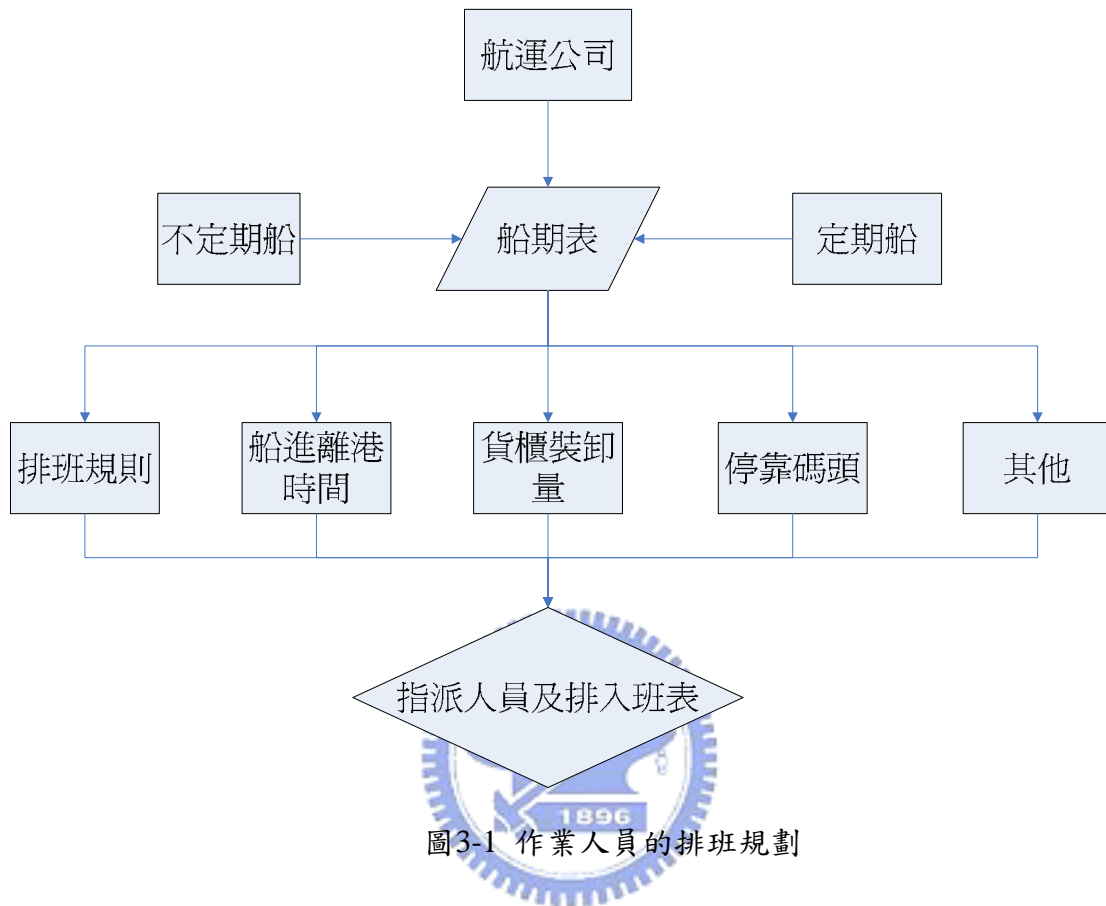
而橋式機司機員為操作起重機之主要人員，負責處理岸上及陸上的貨櫃抓吊及置放。因為橋式機的抓吊作業為高空中進行，避免高空作業時間過長，必須在規定時間內換班，原則上為四小時至六小時為一班次，執勤完一個班次即換班，輪流由下一位司機員作業。



### 3.3 橋式機人員指派及班表規劃考量之現況

在人力安排及班表規劃中，如圖3-1，航運公司只要租用碼頭以後，可以自由分配船舶停靠至特定自用碼頭，不需受到港務局的限制或指派。在所有不定期船及定期船之船期通報後，航運公司在一個星期前會得知在該禮拜即將到港停靠的船舶及大略到港時間。接獲的資料中包括到港停靠船舶的船型大小、到港時間及貨櫃裝卸量。由船舶的船型大小即可安排適當的停靠碼頭，如同時到港的船舶，選其中較大的船隻停靠於長度較長的碼頭。由貨櫃裝卸量即可知道，所需安排的機具數及預計裝卸時間，亦可知道約略其知道離港時間。另外，貨櫃裝卸量多寡也可能影響到停靠的碼頭。如在所有到港船舶皆可安排停靠之狀況下，貨櫃裝卸量較多的船舶就有可能安排至橋式起重機具數較多的碼頭，如七十九號碼頭就有四具。另一方面也可能於連號碼頭中長度較長的八十號碼頭，連同八十一號碼頭的機具自由移動來支援，總共作業機具為五部。航運公司係依據

上述資訊及原則，以及該公司的排班規則，最後指派人員數排入規劃中的班表。



### 3.4 值勤班次時間分配

橋式起重機司機人員分派班次的原則，考量因高空作業，需要高度專注力，耗費大量精神，故工作時數不可過長，所以一天時間中分派早班、中班、晚班、小夜班及大夜班五個班次。早班值勤時間從每日早上八點到下午兩點，中班是下午兩點到晚上八點，晚班從晚上八點到凌晨零點，小夜班從零點到凌晨四點，大夜班從凌晨四點到早上八點。除了早班及午班是執勤值勤6個小時外，晚班、小夜班及大夜班因為執勤時間在晚上，能見度較差，必須耗費更多專注力與精神，所以只值勤四個小時。業者在工作班次時間上之安排如表 3-2 所示。

表 3.2 E 航運公司橋式起重機司機員工作班次

班次	早班	中班	晚班	小夜班	大夜班
值勤時間	08:00-14:00	14:00-20:00	20:00-00:00	00:00-04:00	04:00-08:00
工作時數	6	6	4	4	4

資料來源:航運公司提供，本研究整理

### 3.5 實務上作業指派方式及原則

本論文係以司機員作業指派排班為主要探討項目，進而研究在既定的整體環境下，找出適合的安排方式及班表。然而船泊靠港裝卸的時間並不固定，每個班次的工作時間長短也不盡一致，在此情況下排班的困難度大幅提高。所以需先了解該公司的排班邏輯和原則，始能洞悉可以改善之處。

目前 E 公司於橋式起重機司機員排班的指派方式分為三個階段。茲分述如下:

1. 指派作業的第一階段是將所有司機分組，組別總數共有十組，每一組皆有五名司機。以表 3.3 為例是為第七組。第七組中司機員的編號是 701、702、703、704 及 705。以下以此類推。

表 3.3 實務作業指派第一階段

人事代號	組別
701	7
702	7
703	7
704	7
705	7

資料來源:航運公司提供

2. 第二階段為先規劃好各組別在每一天中處理船隻的順序。以表 3.4 為例，在當天第一艘船到港時由第七組負責裝卸貨櫃，第二艘船到港時由第八組執勤，第三艘船則由第九組執勤，以此類推。

表 3.4 實務作業指派第二階段

人事代號	組別
701	7
702	7
703	7
704	7
705	7
801	8
802	8
803	8
804	8
805	8
901	9
902	9
903	9
904	9
905	9

資料來源:航運公司提供

3. 最後一階段是在既定的組別及日期下安排每一位司機成員的班次，依據編號來安排五個班次。以表 3.5 為例，即為安排一個禮拜的排班。第八組從十二月二日起一個禮拜到十二月八日，皆為先值勤組別。若到下一個禮拜則升序由下一組，即第九組為先值勤組別。但是在十二月二日，第八組及第九組個別為排休及休假，所

以在十二月二日當天由第十組先值勤。在第八組中，分別安排司機編號 801、802、803、804 及 805 分別為晚班、小夜班、大夜班、早班及中班。

表 3.5 實務作業指派第三階段

日期	組	1	日期	組	2	3	4	5	6	7	8
人事代號\姓名\星期	別	六	姓名	別	日	一	二	三	四	五	六
701	7	小	801	8	晚休	晚	休	晚	晚休	晚	晚
702	7	大	802	8	小休	小	休	小	小休	小	小
703	7	早	803	8	大休	大	休	大	大休	大	大
704	7	中	804	8	早休	早	休	早	早休	早	早
705	7	晚	805	8	中休	中	休	中	中休	中	中
801	8	小	901	9	休	晚	晚	晚	晚休	晚	晚
802	8	大	902	9	休	小	小	小	小休	小	小
803	8	早	903	9	休	大	大	大	大休	大	大
804	8	中	904	9	休	早	早	早	早休	早	早
805	8	晚	905	9	休	中	中	中	中休	中	中
901	9	休	1001	10	晚	晚	晚	晚休	晚	休	晚
902	9	休	1002	10	小	小	小	小休	小	休	小
903	9	休	1003	10	大	大	大	大休	大	休	大
904	9	休	1004	10	早	早	早	早休	早	休	早
905	9	休	1005	10	中	中	中	中休	中	休	中
1001	10	小	101	1	晚	晚	晚休	晚	休	晚	晚
1002	10	大	102	1	小	小	小休	小	休	小	小
1003	10	早	103	1	大	大	大休	大	休	大	大
1004	10	中	104	1	早	早	早休	早	休	早	早
1005	10	晚	105	1	中	中	中休	中	休	中	中
101	1	休	201	2	晚	晚休	晚	晚	晚	晚	休
102	1	休	202	2	小	小休	小	小	小	小	休

日期	組	1	日期	組	2	3	4	5	6	7	8
人事代號\姓名\星期	別	六	姓名	別	日	一	二	三	四	五	六
103	1	休	203	2	大	大休	大	大	大	大	休
104	1	休	204	2	早	早休	早	早	早	早	休
105	1	休	205	2	中	中休	中	中	中	中	休
201	2	小	301	3	晚	晚	晚	休	晚	晚休	晚
202	2	大	302	3	小	小	小	休	小	小休	小
203	2	早	303	3	大	大	大	休	大	大休	大
204	2	中	304	3	早	早	早	休	早	早休	早
205	2	晚	305	3	中	中	中	休	中	中休	中
301	3	小休	401	4	晚	晚	晚	休	晚	晚休	晚
302	3	大休	402	4	小	小	小	休	小	小休	小
303	3	早休	403	4	大	大	大	休	大	大休	大
304	3	中休	404	4	早	早	早	休	早	早休	早
305	3	晚休	405	4	中	中	中	休	中	中休	中
401	4	小休	501	5	晚	晚	休	晚	晚休	晚	晚
402	4	大休	502	5	小	小	休	小	小休	小	小
403	4	早休	503	5	大	大	休	大	大休	大	大
404	4	中休	504	5	早	早	休	早	早休	早	早
405	4	晚休	505	5	中	中	休	中	中休	中	中
501	5	小	601	6	晚	休	晚	晚	晚	晚	晚休
502	5	大	602	6	小	休	小	小	小	小	小休
503	5	早	603	6	大	休	大	大	大	大	大休
504	5	中	604	6	早	休	早	早	早	早	早休
505	5	晚	605	6	中	休	中	中	中	中	中休
601	6	小	701	7	休	晚	晚	晚	晚休	晚	晚
602	6	晚	702	7	休	中	中	中	中休	中	中
603	6	早	703	7	休	大	大	大	大休	大	大

