

國立交通大學
運輸科技與管理學系
碩 士 論 文

結合限制規劃與數學規劃求解大型後艙空勤組員排班問題

Applications of Constraint Programming and Mathematical Programming
Techniques to Solve Large-Scale Cabin Crew Scheduling Problems

研 究 生：王國琛

指導教授：韓復華

中華民國九十一年六月

結合限制規劃與數學規劃求解大型後艙空勤組員排班問題

Applications of Constraint Programming and Mathematical Programming
Techniques to Solve Large-Scale Cabin Crew Scheduling Problems

研 究 生：王國琛

Student：Kuo-Chen Wang

指導教授：韓復華

Advisor：Anthony Fu-Wha Han

國立交通大學
運輸科技與管理學系
碩士論文

A Thesis

Submitted to Institute of Transportation Technology & Management

College of Management

National Chiao Tung University

In Partial Fulfillment of the Requirements

for the Degree of

Master of Engineering

in Traffic and Transportation

June 2002

Hsinchu, Taiwan, Republic of China

中華民國 九十一年 六 月

結合限制規劃與數學規劃求解大型後艙空勤組員排班問題

研究生：王國琛

指導教授：韓復華

國立交通大學運輸科技與管理學系碩士班

摘要

空勤組員排班屬於 NP-Hard 之大型組合最佳化問題，也是航空公司營運規劃的重要課題，由於其實務問題之規模龐大計算複雜度高，故在作業研究領域中有諸多文獻探討。本研究將應用新近發展之限制規劃方法與傳統的數學規劃方法，求解大型後艙空勤組員排班問題。

CP 源自於人工智慧領域，主要是用來求解限制滿足問題(CSP)，此方法論之最大優點在於高度的模式彈性與快速的求解績效。一般而言，組員排班可分為「勤務組合產生」與「排班成本最小化」兩個部分。本研究將勤務組合產生問題視為一個 CSP 問題，並應用 CP 建立一個有效率的勤務組合產生模式(簡稱 CSP 模式)，該模式能同時考慮各種排班法規(飛時、工時、休時)與營運因素(基地、機型接續、艙等、空載)，故較傳統之勤務組合產生方法更具彈性。以 CSP 模式為基礎，將航段依班次頻率分類，再加上專家知識之限制式，構建一個 CSP-CP 模式來產生具有實用價值之受限性勤務組合集合(簡稱 CP 集合)，期能有效地縮小最小成本問題之求解規模。

考慮實務問題規模通常屬於大型問題，本研究結合 CSP-CP 模式與 MP 模式建立一套以限制規劃為基礎之變數產生法。在績效比較方面，本研究採用兩篇文獻所使用之測試問題作為比較基準，其問題之實際規模分別為 1.7×10^6 與 6×10^7 。測試結果發現，此求解方法在兩個大型測試例題上所求得之最小成本皆優於文獻結果，其改善幅度分別為 3% 與 2.7%，運算時間分別為 23.9 秒與 203.2 秒。此結果顯示，CSP-CP 模式確實能有效地結合 MP 來求解大型後艙組員排班問題。

基於 CSP-CP 模式能有效地產生具實用性之受限性勤務組合，本研究亦根據 CSP-CP 模式列舉所有不含空載之基本勤務組合(Basic Pairing, BP)集合與恰含 1 個空載之延伸性勤務組合(Extended Pairing, EP)集合，直接作最小成本之優化。結果發現，此種限制列舉式之產生-優化法(Generate-and-Optimize)，對兩個大型測試例題亦可不經由變數產生法的分解機制，直接求得相同最小成本的結果，由此可知，CSP-CP 模式確實能有效地縮減問題規模並保持解之品質。

關鍵字：空勤組員排班、勤務組合產生、限制規劃、限制滿足問題、變數產生法、限制列舉

Applications of Constraint Programming and Mathematical Programming Techniques to Solve Large-Scale Cabin Crew Scheduling Problems

Student : Kuo-Chen Wang

Advisor : Anthony Fu-Wha Han

Institute of Transportation Technology and Management
National Chiao Tung University

ABSTRACT

Because crew cost is only second to fuel in airline operations, the crew scheduling problem has been widely studied in the literature. Most existing literature on this problem is based on mathematical programming (MP) models. This research takes a new approach using constraint programming (CP) to complement MP in solving large-scale crew scheduling problems.

Essentially, a crew scheduling problem can be decomposed into a pairing generation problem, and a minimum-cost set partitioning problem (SPP). In this research, we took the pairing generation problem as a constraint satisfaction problem (CSP) and developed a CSP model to generate feasible pairings. Since the number of feasible pairings tends to be enormous, we developed a CSP-CP model to generate feasible constrained pairings (CP) by adding some expert-knowledge rules to CSP model.

Considering most crew scheduling problems in practice are large-scale, we combine the CSP-CP model and SPP model to develop a CP-based column generation method. To evaluate the performance of this method, we adopted two large-scale problems in literature as our test problems, each has a problem size of 1.7×10^6 and 6×10^7 respectively. Computational results showed that CP-based column generation yields an improvement of 3% and 2.7% over the