資料來源:航運公司提供

### 3.6 現況排班結果與檢討

按照該公司排班原則及指派方式，是以規劃好的每個星期班表中，先值勤組先處理第一艘靠港船舶的貨櫃裝卸作業，而不同的碼頭有不同的機具數量，然而一架機具只能由一位司機員操作。因此視該碼頭在作業時間需多少機具，就派遣相同機具數量的組別去處理船舶之貨櫃裝卸作業。例如，80 號碼頭在 12 月 1 日凌晨 00:00 安排兩部橋式起重機作業，所以需要兩組司機員支援值勤。所以在 12 月 1 日第一艘靠港的船舶或會裝卸作業由該禮拜的先值勤組別及下一組，意即第八組及第九組負責 80 號碼頭該時段的貨櫃裝卸作業。若下一艘船在若干時間後靠泊，則端看停靠碼頭安排作業中所需上工之機具數，即可知道需安排幾組司機員操作作業。

E 公司目前所規劃班表的方式是類同一般企業公司的排班原則，一次將一個月的班表規劃出來，所有的員工皆可清楚的知道上班的時間，並且在排班的方法上也較為簡單。但是由於一般公司企業上班時間較為穩定，工作時數也比較固定，所以較適用此排班方法。以海運來說，由於船舶進離場時間不確定因素、碼頭大小、起重機機具數及貨櫃裝卸量多寡，對於在港邊工作人員的排班作業皆是必須考量的。這些在港邊岸肩作業較為不穩定的因素若套用一般公司企業穩定式排班方法，就不是那麼適合了。

依該公司的排班方式，在不確定船隻到港時間的狀況下，極有可能造成分工不均的情形。如同統計學中重複抽取紅球的理論，抽取紅球之後再度放回箱子中，極有可能皆為紅球。換句話說，在該公司指派班表之案例上，很可能在一個月中，大多都由某幾組，甚至是某幾位司機執勤，而其他司機皆為待命放假的狀態。只有在長期時間，一年或者數年以上，意即是樣本數夠多的情況下，各司機的執勤班次總和才可能有不顯著的差異。另外，在班表中的休假方式並沒有一個特定的方式。

本論文先依循該航運公司的排班邏輯，以公司提供之船期表、司機員班表為結果，進而探討目前該公司排班上有何缺失及待改善之處。輸入資料如表 3.6，時間為自 2007 年 12 月 2 日至 12 月 8 日為期一週之船期表，船舶名因公司隱私以英文符號代替。BOX 為該船之貨櫃裝卸量，ETB 為開始裝卸時間，ETD 為結束裝卸時間。工作班次如表 3.2 司機員之工作時間表以及執勤休假表如表 3.5。

表 3.6 船舶到港裝卸貨櫃之資料

碼頭	船名 VVD	BOX	ETB	ETD	GC'S	WORK HOURS
81 號	A	990	12/2 01:00	12/2 19:00	2	18
80 號	B	1360	12/2 03:00	12/2 19:00	3	16
79 號	C	1160	12/2 15:00	12/3 01:35	4	10.5
79 號	D	2530	12/3 08:00	12/4 07:00	4	23
80 號	E	2240	12/3 09:30	12/4 01:30	5	16
80 號	F	2520	12/4 02:30	12/4 20:00	5	17.5
79 號	G	1490	12/4 08:00	12/4 21:30	4	13.5
80 號	H	3360	12/5 01:00	12/6 01:00	5	24
79 號	I	1650	12/5 08:00	12/5 23:00	4	15
80 號	J	250	12/7 00:20	12/7 05:00	2	4.5
81 號	K	980	12/7 00:30	12/7 12:00	3	11
81 號	L	1960	12/7 08:00	12/7 22:00	5	14
80 號	M	1260	12/7 22:00	12/8 07:00	5	9
80 號	N	1020	12/8 11:25	12/8 23:30	3	12
81 號	O	660	12/8 12:30	12/8 00:30	2	12

資料來源:航運公司提供，本研究整理



如表 3.6 所示，12 月 7 日第一艘停靠的船舶為 J 船，靠泊的碼頭為第 80 號碼頭，貨櫃裝卸量為 250 個貨櫃。預計開工時間(ETB)為凌晨 00:20，結束時間(ETD)為 05:00。安排作業之橋式起重機機具數為兩部，其中一部因尚在作業時間中，下一艘停靠於第 81 號碼頭的船舶 L 因貨櫃裝卸量較大，故先過去支援。貨櫃裝卸之作業時間約為 4.5 小時。其中船舶 L 到港後 08:00 先停靠於第 80 號碼頭，在早班時段中(08:00-12:00)因 81 號碼頭正在使用其餘三部橋式機，故在 08:00 至 12:00 只安排兩部橋式機作業。到了 12:00 時，停靠第 81 號碼頭之船舶 K 貨櫃裝卸作業結束後離港，才將原屬第 80 號碼頭派遣過去支援船舶 K 的橋式機連同第 81 號碼頭原有的兩部機具移動去第 80 號碼頭供應船舶 L 貨櫃裝卸作業使用。

航運公司依據此排班邏輯及現有資料對作業司機員指派班表。現況輸入需求時間自 2007 年 12 月 2 日至 12 月 8 日，安排出之司機員班表如下表 3.7 所示。



表 3.7 E 航運公司之司機員班表 (2007 年 12 月 2 日至 12 月 8 日)

BO X	ETB	ET D	碼頭	GC'S	值勤組別	休假組別	待命組別	WORK HOURS	小夜班 00:00-04:00	大夜班 04:00-08:00	早班 08:00-14:00	中班 14:00-20:00	晚班 20:00-00:00
990	12/2 01:00	12/2 19:00	81號	2	10,1	7	8,9	18	1002,102	1003,103	1004,104	1005,105	
1360	12/2 03:00	12/2 19:00	80號	3	2,3,4			16	202,302,402	203,303,403	204,304,404	205,305,405	
1160	12/2 15:00		79號	4	5,6,8,9			10.5				505,605,805 905	501,601,801 901
		12/3 01:35	79號	4	8,9,10,1	2	6	10.5	802,902,1002 102				
2530	12/3 08:00		79號	4	10,1,3,4			23			1004,104,304 404	1005,105,305 405	1001,101,301 401
		12/4 07:00	79號	4	9,10,2,3	1	5,8	23	902,1002,202 302	903,1003,203 303			
2240	12/3 09:30		80號	5	5,6,7,8, 9	2	6	16			504,604,704 804,904	505,605,705 805,905	501,601,701 801,901
		12/4 01:30	80號	5	4,5,6,7, 8	1	5,8	16	402,502,602,702,8 02 (共用班次)				
2520	12/4 02:30	12/4 20:00	80號	5	4,5,6,7, 8			17.5	402,502,602,702,8 02 (共用班次)	403,503,603 703,803	404,504,604 704,804	405,505,605 705,805	
1490	12/4 08:00	12/4 21:30	79號	4	9,10,2,3			13.5			904,1004,204 304	905,1005,205 305	901,1001,201 301
3360	12/5 00:30		80號	5	8,9,1,2, 5	10	3,4	23.5	802,902,102,202,3 02	803,903,103 203,303	803,904,104 204,304	805,905,105 205,305	801,901,101 201,301
		12/6 01:00	80號	5	10,2,3,4 .6	1	5,7,8, 9	23.5	1002,202,302,402, 602				
1650	12/5 08:00	12/5 23:00	79號	4	6,7,3,4	10	3,4	15			604,704,304, 404	605,705,305,405	601,701,301,401
250	12/7 00:20	12/7 05:00	80號	2	8,9	3, 4	10	4.5	802,902	803,903			
980	12/7 00:30	12/7 12:00	81號	3	1,2,5			11	102,202,502	103,203,503	104,204,504		
1960	12/7 08:00	12/7 22:00	81號	5	6,7,8,9, 1			14			604,704	605,705,805,905, 105	601,701,801,901, 101
1260	12/7 22:00		79號	4				9					601,701,801,901 (接上一班)
		12/8 07:00	79號	4	8,9,10,1			9		802,902,1002,102	803,903,1003, 103		
1020	12/8 11:25	12/8 23:30	80號	3	3,4,5	6	2	12			304,404,504	305,405,505	301,401,501
660	12/8 12:30	12/9 00:00	81號	2	7,8			12			704,804	705,805	701,801

資料來源:航運公司提供，本研究整理

依循目前現有的排班邏輯，依據船務公司提供所安排之班表(表 3.7)經統計後如表 3.8 所示。總值勤班次需求為 189 個班次。各司機一周值勤班次由一班次至 7 班次不等，分別為執勤一個班次共有 3 人，值勤二個班次有 7 人，值勤三個班次有 9 人，四個班次有 13 人，五個班次為 16 人，六個班次為 1 人，七個班次也為 1 人。司機員值勤班次的平均數為 3.78 個班次，而值勤班次的變異數卻高達 1.81 個班次。

表 3.8 各司機員值勤班次數(範例)

司機員編號	<b>101</b>	<b>102</b>	<b>103</b>	<b>104</b>	<b>105</b>	<b>201</b>	<b>202</b>	<b>203</b>	<b>204</b>	<b>205</b>
值勤班次數	3	5	4	4	4	2	5	4	5	2
司機員編號	<b>301</b>	<b>302</b>	<b>303</b>	<b>304</b>	<b>305</b>	<b>401</b>	<b>402</b>	<b>403</b>	<b>404</b>	<b>405</b>
值勤班次數	5	4	3	7	5	3	3	2	5	5
司機員編號	<b>501</b>	<b>502</b>	<b>503</b>	<b>504</b>	<b>505</b>	<b>601</b>	<b>602</b>	<b>603</b>	<b>604</b>	<b>605</b>
值勤班次數	3	2	2	4	4	4	2	1	4	5
司機員編號	<b>701</b>	<b>702</b>	<b>703</b>	<b>704</b>	<b>705</b>	<b>801</b>	<b>802</b>	<b>803</b>	<b>804</b>	<b>805</b>
值勤班次數	4	1	1	5	5	5	5	4	4	6
司機員編號	<b>901</b>	<b>902</b>	<b>903</b>	<b>904</b>	<b>905</b>	<b>1001</b>	<b>1002</b>	<b>1003</b>	<b>1004</b>	<b>1005</b>
值勤班次數	5	5	4	3	5	2	5	3	3	3

資料來源:本研究根據 E 公司提供資料整理

### 3.7 小結

由此結果得知，雖說司機員執勤大多集中於四至六個班次，但是變異的幅度過大，值勤班次最少為一班，最多為七班，司機員在總執勤班次數之差異達到六個班次之多。在工時的統計上顯示了分工極度不均之情形，目前現況之排班規則及指派邏輯尚有值得改善之處。

## 第四章 模式構建

本章首先對於限制式規劃及整數規劃進一步加以比較，並討其求解演算法之程序。其次再比較目前常見之解算最佳化問題之套裝軟體以利構建模式之用。最後以整數規劃建構一數學模式，再以航運公司提供實務上之原始資料做為測試背景，分析本模式是否能尋找並提供一更理想的排班結果。本模式主要以 C 程式語言撰寫輸入檔產生器，並使用 ILOG CPLEX 數學規劃套裝軟體求解。

### 4.1 限制式規劃與整數規劃

此節主要於限制式規劃及整數規劃概念性之分析及探討。因一般性作業問題常以混合整數規劃 (Mixed Integer Programming, MIP) 的方式構建模式以進行求解。數學規劃 MIP 之變數大多為連續型及離散型，目標式為線性函數且限制式為標準型式，且求解演算法為樹狀搜尋法。而限制式規劃之變數與值域為離散型、不必需要目標式、無標準型式之限制式且求解演算法結合一致性技巧與樹狀搜尋法。

### 4.2 限制式規劃之求解

限制式規劃的求解主要分為兩部分，第一部分為一致性技巧 (Consistency Technique)，其內容為限制式繁衍 (Constraint Propagation) 與值域刪除 (Constraint Propagation and Domain Reduce)。另一部分則起源於 CSP 之樹狀搜尋演算法，常見之求解方法分 Generate and Test、Backtracking 與 Forward Checking 等，由於前兩種演算方式效率較差，現今套裝軟體大多使用 Forward Checking 配合一致性技巧進行求解。

#### 4.2.1 一致性技巧

一致性技巧之概念主要分為兩個部分。一為求解過程中之值域刪除 (Domain

Reduce)，主要為刪除不滿限制式之變數值域，再加以修正其個數。另一為限制式繁衍，其意義為將變數值域之改變繁衍至與該變數相關之限制式。如下圖4-1所示，當變數修改後，限制式更新其變數值域，並且將新增之值域限制式衍生傳達至原始之限制式 (Primitive Constraint) 中，再由原始限制式移除不滿足限制式之變數。

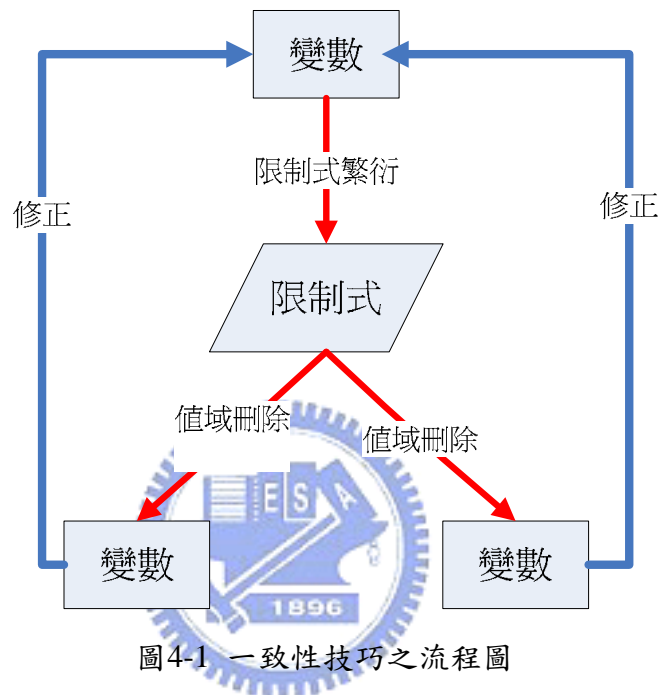


圖4-1 一致性技巧之流程圖

#### 4.2.2 樹狀搜尋演算法

常見之樹狀搜尋演算法大致分為以下三類

##### 1. Generate and Test

Generate and Test 為最基本之演算法，運作之概念由 Generator 產生所有解，接續由 Tester 檢查 Generator 產生之所有解是否符合問題之限制式。

##### 2. Backtracking

Backtracking 演算法之概念是利用樹狀搜尋基本概念，將其變數依據值域各數分枝，以樹狀圖之節點表達其變數。進行搜尋作業時，正在搜尋的變數節點稱作

「Current Variable」，而先前蒐尋過之節點稱為「Past Variable」，而尚未蒐尋之節點稱為「Future Variable」。此類演算法搜尋只能檢視 Current Variable 與 Past Variable 相關限制式，無法防範可能產生的衝突，故此類樹狀搜尋之空間相當龐大且效率也較差。

### 3. Forward Checking

相較於上述兩種演算方法，Forward Checking 效率較高，其搜尋概念雖同樣以樹狀搜尋法為基礎，但與 Backtracking 之不同在於 Forward Checking 同時考慮 Current Variable、Past Variable 與 Future Variable 相關之限制式，可預防未來可能發生之衝突。當發生衝突時，之後考慮相關之位置皆會從其值域中刪除，故搜尋空間大幅縮小且求解效率較佳。

## 4.3 最佳化問題之限制式規劃求解概念

雖然限制式規劃概念起源於滿足限制式之問題，但也可用來求解最佳化問題，本節以 ILOG Solver 為例，說明限制式規劃處理目標函數方式，軟體提供之搜尋方法有兩種，分別為標準法 (Standard) 與二分法 (Dichotomic) 兩種方式。

### 1. 標準搜尋程序

最佳化問題之處理方式是將目標函數視為一個限制式，如下圖4-2所示，標準搜尋程序每一次產生一個解當作目標函數之式上限值 (Upper Bound, UB)，當心的上限值比上一個小時，則將限制式 (目標函數 < UB) 傳達出去，並且重覆搜尋下一個解，直到新的上限值不再比之前的小始為最佳解。

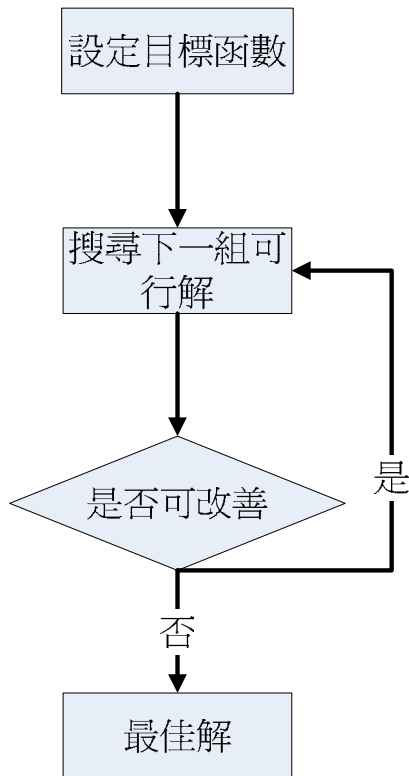


圖4-2 標準搜尋程序流程圖



## 2. 二分搜尋法程序

二分搜尋法利用目標函數之上限值 (Upper Bound, UB) 與下限值 (Lower Bound, LB) 來設定目標限制式。如下圖4-3所示，首先設立目標函數之LB，之後取得一組可行解，再將該組可行解做為目標函數之UB，接續檢視(LB-UB)是否落在設定區域內，如果是則停止，否則進而令 $Z=(UB+LB)/2$ ，增加一新限制式 (目標函數 $<Z$ ) 並將之繁衍傳達出去，繼續求解加入新限制式之CSP。若存在可行解，則以Z作為UB重新計算Z；若可行解不存在，則以Z做為LB再次重新計算Z。

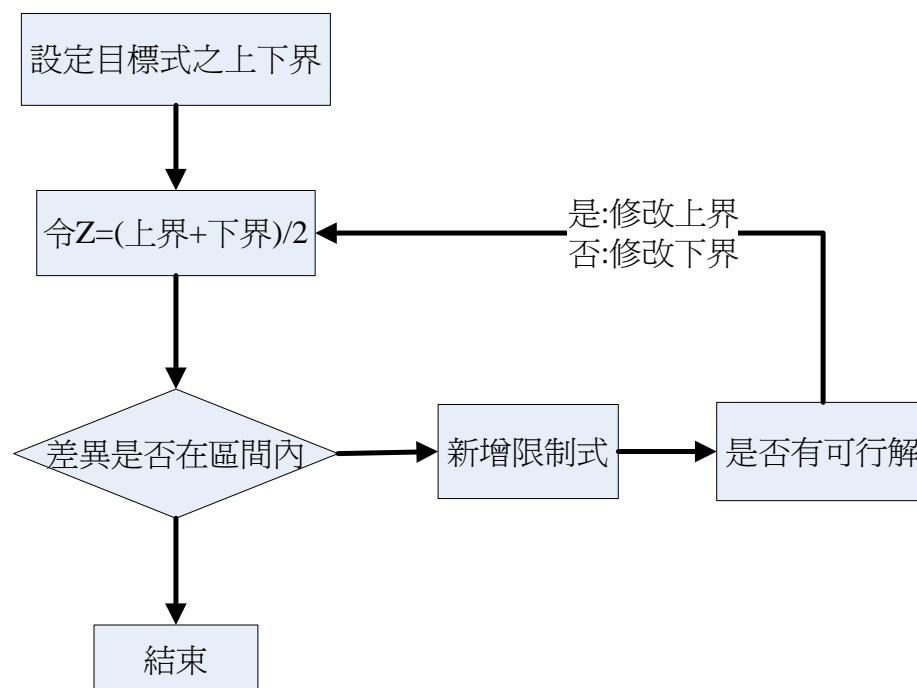


圖4-3 二分搜尋法之流程圖



## 4.4 最佳化求解軟體

撰寫限制式之寫法關鍵在於使用的軟體，視軟體類型的不同，寫入限制式中所需要的語法亦不相同。ILOG Solver 的優點為接受大多電腦語言體系之表達方式，如C語言 (C Language) 或者接近C語言的程式，也有該軟體所專用之更簡便的語言方式。以下介紹一些較為常見之最佳化求解軟體與應用其之研究。

### 4.4.1 CPLEX

辛怡達與林正章【9】在郵局快捷專車排班排程問題之研究中分析國內快遞市場的特性及瞭解郵局快捷業務目前的營運作業方式。以不具容量限制的不同路線間允許轉運的網路架構為基礎，研擬出適用於已知服務時間的車輛排程排班模式，利用 cplex 軟體求解模式，以決定總運輸成本最低的運輸路徑。



#### 4.4.2 ILOG OPL

林詩芹與韓復華 【11】 將客服人員排班問題定式為一限制滿足問題，以國內銀行客服中心為個案，將客服人員排班問題定式為一限制滿足問題，依據其實際排班情形及服務單位之排班規定與班表公平原則並應用限制規劃方法(Constraint Programming, CP)可彈性處理各類限制式之特性構建人員排班問題之限制規劃模式，最後以ILOG OPL Studio求解。

謝玉霜與陳春益 【10】 針對港區貨櫃場軌道式門型起重機之特性與移動路徑問題進行探討，參考Kim之門型起重機排程問題並且重新撰寫移動路徑模式，利用限制式規劃協助進行模式之構建，最後使用由C++撰寫成的ILOG OPL軟體進行求解。

#### 4.4.3 AMPL CPLEX

翁得榮與喻奉天 【12】 收集高雄第一科技大學運籌管理系排課相關資料並且加入教師偏好時段及其他眾多因素等因子，利用整數規劃法建立模型，並且藉由AMPL/CPLEX 軟體結合 Microsoft Office Access 建構之資料庫以快速解決排課問題。

#### 4.4.5 小結

AMPL 和 GAMS，LINDO，LINGO類似，都是代數模型的介面，可以把數學模式轉換成CPLEX或其他Solver能接受的輸入格式，比直接以C++呼叫CPLEX容易，但是也失去了一些彈性。OPL大概介於AMPL和C++之間，寫法有點近似C++。

以本研究為例，先以C++撰寫限制式產生器，如TEXT檔，包含所有目標式及限制式，並將需求部分編輯成輸入檔產生所有限制式，最後加入 CPLEX 語法修改標頭檔，再用軟體輸出結果。

#### 4.5 排班邏輯修正

考慮目前排班規則上的缺失，在本研究重建一邏輯模式，期能尋找更為理想的排班方式。首先，將原來公司分派司機員至各組數的方式改成打散為每個司機員為一個體，不再以原有組數的概念分配至裝卸作業中。第二，將公司沿用一般企業排班模式，意即每個司機員原先分配一個禮拜中固定某個班次改成按照司機員編號順序分配，變成有工作時才上工，並且依照操作橋式機所需人數安排司機員，以編號順序輪流值勤。最後將原來班表中有些共用班次及接續班次的部分拆開。共用班次的部分為 12 月 4 日於 80 號碼頭小夜班班次，安排之司機員原為 402,502,602,702,802。接續班次的部分為 12 月 7 日 79 號碼頭由司機員 601,701,801,901 接續 81 號碼頭工作部分。因實際上班次部分在交接時換船還必須一部份的時間，避免司機員因換班過於匆忙以及為後續分析上之便利，故將班次拆開。

#### 4.6 休假方式之重新擬定



作業人員之一日值勤時間計算，從第一天凌晨 00:00 至次日凌晨 00:00 為一個工作天，除了公司規定每個禮拜一天假日外，也可另行請假，另外還有額外一天待命支援的排休假日。而無船舶到達時皆屬待命狀態，不算休假。由於公司訂定每一組一週約休一天，因此本研究沿用公司休假天數來排定休假，按照組員編號順序(如 101、102、103...等)安排休假。另外該公司有提及，一個禮拜內因定期船部分，週日開始至週三較多班次到達，故在本研究中將週一至週三擬定排入休假的人數較少為六人，其餘天數皆為八人，嘗試以此休假方式測試排班結果。

表 4.1 司機員休假表

星期	12 月	休假之司機員編號
日	2 日	101、102、103、104 105、201、202、203
一	3 日	204、205、301 302、303、304
二	4 日	305、401、402 403、404、405
三	5 日	501、502、503 504、505、601
四	6 日	602、603、604、605 701、702、703、704
五	7 日	705、801、802、803 804、805、901、902
六	8 日	903、904、905、1001 1002、1003、1004、1005

資料來源:本研究整理

#### 4.7 數學模式建構

在參考過業者實際需求及意見後，為了確保模型的實用性及合理性，針對目前實務上需求，在建構模式前提出以下幾個假設以利尋求更佳之排班結果。

1. 原始需求如船期表、每天各船舶數及各工作時段之司機員人力需求數為已知。
2. 模式中對於每日休假之司機員為已知，且並無額外請假部分。而實務上排班若有額外請假部分亦先可列入休假人員再求解排班。
3. 為了之後敏感度測試分析最適司機員人數，因此將本研究排班規劃作業以一週為期進行規劃。
4. 規劃期間中，司機員同一天同船舶同班次的工作時數相同，且無意外發生導致必須

更換司機員。

5. 每一位司機員在同時段只能服務一艘船之該班次，並無調度支援之情形。
6. 由於考慮值勤中精神上專注問題及工作負荷，司機員不可連續排兩個班次。
7. 司機員在完成該班次執勤後若有提早完工之情形，不管該班次工作時數多寡，皆視為一完整班次，不再重新指派。
8. 司機員同樣熟悉不同碼頭的所有機具。

本研究主要在實務上排班限制下，化成模式之限制式後，以追求分工平均為目的。在工時符合勞基法規定，一週工時不超過 48 小時為基準下，指標以規劃期中每位司機員指派之班次數總和為衡量依據。

Kim and Kim【】等人將此類司機排班問題構建為一混合整數規劃問題。而針對上述問題

題假設及排班概念下，本研究依其概念將目標式及限制式構建如下：



令  $i$  為規劃期間總天數的符號，規劃期天數為  $d$  天， $i=1, \dots, d$ 。  $j$  為規劃期中船隻數的符號， $j=1, \dots, n$ ，規劃天數第  $i$  日中到達的船隻為  $n_i$ 。  $k$  為規劃期間班次的符號， $k=1, \dots, s$ ，共有  $s$  班次。  $l$  為司機員的代表符號，司機共有  $o$  位， $l=1, \dots, o$ 。

**Min.**  $\omega$

**s.t**

$$\omega \geq \sum_{i=1}^d \sum_{j=1}^{n_i} \sum_{k=1}^s Y_{ijkl} \quad \forall l \quad (1)$$

$$\sum_{l=1}^o Y_{ijkl} (X_{ijk} \neq 0) \geq D_{ijk} \quad \forall i, j, k \quad (2)$$

$$\sum_{j=1}^{n_i} Y_{ijkl} (X_{ijk} = l) \leq 1 \quad \forall i, k, l \quad (3)$$

$$X_{ijk} = l, Y_{ijkl} = 1 \text{ and } Y_{ijkp} = 0, p \neq l \quad (4)$$

$$Y_{ijkl} = 1 \text{ or } 0 \quad (5)$$

$$Y_{ijkl} = 0 \quad \forall i, j, k, l \in (f_i) \quad (6)$$

$$\sum_{j=1}^{n_i} Y_{ijkl} + Y_{ij(k+1)l} \leq 1 \quad \forall i, l, k = 1, \dots, s-1 \quad (7)$$



$$\sum_{j=1}^{n_i} Y_{ijsl} + \sum_{j=1}^{n_{i+1}} Y_{(i+1)jll} \leq 1 \quad \forall i = 1, \dots, d-1, l, k \quad (8)$$

$Y_{ijkl}$ : 即代表第  $i$  工作天，第  $j$  艘船，第  $k$  個班次由編號第  $l$  位司機員執勤。

若有司機員排入班表，則  $Y_{ijkl} = 1$ ，反之則為 0。

$X_{ijk}$ : 代表在第  $i$  天，第  $j$  艘船隻，第  $k$  班次時，是由  $X_{ijk}$  此編號司機員排入該天

該船之該班次，亦即  $X_{ijk} = l$ 。若無需求即該天該船之該班次不排入司機

員， $X_{ijk} = 0$ 。

$D_{ijk}$ : 即為第  $i$  工作天，第  $j$  艘船，第  $k$  個班次中所需司機數目的需求。

$(f_i)$  為第  $i$  工作天休假人員之集合。

$\omega$  為總值勤班次數目中的最大值，意即為求取所有司機員一週總值勤班次數之最大值。目標式及限制式(1)的目的為除了尋求最大總值勤班次數也要求其最小化。在此目標式及限制式的要求及壓縮後，可使得作業人員的總值勤班次數平均化。

限制式(2)為需求滿足限制式，意即在第  $i$  天，第  $j$  艘船，第  $k$  個班次中安排的第  $l$  位司機員，必須滿足當天在該船舶下及該班次中所需司機數的需求量  $D_{ijk}$ 。

限制式(3)為時段衝突限制式，代表同一位司機員不可被安排至同一天同一班次的其他船舶執勤。

限制式(4)、(5)及(6)代表參數  $X_{ijk}$  有需求  $l$  時， $Y_{ijkl}=1$ ，即為有需求，反之則為  $Y_{ijkl}=0$ ，待命休假狀態。

限制式(7)為因應高空作業需耗費大量專注力，避免連續執勤，限制式即規劃不可連續執勤兩班次之限制式。

限制式(8)為限制式(7)之補強。單只有限制式(7)無法保證在過夜後隔天是否連續執勤，意即若無限制式(8)，有可能出現 12 月 2 日值勤最後一個班次晚班，而 12 月 3 日小夜班繼續執勤，此為保證換日後絕對不會連續執勤班次。

## 第五章 實證分析

本研究以高雄港 79 號、80 號、81 碼頭為對象，與業者原排班對象相同，而 80 號碼頭及 81 號碼頭為連號碼頭且皆位於同一邊，機具可互相支援以因應較大之船泊靠港時使用。以 E 公司一週船期之作業司機員人力需求作為實例分析，測試本研究建構模式之求解效率。本模式以 C 語言程式編寫目標式及限制式以產生展開限制式輸出產生器，並使用 ILOG Solver 之數學規劃套裝軟體 ILOG CPLEX 10.0 求解，測試平台為 Microsoft Windows XP 作業系統，處理器為 Intel Pentium 4.2G Hz CPU、512MB DDR RAM。

### 5.1 測試及輸入資料背景

由於 E 業者所租用碼頭服務之船舶大多為定期靠泊，而船期表皆為一週前可得知，亦可從港務局得到資料，所以本研究以一週時間為規劃期。輸入資料部分比照原排班方式所用資料，方可比較其改善情形。輸入資料之船期如表 3.6，時間為自 2007 年 12 月 2 日至 12 月 8 日為期一週之船期表。工作時間班次如表 3.2，仍沿用原排班方式，將一天分為五個班次，一樣為小夜班(00:00-04:00)、大夜班(04:00-08:00)、早班(08:00-14:00)、中班(14:00-20:00)及晚班(20:00-00:00)。司機員之一週執勤休假表如表 4.1。最後由已知之原始船期表統計出每天每船每班次之人力需求部分，一週中人力需求整理如表 5.1 所示。



表 5.1 一周司機員人力需求表

BOX	ETB	ETD	碼頭	GC'S	WORK HOURS	小夜班 00:00-04:00	大夜班 04:00-08:00	早班 08:00-14:00	中班 14:00-20:00	晚班 20:00-00:00
990	12/2 01:00	12/2 19:00	81 號	2	18	2	2	2	2	
1360	12/2 03:00	12/2 19:00	80 號	3	16	3	3	3	3	
1160	12/2 15:00		79 號	4	10.5				4	4
		12/3 01:35	79 號	4	10.5	4				
2530	12/3 08:00		79 號	4	23			4	4	4
		12/4 07:00	79 號	4	23	4	4			
2240	12/3 09:30		80 號	5	16			5	5	5
		12/4 01:30	80 號	5	16	5				
2520	12/4 02:30	12/4 20:00	80 號	5	17.5	5	5	5	5	
1490	12/4 08:00	12/4 21:30	79 號	4	13.5			4	4	4
3360	12/5 00:30		80 號	5	23.5	5	5	5	5	5
		12/6 01:00	80 號	5	23.5	5				
1650	12/5 08:00	12/5 23:00	79 號	4	15			4	4	4
250	12/7 00:20	12/7 05:00	80 號	2	4.5	2	2			
980	12/7 00:30	12/7 12:00	81 號	3	11	3	3	3		
1960	12/7 08:00	12/7 22:00	81 號	5	14			2	5	5
1260	12/7 22:00		79 號	4	9					4
		12/8 07:00	79 號	4	9	4	4			
1020	12/8 11:25	12/8 23:30	80 號	3	12			3	3	3
660	12/8 12:30	12/9 00:00	81 號	2	12			2	2	2

資料來源:航運公司提供、本研究整理



上述資料及模式在利用 C 語言程式編寫目標式及限制式生成 text 檔案，再輸入各需求輸入檔，最後生成輸出限制式展開之編碼檔案。由於程式內容多達 160 多頁，故只節錄其中部分編碼作為參考。

TEXT 檔案:

```
#include <stdlib.h>

#include <stdio.h>

#define DAY 7      /*全部天數為 7 天*/

#define SLOT 5     /*SLOT 有 5 個班次*/

#define GROUP 50  /*可供排班有 50 位人員*/

#define FDEMAND1 "D:\\論程式\\TEST 包\\最終版\\demand1.txt"
#define FDEMAND2 "D:\\論程式\\TEST 包\\最終版\\demand2.txt"
#define FDEMAND3 "D:\\論程式\\TEST 包\\最終版\\demand3.txt"
#define FDEMAND4 "D:\\論程式\\TEST 包\\最終版\\demand4.txt"
#define FDEMAND5 "D:\\論程式\\TEST 包\\最終版\\demand5.txt"
#define FDEMAND6 "D:\\論程式\\TEST 包\\最終版\\demand6.txt"
#define FDEMAND7 "D:\\論程式\\TEST 包\\最終版\\demand7.txt"

int *day;

int *ship;

int **demand1=NULL;

int **demand2=NULL;

int **demand3=NULL;

int **demand4=NULL;

int **demand5=NULL;

int **demand6=NULL;
```

```

int **demand7=NULL;

int **off=NULL;

int shiftkind[SHIFT]={1,2,3,4,5};

int i,j,k,l;

int d;

void read_all_demand(void);           /*讀入所有的資料*/

void write_model(void);              /*印出所有的 model*/

void read_day(char *);               /*給予天數代號，如第一天為 day[0]等*/

void read_ship(char *);              /*ship 函數*/

void requiredemand_space(int ***,int); /*各天需求的空間*/

void read_demand(char *,int,int **); /*讀取各天的需求*/

void read_rest(int ***);

int objective(FILE *model);          /*宣告目標式函數*/

int a(void);

void min_group(FILE *model);

void work_need(int,int,int **,FILE *model); /*宣告工作時段需求限制式*/

void time_conflict(FILE *model);      /*時段衝突限制式*/

void holiday(FILE *model);            /*輪休限制式*/

void continue_work(FILE *model,int,int); /*連續執勤限制式*/

void day_no_work(FILE *model);        /*隔日不立刻勤執限制式*/

int c(void);

void end(FILE *model);

int main(void)

{

    read_all_demand();

```



```

write_model();

return 0;

}

```

生成之 MODEL 展開式檔案，以限制式(1)及限制式(2)為例：

**限制式(1):**

$$\begin{aligned}
&+Y_{1_1_1_1}+Y_{1_1_2_1}+Y_{1_1_3_1}+Y_{1_1_4_1}+Y_{1_1_5_1}+Y_{1_2_1_1}+ \\
&Y_{1_2_2_1}+Y_{1_2_3_1}+Y_{1_2_4_1}+Y_{1_2_5_1}+Y_{1_3_1_1}+Y_{1_3_2_1}+ \\
&Y_{1_3_3_1}+Y_{1_3_4_1}+Y_{1_3_5_1}+Y_{2_1_1_1}+Y_{2_1_2_1}+Y_{2_1_3_1}+ \\
&Y_{2_1_4_1}+Y_{2_1_5_1}+Y_{2_2_1_1}+Y_{2_2_2_1}+Y_{2_2_3_1}+Y_{2_2_4_1}+ \\
&Y_{2_2_5_1}+Y_{3_1_1_1}+Y_{3_1_2_1}+Y_{3_1_3_1}+Y_{3_1_4_1}+Y_{3_1_5_1}+ \\
&Y_{3_2_1_1}+Y_{3_2_2_1}+Y_{3_2_3_1}+Y_{3_2_4_1}+Y_{3_2_5_1}+Y_{4_1_1_1}+ \\
&Y_{4_1_2_1}+Y_{4_1_3_1}+Y_{4_1_4_1}+Y_{4_1_5_1}+Y_{6_1_1_1}+Y_{6_1_2_1}+ \\
&Y_{6_1_3_1}+Y_{6_1_4_1}+Y_{6_1_5_1}+Y_{6_2_1_1}+Y_{6_2_2_1}+Y_{6_2_3_1}+ \\
&Y_{6_2_4_1}+Y_{6_2_5_1}+Y_{6_3_1_1}+Y_{6_3_2_1}+Y_{6_3_3_1}+Y_{6_3_4_1}+ \\
&Y_{6_3_5_1}+Y_{6_4_1_1}+Y_{6_4_2_1}+Y_{6_4_3_1}+Y_{6_4_4_1}+Y_{6_4_5_1}+ \\
&Y_{7_1_1_1}+Y_{7_1_2_1}+Y_{7_1_3_1}+Y_{7_1_4_1}+Y_{7_1_5_1}+Y_{7_2_1_1}+ \\
&Y_{7_2_2_1}+Y_{7_2_3_1}+Y_{7_2_4_1}+Y_{7_2_5_1}-W \leq 0
\end{aligned}$$

上列展開式為限制式第一條，以司機編號 1 為例。上列式子代表總合編號 1 之司機員之班次必須小於總合平均班次數之目標式 W。(在程式中以 W 代表  $\omega$ )。

**限制式(2):**

$$+Y_{1202_1_1_1}+Y_{1202_1_1_2}+Y_{1202_1_1_3}+Y_{1202_1_1_4}+Y_{1202_1_1_5}+Y$$

1202\_1\_1\_6+ Y 1202\_1\_1\_7+ Y 1202\_1\_1\_8+ Y 1202\_1\_1\_9+ Y 1202\_1\_1\_10+ Y  
1202\_1\_1\_11+ Y 1202\_1\_1\_12+ Y 1202\_1\_1\_13+ Y 1202\_1\_1\_14+ Y 1202\_1\_1\_15+ Y  
1202\_1\_1\_16+ Y 1202\_1\_1\_17+ Y 1202\_1\_1\_18+ Y 1202\_1\_1\_19+ Y 1202\_1\_1\_20+ Y  
1202\_1\_1\_21+ Y 1202\_1\_1\_22+ Y 1202\_1\_1\_23+ Y 1202\_1\_1\_24+ Y 1202\_1\_1\_25+ Y  
1202\_1\_1\_26+ Y 1202\_1\_1\_27+ Y 1202\_1\_1\_28+ Y 1202\_1\_1\_29+ Y 1202\_1\_1\_30+ Y  
1202\_1\_1\_31+ Y 1202\_1\_1\_32+ Y 1202\_1\_1\_33+ Y 1202\_1\_1\_34+ Y 1202\_1\_1\_35+ Y  
1202\_1\_1\_36+ Y 1202\_1\_1\_37+ Y 1202\_1\_1\_38+ Y 1202\_1\_1\_39+ Y 1202\_1\_1\_40+ Y  
1202\_1\_1\_41+ Y 1202\_1\_1\_42+ Y 1202\_1\_1\_43+ Y 1202\_1\_1\_44+ Y 1202\_1\_1\_45+ Y  
1202\_1\_1\_46+ Y 1202\_1\_1\_47+ Y 1202\_1\_1\_48+ Y 1202\_1\_1\_49+ Y 1202\_1\_1\_50=2

上列展開式為限制式第二條，以第一天 12 月 2 日，第一艘船舶，第一個班次為例。  
在該天該船舶該班次之需求為 2 個司機員。



以上述展開式加入標頭檔及限制整數規劃之參數後輸入 CPLEX 後，共產生 15380 條限制式，26250 個變數，求解時間為 0.16 秒。輸出後得到數學規劃之班表結果如表 5.2 所示。

表 5.2 數學規劃所得一周班表

船名 VVD	BOX	ETB	ETD	星期	休假人員	碼頭	GC'S	WORK HOURS	小夜班 00:00-04:00	大夜班 04:00-08:00	早班 08:00-14:00	中班 14:00-20:00	晚班 20:00-00:00	
A	990	12/2 01:00	12/2 19:00	日	101,102 103,104 105,201 202,203	81 號	2	18	401,604	205,301	204,801	305,704		
B	1360	12/2 03:00	12/2 19:00			80 號	3	16	304,402 803	405,502 701	404,504 802	602,705 805		
C	1160	12/2 15:00				79 號	4	10.5					304,501 605,702	302,603 703,1001
C	1160		12/3 01:35	一	204,205 301,302 303,304	79 號	4	10.5	403,503 601,902					
D	2530	12/3 08:00				79 號	4	23			201,402 901,1003	102,501 505,902	105,401 905,1005	
E	2240	12/3 09:30				80 號	5	16			103,404 503,903 1004	101,203 305,403 904	104,202 405,502 1002	
D			12/4 07:00	二	305,401 402,403 404,405	79 號	4	23	201,304 505,702	604,701 704,902				
E			12/4 01:30			80 號	5	16	504,605 703,705 801					
F	2520	12/4 02:30	12/4 20:00			80 號	5	17.5	104,301 503,804 1004	204,501 603,903 904	101,202 504,803 1005	102,205 601,802 901		
G	1490	12/4 08:00	12/4 21:30			79 號	4	13.5				103,203 905,1002	302,303 502,602	105,803 1001 1003
H	3360	12/5 00:30		三	501,502 503,504 505,601	80 號	5	23.5	101,203 304,401 604	302,704 804,901 1001	201,301 705,802 1004	402,603 701,902 904	105,703 804,905 1002	
I	1650	12/5 08:00	12/5 23:00			79 號	4	15			102,103 305,602	104,202 803,1003	303,204 702,801	
H			12/6 01:00	四	602,603 604,605 701,702 703,704	80 號	5	23.5	205,505 601,805 1005					
J	250	12/7 00:20	12/7 05:00	五	705,801 802,803 804,805 901,902	80 號	2	4.5	402,605	404,903				
K	980	12/7 00:30	12/7 12:00			81 號	3	11	501,602 1001	504,702 1004	505,605 1002			
L	1960	12/7 08:00	12/7 22:00			81 號	5	14				403,503	101,405 601,704 904	604,701 903,905 1003
M	1260	12/7 22:00				79 號	4	9						102,301 703,1005
M			12/8 07:00	六	903,904 905,1001 1002,1003 1004,1005	79 號	4	9	105,401 603,805	103,203 205,804				
N	1020	12/8 11:25	12/8 23:30			80 號	3	12			302,801 402	201,305 405	502,802 901	
O	660	12/8 12:30	12/9 00:00			81 號	2	12				104,404	204,303	202,403

資料來源:本研究整理

以  $X_{111}$  為例，即為規劃期第一天星期日、第一艘船 A 船、第一個工作時段小夜班，執勤需求為兩位司機員，分別由司機員編號 401、604 值勤。該天該船之第二個工作時段由司機員編號 205、301 值勤，以此類推。

而總和模式輸出班表之結果經統計後，將每位司機的總值勤班次數目做一整理，得出每位司機員一週總值勤班次如表 5.3 所示。

表 5.3 司機員一周總值勤班次數

司機編號	<b>101</b>	<b>102</b>	<b>103</b>	<b>104</b>	<b>105</b>	<b>201</b>	<b>202</b>	<b>203</b>	<b>204</b>	<b>205</b>
班次	4	4	4	4	4	4	4	4	4	4
司機編號	<b>301</b>	<b>302</b>	<b>303</b>	<b>304</b>	<b>305</b>	<b>401</b>	<b>402</b>	<b>403</b>	<b>404</b>	<b>405</b>
班次	4	4	3	4	4	4	4	4	4	4
司機編號	<b>501</b>	<b>502</b>	<b>503</b>	<b>504</b>	<b>505</b>	<b>601</b>	<b>602</b>	<b>603</b>	<b>604</b>	<b>605</b>
班次	4	4	4	4	4	4	4	4	4	4
司機編號	<b>701</b>	<b>702</b>	<b>703</b>	<b>704</b>	<b>705</b>	<b>801</b>	<b>802</b>	<b>803</b>	<b>804</b>	<b>805</b>
班次	4	4	4	4	4	4	4	4	4	3
司機編號	<b>901</b>	<b>902</b>	<b>903</b>	<b>904</b>	<b>905</b>	<b>1001</b>	<b>1002</b>	<b>1003</b>	<b>1004</b>	<b>1005</b>
班次	4	4	4	4	4	4	4	4	4	4

資料來源:本研究整理

總班次需求數為 198 個班次，每位司機員平均值勤班次為 3.96 個班次，值勤班次之變異數為 0.038。從統計資料可得知，每位司機員之一週值勤班次數都界於 3 個班次到 4 個班次之間。

相較於業者之規劃方法，值勤之最多班次及最少班次總數達 6 個班次之多之結果而言，由模式輸出所得到的班表在工作的均勻分配上有良好的改善結果。

表 5.4 數學規劃及原始班表之比較

比較項目 班表類別	總值勤班次數	值勤班次變異數	值勤班次之最多 及最少之差異
原始班表	由 1 個班次到 7 個 班次不等	1.81	6 個班次
數學規劃之班表	集中於 3-4 個班次	0.038	1 個班次

資料來源:本研究整理

而經由數學整數規劃模式所求解之班表，除了在排班上人員較無順序性外，就本研究所設置的目標式，亦即每位司機員平均班次數而言，有其改善效果。而在安排司機員之順序較無軌跡可循是因在整數規劃模式中尋求最佳化解時並沒有將特定司機員綁定於某座碼頭之機具及特定之值勤班次時間上。

另外，求解之可行解及最佳解還尚存很多組空間，即代表目前的求解空間較鬆散也較自由，此輸出最佳化班表只為其中一組最佳解。由此可見，原始給定的人力資源之司機員人數在此安排作業下仍有充足的空間可供指派。

由上述資料之分析，不難發現目前所有的司機員在勞力方面非常充裕。以一般公司企業為基準，周休二日，每日工作八小時為標準也仍有大量多餘的閒置時間(Idle Time)。其中夜班時間如晚班、小夜班、大夜班值勤一個班次只有四個小時，白日班如早班及中班一個班次為六個小時。以業者角度安排的班次規劃認為夜班因天色較暗，需耗費更多精神力於工作上，故認定夜班一個班次四小時等同於白日班一個班次六小時。故本研究在計算值勤時數上，將夜班加權換算成白日班次的工作時數，意即夜班的一個工作小時為白日班 1.5 個工作小時。由輸出之班表進一步分析統計每位司機員的總值勤工時及閒置時間，如表 5.5 所示。



表 5.5 每位司機員一周總值勤時間(小時)

司機編號		101	102	103	104	105	201	202	203	204	205
班次 工作時數	日班	6+6+6	6+4+6	4.5+6+6	6+1.5	0	6+6+6	6+6	6+6	6+6	4
	夜班	3.5	2	3	4+1.5	4+1.5+4+4	4	4+4	3.5+3	4+3	4+1+3
總值勤時數		23.25	19	21	15.75	20.25	24	24	21.75	22.5	16
IDEL		24.25	29	27	32.25	27.25	24	24	26.25	25.5	32
司機編號		301	302	303	304	305	401	402	403	404	405
班次 工作時數	日班	6	2.5+6	6+6	5	5+6+6+6	0	6+6	6+6	6+4.5+6	6+6
	夜班	4+1.5+2	4+4	3	3+4+3.5	0	3+4+3.5+4	1+3.5	1.5+4	3	4+4
總值勤時數		17.25	20.5	16.5	20.75	23	21.75	18.75	20.25	21	24
IDEL		30.75	27.5	31.5	27.25	25	26.25	29.25	27.75	23	24
司機編號		501	502	503	504	505	601	602	603	604	605
班次 工作時數	日班	5+6	6	4.5+6	6+6	6+4	4+6	5+6+6	6	0	5+4
	夜班	4+3.5	4+4+3.5	1.5+1.5	1.5+4	4+1	1.5+1	3.5	4+4+4	3+3+3.5+2	1.5+3.5
總值勤時數		22.25	23.25	15	20.25	17.5	13.75	22.25	24	17.25	16.5
IDEL		25.75	24.75	33	27.75	30.5	34.25	25.75	24	30.75	31.5
司機編號		701	702	703	704	705	801	802	803	804	805
班次 工作時數	日班	6	5	0	5+6	5+6+2.5	6+2.5	6+4+6	6+6	0	5
	夜班	4+3+2	4+3+4	4+1.5+4+2	3+4	1.5	1.5+3	3.5	1+1.5	1.5+4+6+3	3+4
總值勤時數		19.5	21.5	17.25	21.5	15.75	15.25	21.25	15.75	21.75	15.5
IDEL		28.5	26.5	30.75	26.5	32.25	32.75	26.75	32.25	26.25	35.5
司機編號		901	902	903	904	905	1001	1002	1003	1004	1005
班次 工作時數	日班	6+4	6+6	4.5	6+6+6	6	0	6+4	6+6	4.5+6	6
	夜班	4+3.5	1.5+3	4+1+2	4	4+4+2	4+1.5+4+3.5	4+4	1.5+2	1.5+4	4+1+2
總值勤時數		21.25	18.75	15	24	21	19.5	17.25	17.25	18.75	16.5
IDEL		26.75	29.25	33	24	27	28.5	30.75	30.75	29.75	31.5

資料來源:本研究整理

## 5.2 作業司機員人數變動之敏感度分析

從班表結果及執勤時間統計顯示，目前該公司的作業司機員每週總值勤時間之平均範圍約為 15 個小時至 23 個小時不等，距離勞基法規定，一週工作時數不得超過 48 小時，此班表在人力執勤尚有大量空閒時間可供進一步改善。總和前述分析，E 公司在司機員指派作業之人力數有過剩的情形。

本節嘗試將人數縮減為 45 人後，再次經過軟體輸出班表得到結果統計後，每位司機員值勤班次數如表 5.6 所示。每位司機員一週值勤班次數介於 4 個班次至 5 個班次。平均值勤班次數為 4.4，總值勤班次之變異數為 0.24。

表 5.6 45 位司機員人數之一週值勤班次數

司機編號	<b>101</b>	<b>102</b>	<b>103</b>	<b>104</b>	<b>105</b>	<b>201</b>	<b>202</b>	<b>203</b>	<b>204</b>	<b>205</b>
班次	5	5	4	4	5	4	5	4	5	4
司機編號	<b>301</b>	<b>302</b>	<b>303</b>	<b>304</b>	<b>305</b>	<b>401</b>	<b>402</b>	<b>403</b>	<b>404</b>	<b>405</b>
班次	4	4	4	5	5	4	5	4	4	5
司機編號	<b>501</b>	<b>502</b>	<b>503</b>	<b>504</b>	<b>505</b>	<b>601</b>	<b>602</b>	<b>603</b>	<b>604</b>	<b>605</b>
班次	4	5	4	4	5	4	4	4	5	4
司機編號	<b>701</b>	<b>702</b>	<b>703</b>	<b>704</b>	<b>705</b>	<b>801</b>	<b>802</b>	<b>803</b>	<b>804</b>	<b>805</b>
班次	4	5	4	4	5	4	5	4	4	5
司機編號	<b>901</b>	<b>902</b>	<b>903</b>	<b>904</b>	<b>905</b>					
班次	4	5	4	5	4					

資料來源:本研究整理

再將司機員減少至 40 位後，結果如表 5.7，每位司機員一週執勤班次數亦在 4 個班次至 5 個班次之間，平均值勤班次數為 4.95，而變異數下降至 0.0475。而工作時間方面雖在勞基法規定之內，但是工作最多之時數與最少之時數介於 30 個小時至 20.5 小時之間，以最多 30 小時來看，距離規定還有可改善空間。

表 5.7 40 位司機員人數之一週值勤班次數

司機編號	<b>101</b>	<b>102</b>	<b>103</b>	<b>104</b>	<b>105</b>	<b>201</b>	<b>202</b>	<b>203</b>	<b>204</b>	<b>205</b>
班次	5	5	5	5	5	5	5	5	5	5
司機編號	<b>301</b>	<b>302</b>	<b>303</b>	<b>304</b>	<b>305</b>	<b>401</b>	<b>402</b>	<b>403</b>	<b>404</b>	<b>405</b>
班次	4	5	5	5	5	5	5	5	5	5
司機編號	<b>501</b>	<b>502</b>	<b>503</b>	<b>504</b>	<b>505</b>	<b>601</b>	<b>602</b>	<b>603</b>	<b>604</b>	<b>605</b>
班次	5	5	5	5	5	5	5	5	5	5
司機編號	<b>701</b>	<b>702</b>	<b>703</b>	<b>704</b>	<b>705</b>	<b>801</b>	<b>802</b>	<b>803</b>	<b>804</b>	<b>805</b>
班次	5	5	5	5	5	5	4	5	5	5

資料來源:本研究整理

再次嘗試將人數減少為 35 人，輸出結果如表 5.8。每位司機員一週值勤班次數範圍介於 5 個班次至 6 個班次，值勤班次之平均數為 5.65，變異數為 0.306。

平均工時為 29 個小時，值勤最多為 35.25 小時，最少則為 25.75 小時。雖然變異數上升了，但是仍可在接受範圍之內，而司機員間之一週值勤班次數也落差也僅於 1 個班次。若再將人數刪減，則會出現一周最多值勤超過 40 個小時的狀況，雖說仍在勞基法規定 1 週內不得工作超過 48 小時，但以一般企業一周休兩天，總工時約 40 小時的基本來看，人數若再減少，則會出現工作超時的情形。

表 5.8 35 位司機員人數之一週值勤班次數

司機編號	<b>101</b>	<b>102</b>	<b>103</b>	<b>104</b>	<b>105</b>	<b>201</b>	<b>202</b>	<b>203</b>	<b>204</b>	<b>205</b>
班次	6	6	6	5	6	6	5	5	6	5
司機編號	<b>301</b>	<b>302</b>	<b>303</b>	<b>304</b>	<b>305</b>	<b>401</b>	<b>402</b>	<b>403</b>	<b>404</b>	<b>405</b>
班次	6	6	5	5	6	6	6	6	6	5
司機編號	<b>501</b>	<b>502</b>	<b>503</b>	<b>504</b>	<b>505</b>	<b>601</b>	<b>602</b>	<b>603</b>	<b>604</b>	<b>605</b>
班次	5	6	6	6	5	6	6	6	6	6
司機編號	<b>701</b>	<b>702</b>	<b>703</b>	<b>704</b>	<b>705</b>					
班次	5	6	5	5	6					

資料來源:本研究整理

表 5.9 司機員人數之敏感度分析

比較項目 排班方法		每人一週值 勤班次數	每人值勤班 次數差異	一週值勤班 次變異數	一週工作時數(小時)		
					平均	最多	最少
原始班表		1-7 個班次 不等	6 個班次	1.18	20.24	41.5	6
數學 規劃	50 位	3-4 個班次	1 個班次	0.038	20.24	24	15
	45 位	4-5 個班次	1 個班次	0.24	22.48	28.5	20.25
	40 位	4-5 個班次	1 個班次	0.048	25.3	30	20.5
	35 位	5-6 個班次	1 個班次	0.306	29	35.25	25.75

資料來源:本研究整理

### 5.3 連休排班模式



由於這樣排班制的工作，有大部份的員工再休假方面傾向於一次將一週的工作份量提前做完，而剩下的休息時間排連休。考慮此因素，本研究對此部分員工之考量提供另一種休假模式之決策。在本研究之數學模型中，額外增加一條目標式，主要目的在於將連續天數中之工作量極大化，其餘限制式及目標式不變。

Max.

$$\sum_{j=1}^{n_i} \sum_{k=1}^s Y_{ijkl} + \sum_{j=1}^{n_{i+1}} \sum_{k=1}^s Y_{(i+1)jkl} + \sum_{j=1}^{n_{i+2}} \sum_{k=1}^s Y_{(i+2)jkl} + \sum_{j=1}^{n_{i+3}} \sum_{k=1}^s Y_{(i+3)jkl} \quad \forall i=1, \dots, d-3, l$$

因模式內並無規劃工作於特定時間或連續天數內，排班之結果則會較分散(如表

5.2)，此為較為一般的排班方式。若將連續某幾天內之值勤班次極大化，加上原本限制式將總值勤班次極小化，則可將司機員之值勤班次集中於某連續數天內，並且符合限制式之規劃，不會連續執勤兩個時段。在此壓縮下，工作班次被集中於連續數天，而剩下未被指定之天數則沒有班次成為休假，加上原本的即可規劃為連休。由於 35 人的排班約為每位司機員執勤 5 至 6 個班次，約為連續執勤三至四天，每一天執勤一個或兩個班次左右，故本研究模式中，此最大化目標式採用連續特定四天( $i, i+1, i+2, i+3$ )。以同樣 35 人的人數及需求以及其他相同的以之資料及條件下，再次以軟體進行排班作業，結果如表 5.10



表 5.10 連休之排班結果

船名 VVD	BOX	ETB	ETD	星期	休假人員	碼頭	GC'S	WORK HOURS	小夜班 00:00-04:00	大夜班 04:00-08:00	早班 08:00-14:00	中班 14:00-20:00	晚班 20:00-00:00	
A	990	12/2 01:00	12/2 19:00	日	101,102 103,104 105,201 202,203	81 號	2	18	205,703	201,303	305,501	301,402		
B	1360	12/2 03:00	12/2 19:00			80 號	3	16	302,401 705	301,601 704	304,503 604	201,303 602		
C	1160	12/2 15:00				79 號	4	10.5					205,302 403,605	401,502 604,705
C	1160		12/3 01:35	一	204,205 301,302 303,304	79 號	4	10.5	102,303 602,704					
D	2530	12/3 08:00				79 號	4	23			102,301 403,604	203,304 602,704	301,305 405,604	
E	2240	12/3 09:30				80 號	5	16			302,402 405,504 703	401,503 505,603 705	303,402 502,504 703	
D			12/4 07:00	二	305,401 402,403 404,405	79 號	4	23	104,403 501,702	101,203 304,404				
E			12/4 01:30			80 號	5	16	102,502 601,605 704					
F	2520	12/4 02:30	12/4 20:00			80 號	5	17.5	304,402 503,604 701	305,602 603,703 705	104,402 502,601 702	101,304 403,602 701		
G	1490	12/4 08:00	12/4 21:30			79 號	4	13.5			102,501 505,704	203,404 603,603	305,501 505,604	
H	3360	12/5 00:30		三	501,502 503,504 505,601	80 號	5	23.5	102,202 603,704 705	101,104 203,404 701	103,403 601,603 704	202,404 501,605 703	104,403 502,601 702	
I	1650	12/5 08:00	12/5 23:00			79 號	4	15			203,305 502,702	105,204 404,701	103,203 305,603	
H			12/6 01:00	四	602,603 604,605 701,702 703,704	80 號	5	23.5	101,105 204,601 702					
J	250	12/7 00:20	12/7 05:00	五	705,801 802,803 804,805 901,902	80 號	2	4.5	104,201	103,105				
K	980	12/7 00:30	12/7 12:00			81 號	3	11	202,301 504	204,303 405	101,302 605			
L	1960	12/7 08:00	12/7 22:00			81 號	5	14			104,205	103,201 204,401 702	101,105 202,504 701	
M	1260	12/7 22:00				79 號	4	9					205,303 405,605	
M			12/8 07:00	六	903,904 905,1001 1002,1003 1004,1005	79 號	4	9	201,302 402,503	103,202 205,505				
N	1020	12/8 11:25	12/8 23:30			80 號	3	12			105,301 504	201,405 505	103,204 401	
O	660	12/8 12:30	12/9 00:00			81 號	2	12			401,602	205,503	105,302	

資料來源:本研究整理

在此連休模型下，得到的工作班次數與先前 35 人未連休的結果大同小異，值勤班次仍為 5 個至 6 個，並未增多或減少，唯值勤班次較為緊湊，集中於特定之連續數天內。此結果也符合原先限制之不可連續排班，雖然會出現中間間隔休息 4 至 6 個小時又繼續值班的狀況，此為司機員偏好連休之模型。

#### 5.4 小結

總合以上所有分析，得知本研究所構建之整數規劃排班模型，對於司機員排班上之追求班次平均的目標來說有大幅的改善結果，且經由軟體求解之速度都在一秒內就可得到與原實務單位耗時排班之結果，輸出之班表更為明顯改善分工不均之情形。而對於人員過剩之情形也可以由結果中得到較佳的經濟人數組合。另外，對於司機員偏好連休的部分，也提供公司往後排班作業中一種休假模式。此司機員排班結果可提供給 E 公司作日後排班或研究參考。





## 第六章 結論與建議

本研究主要針對航運公司之起重機司機作業人員排班方式及流程與以深入了解，結合整數規劃法構建數學模式，以求解得司機員班表最佳化之問題。綜合前述幾章的探討，提出以下結論，並對未來的研究提出建議。

### 6.1 結論

1. 以業者目前司機員指派作業之現況而言，原排班方式出現分工不均的情況，如一週內有值勤班次數高達 7 個班次，且一週中只值勤 1 個班次的司機員也有三位。最多與最少班次之落差達 6 個班次，變異數為 1.18。而值勤之工作時間最多為 41.5 小時，最少為 6 個小時，平均每位司機工作時間為 20 小時，司機員在值勤時數方面非常少，人力資源過剩的情形有待改善。
2. 以數學規劃方式，構建涵蓋排班原則與限制之司機員排班模式可滿足原本以經驗法則之人力指派方式所面臨之限制。如安排值勤人員、休假人員，以及不連續執勤以避免工時過長可能引起安全意外之考量。
3. 本研究於排班求解效率上可透過電腦化排班及套裝軟體 ILOG CPLEX 求解，並且在短時間內完成，節省人工排班所耗費的時間。當往後模式及求解的規模擴大時，以電腦化排班的效率更加明顯。
4. 經數學規劃改善過後，每一位司機員之值勤班次集中於 3 個班次至 4 個班次，在總值勤班次數方面顯得較為平均，相差不到 1 個班次。一週值勤班次之變異數降為 0.038，相較於原值勤班表之班次數不均的情況能有效改善，達到分工較為均勻的結果。
5. 本研究於最適人數之分析結果可在符合勞基法之規定下，將原本過多的 50 位司機員縮減至 35 位。每位司機員執勤班次集中於 5 個班次至 6 個班次，每週值勤班次

變異數為 0.306，雖然比人數為 40 人或者 45 人之班次變異數高一些，但仍可在接受範圍之內，每位司機員之一週值勤班次數落差也僅為一班，與其他測試人數相同，並且滿足模式中之限制。

6. 本研究從司機員角度，考慮值勤班次平均化，也兼顧業主角度可將司機員達到更經濟的人數，減少人力支出，使得電腦化排班之應用在實務上更具可行及實用性。

## 6.2 建議

1. 本研究之模式係假設在船期及裝卸櫃量為已知之前提下進行排班作業，後續研究可以考慮將船期規劃與人員排班一併納入，以擴展模式的功能。
2. 本研究在指派排班作業中將人員操作熟練度以及起重機機具之裝卸效率視為一致，建議未來可對其差異進行考量，以提升模式的功能。
3. 在本研究中，休假方式嘗試以不同方法安排，建議後續研究可研擬不同之休假方案進一步測試不同的休假方式所得到之結果，使排班作業更為完備。
4. 後續研究可深入探討海運中貨櫃量之淡季與旺季之人力需求，以完整規劃人力需求與調度。
5. 實務單位若能採用電腦化排班以替代目前的人工排班方式，將能有效提升工作效率與排班之平均化。

## 參考文獻

1. 盧華安、林正偉，「岸肩橋式起重機工作排程之研運輸計劃季刊」，運輸計劃季刊，第三十五卷，第三期，頁 281-308，民國 95 年。
2. 杜宇平、顏上堯，「空服員排班網路模式與求解演算法之研究」，運輸學刊，第十四卷第二期，頁 15-29，民國 91 年。
3. 陳立欣，「考慮平均化之後艙組員派遣模式---以座艙長為例」，碩士論文，國立交通大學。
4. 蘇昭銘、張靖，「捷運系統站務人員排班模式之研究」，運輸學刊，第十二卷，第二期，頁 1-14，民國 89 年。
5. 陳豪雷，「長途客運排程規劃暨班次表設計之研究」，中央大學土木工程學系，碩士論文，民國 88 年。
6. 顏上堯、林錦翌，「空服員排班組合最佳化之研究」，中國土木水利工程學刊，第 9 卷 2 期，頁 303-313，民國 86 年。
7. 高建元，「護理人員之排班研究」，國立台灣科技大學管理技術研究所，民國 83 年。
8. 陳柏榮、王國琛，限制規劃 Constraint Programming 之應用研究 - 以排課問題為例，國立交通大學運輸工程與管理學系畢業專題，民國 89 年。
9. 辛怡達、林正章，郵局快捷專車排班排程問題之研究，國立成功大學交通管理科學研究所碩士班論文，民國 90 年。
10. 謝玉霜、陳春益，限制式規劃應用於港區貨櫃場軌道式門型起重機移動路徑之研究，國立成功大學交通管理科學研究所碩士論文，民國 90 年。
11. 林詩芹、韓復華，以限制規劃構建全年無休服務人員排班模式 - 以客服人員排班為例，國立交通大學運輸科技與管理學系，碩士論文，民國 92 年。
12. 翁得榮、喻奉天，排課問題之研究 - 以高雄第一科技大學運籌管理系為例，高雄第一科技大學，民國 96 年。

13. ILOG OPL: Optimization Programming Language , 皮托科技訓練課程教材 , 民國 90 年。
14. Abboud, N., M. Inuiguchi, M. Sakawa and Y. Uemura, “Manpower allocation using genetic annealing,” *European Journal of Operational Research*, Vol. 111, pp. 405-420, 1998.
15. A. Caprera, M. Fischetti, P. Toth, D. Vigo, and P. L Guida , *Algorithms for Railway Crew Management, Mathematical Programming*, Vol. 79, pp. 125-141, 1997.
16. Appelgren, L. H., “A Column Generation Algorithm for a Ship Scheduling Problem”, *Transportation Science*, 3, pp. 53-68, 1969
17. Bartholdi, J.J., “A Guaranteed-Accuracy Round-off Algorithm for Cyclic Scheduling and Set Covering,” *Operations Research*, Vol.29, pp.501-510, 1981.
18. Beasley J.E and Cao B. , A tree Search Algorithm for the Crew Scheduling Problem , *European Journal of Operational Research* , Vol. 94 , No.3 , pp517-526(1996)。
19. Caprera, Alberto, Fischetti, Matteo, Toth, Paolo and Vigo, Daniele, “Algorithms for Railway Crew Management”, *Mathematical Programming*, Vol. 79, pp.125-141, 1997.
20. Cheng, B., J. Lee, and J. Wu, “A Nurse Rostering System Using Constraint Programming and Redundant Modeling,” *IEEE Transactions on Information Technology in Biomedicine*, Vol. 1, pp. 44-54, 1997.
21. Chew, K. L. (1991), *Cyclic Schedule for Apron Services*, *Journal of the Operational Research Society*. Vol. 42, No. 12, pp. 1061-1069.
22. Chu Hai D. , Eric Gelman and Ellis L. Johnson , “Solving large scale crew scheduling problems”, *European Journal of Operational Research*, Vol 97, pp260-268, 1997.
23. Chun, H.W., H.C. Chan, P.S. Lam, M.F. Tsang, J. Wang and W.M. Yeung, “Nurse

- Rostering at the Hospital Authority of Hong Kong,” In Proceedings of AAAI/IAAI Conference, 2000.
24. Dantzig, G.B., “A Comment on Edie’s Traffic Deals at Toll Booths,” Operations Research, Vol.3, pp. 339-341, 1954.
  25. Higgins, A., Scheduling of Railway Track Maintenance Activities and Crews, Journal of the Operational Research Society, Vol. 49, pp. 1026-1033, 1998.
  26. Hillier, F. S., Lieberman, G. J. “Introduction to Operation Research, McGraw-Hill.
  27. Hoffman Karla L. and Padberg Manfred, ”Solving Airline Crew Scheduling Problem by Branch-and-Cut”, Management Science, Vol.39, NO.6, pp657-682, 1993.
  28. Kim, K. H. and Kim, K. W., Operator-Scheduling Using A Constraint Satisfaction Technique in Port Container Terminals, Computer & Industrial Engineering Vol. 46, pp. 373-381, 2004.
  29. Lau, H. C., “On the Complexity of Manpower Shift Scheduling,” Computers Operations Research, Vol.23, No.1, pp. 93-102, 1996.
  30. Lim Andrew, Brian Rodrigues, Zhou Xu, (2004), Solving the Crane Scheduling Problem Using Intelligent Search Schemes, Lecture Notes in Computer Science, Volume 3258
  31. Smith, L.D. and Wiggins A., “A Computer-Based Nurse Scheduling System,” Computer Operation Research, Vol.4, pp.195-212, 1977.
  32. Young-Man Park, Kap Hwan Kim, A scheduling method for Berth and Quay cranes, OR Spectrum, Volume 25, Number 1, 2003.

## 附錄一、C語言程式

```
#include <stdlib.h>

#include <stdio.h>

#define DAY 7      /*全部天數為 7 天*/

#define SLOT 5    /*SLOT 有 5 個班次*/

#define GROUP 50  /*可供排班有 50 位人員*/

#define FDEMAND1 "D:\\論文程式\\TEST 包\\最終版\\demand1.txt"
#define FDEMAND2 "D:\\論文程式\\TEST 包\\最終版\\demand2.txt"
#define FDEMAND3 "D:\\論文程式\\TEST 包\\最終版\\demand3.txt"
#define FDEMAND4 "D:\\論文程式\\TEST 包\\最終版\\demand4.txt"
#define FDEMAND5 "D:\\論文程式\\TEST 包\\最終版\\demand5.txt"
#define FDEMAND6 "D:\\論文程式\\TEST 包\\最終版\\demand6.txt"
#define FDEMAND7 "D:\\論文程式\\TEST 包\\最終版\\demand7.txt"
#define DAYNUM "D:\\論文程式\\TEST 包\\最終版\\day.txt"
#define DAY_SHIP "D:\\論文程式\\TEST 包\\最終版\\ship.txt"
#define SHIFT_DEMAND "D:\\論文程式\\TEST 包\\最終版\\shift_demand.txt"
#define outmodel "D:\\論文程式\\TEST 包\\最終版\\model.txt"

int *day;

int *ship;

int **demand1=NULL;

int **demand2=NULL;

int **demand3=NULL;
```

```

int **demand4=NULL;

int **demand5=NULL;

int **demand6=NULL;

int **demand7=NULL;

int **off=NULL;

int shiftkind[SHIFT]={1,2,3,4,5};

int i,j,k,l;

int d;

void read_all_demand(void);           /*讀入所有的資料*/

void write_model(void);              /*印出所有的 model*/

void read_day(char *);               /*給予天數代號，如第一天為 day[0]等*/

void read_ship(char *);             /*ship 函數*/

void requiredemand_space(int ***,int); /*各天需求的空間*/

void read_demand(char *,int,int **); /*讀取各天的需求*/

void read_rest(int ***);

int objective(FILE *model);         /*宣告目標式函數*/

int a(void);

void min_group(FILE *model);

void work_need(int,int,int **,FILE *model); /*宣告工作時段需求限制式*/

void time_conflict(FILE *model);     /*時段衝突限制式*/

void holiday(FILE *model);          /*輪休限制式*/

void continue_work(FILE *model,int,int); /*連續執勤限制式*/

```



```

void day_no_work(FILE *model);           /*隔日不立刻勤執限制式*/
int c(void);
void end(FILE *model);

```

```
int main(void)
```

```

{
    read_all_demand();
    write_model();
    return 0;
}

```

```
void read_all_demand(void)
```

```

{
    read_day(DAYNUM);
    read_ship(DAY_SHIP);
    requiredemand_space(&demand1,ship[0]); /*第一天的需求空間*/
    requiredemand_space(&demand2,ship[1]); /*第二天的需求空間*/
    requiredemand_space(&demand3,ship[2]); /*第三天的需求空間*/
    requiredemand_space(&demand4,ship[3]); /*第四天的需求空間*/
    requiredemand_space(&demand5,ship[4]); /*第五天的需求空間*/
    requiredemand_space(&demand6,ship[5]); /*第六天的需求空間*/
    requiredemand_space(&demand7,ship[6]); /*第七天的需求空間*/
    read_demand(FDEMAND1,ship[0],demand1); /*讀取第一天的需求*/
    read_demand(FDEMAND2,ship[1],demand2); /*讀取第二天的需求*/
    read_demand(FDEMAND3,ship[2],demand3); /*讀取第三天的需求*/
}

```





```

    read_demand(FDEMAND4,ship[3],demand4); /*讀取第四天的需求*/
    read_demand(FDEMAND5,ship[4],demand5); /*讀取第五天的需求*/
    read_demand(FDEMAND6,ship[5],demand6); /*讀取第六天的需求*/
    read_demand(FDEMAND7,ship[6],demand7); /*讀取第七天的需求*/

    read_rest(&off);

    return;
}

```

```

void read_day(char *fileday) /*以下為各天代號的讀取*/

```

```

{
    FILE *fday=NULL;
    fday=fopen(fileday,"r");
    day=(int *)malloc(DAY*sizeof(int));

    for(i=0;i<DAY;i++)
    {
        fscanf(fday,"%d",&day[i]);
    }
    fclose(fday);
    return;
}

```



```

void read_ship(char *fileship) /*以下為各天船舶艘數的讀取*/

```

```

{
    FILE *fship=NULL;

```

```

fship=fopen(fileship,"r");
ship=(int *)malloc(DAY*sizeof(int));

for(j=0;j<DAY;j++)
{
    fscanf(fship,"%d",&ship[j]);
}

fclose(fship);
return;
}

```

```

void requiredemand_space(int ***pta_demand,int shipnum) /*以下為各天需求的空間*/

```

```

{
    int **a_demand=NULL;

    a_demand=(int **)malloc(shipnum*sizeof(int *));
    for(j=0;j<shipnum;j++)
        a_demand[j]=(int *)malloc(SHIFT*sizeof(int));
    *pta_demand=a_demand;
    return;
}

```

```

void read_demand(char *filename,int shipnum,int **demand) /*以下為讀取各天的需求*/

```

```

{
    FILE *fd=NULL;

```

```

fd=fopen(filename,"r");

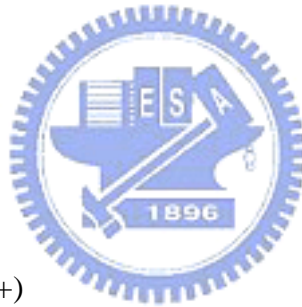
for(j=0;j<shipnum;j++)
    for(k=0;k<SHIFT;k++)
        {
            fscanf(fd,"%d",&demand[j][k]);
        }
fclose(fd);
return;
}

```

```

void read_rest(int ***poff)
{
    for(i=0;i<DAY;i++)
        for(l=0;l<GROUP;l++)
            {
                fscanf(fo,"%d",&off[i][l]);
            }
    fclose(fo);
    return;
}

```



```

void write_model(void)
{
    FILE *model=NULL;

```

```

model=fopen(outmodel,"w");

fprintf(model,"MIN\n");

objective(model);          /*目標式*/

fprintf(model,"\nST.\n");

min_group(model);

work_need(day[0],ship[0],demand1,model);    /*第一天工作時段限制式*/
work_need(day[1],ship[1],demand2,model);    /*第二天工作時段限制式*/
work_need(day[2],ship[2],demand3,model);    /*第三天工作時段限制式*/
work_need(day[3],ship[3],demand4,model);    /*第四天工作時段限制式*/
work_need(day[4],ship[4],demand5,model);    /*第五天工作時段限制式*/
work_need(day[5],ship[5],demand6,model);    /*第六天工作時段限制式*/
work_need(day[6],ship[6],demand7,model);    /*第七天工作時段限制式*/

time_conflict(model);    /*時段衝突限制式*/

holiday(model);          /*輪休限制式*/

continue_work(model,shiftkind[0],shiftkind[1]);    /*連續勤執限制式 1*/
continue_work(model,shiftkind[1],shiftkind[2]);    /*連續勤執限制式 2*/
continue_work(model,shiftkind[2],shiftkind[3]);    /*連續勤執限制式 3*/
continue_work(model,shiftkind[3],shiftkind[4]);    /*連續勤執限制式 4*/

day_no_work(model);          /*隔日不立刻勤執限制式*/

end(model);

fclose(model);

return;

```

```
}
```

```
int objective(FILE *model)    /*以下為目標式*/
```

```
{
```

```
    fprintf(model,"Z");
```

```
    return 0;
```

```
}
```

```
int a(void)
```

```
{
```

```
    for(j=0;j<=DAY;j++)
```

```
    {
```

```
        if(i-j==1)
```

```
            return ship[j];
```

```
    }
```

```
    return 0;
```

```
}
```



```
void min_group(FILE *model)
```

```
{
```

```
    int b;
```

```
    for(l=1;l<=GROUP;l++)
```

```
    {
```

```
        for(i=1;i<=DAY;i++)
```

```

    {
        b=a();
        for(j=1;j<=b;j++)
            for(k=1;k<=SHIFT;k++)
                {
                    fprintf(model,"+X%d_%d_%d_%d",i,j,k,l);
                }
    }
    fprintf(model,"-Z<=0\n");
}

return;
}

```



```

void work_need(int day,int shipnum,int **demand,FILE *model) /*以下為工作時段限制

```

```

式*/

```

```

{
    for(j=0;j<shipnum;j++)
        for(k=0;k<SHIFT;k++)
            {
                for(l=1;l<=GROUP;l++)
                    {
                        fprintf(model,"+X%d_%d_%d_%d",day,j+1,k+1,l);
                    }
                fprintf(model,"=%d\n",demand[j][k]);
            }
}

```

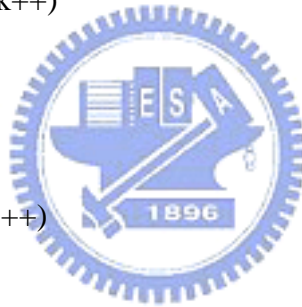
```

    }
return;
}

void time_conflict(FILE *model) /*以下為時段衝突限制式*/
{
    int b;

    for(l=1;l<=GROUP;l++)
        for(i=1;i<=DAY;i++)
            for(k=1;k<=SHIFT;k++)
                {
                    b=a();
                    for(j=1;j<=b+1;j++)
                        {
                            if(j<b+1)
                                {
                                    fprintf(model,"+X%d_%d_%d_%d",i,j,k,l);
                                }
                            if(j==b+1 && b!=0)
                                fprintf(model,"<=1\n");
                        }
                }
    return ;
}

```



```
void holiday(FILE *model)    /*以下為輪休限制式*/
```

```
{
```

```
    int b;
```

```
    for(i=1;i<=DAY;i++)
```

```
    {
```

```
        b=a();
```

```
        for(j=1;j<=b;j++)
```

```
            for(k=1;k<=SHIFT;k++)
```

```
                for(l=1;l<=GROUP;l++)
```

```
void continue_work(FILE *model,int o,int p) /*以下為不可連續執勤時段限制式*/
```

```
{
```

```
    int b;
```

```
    for(l=1;l<=GROUP;l++)
```

```
        for(i=1;i<=DAY;i++)
```

```
void day_no_work(FILE *model)    /*以下為隔日不立刻勤執限制式*/
```

```
{
```

```
    int b,e;
```

```
    k=SHIFT;
```



```

for(l=1;l<=GROUP;l++)
{
    for(i=1;i<=DAY-1;i++)
    }
}
return;
}

```

```

int c(void)
{
    for(j=0;j<=DAY;j++)
    {
        if(d-j==1)
            return ship[j];
    }
    return 0;
}

```



```

void end(FILE *model)
{
    int b;
    fprintf(model,"BIN\n");
    }
    fprintf(model,"\nend\n");
    return;
}

```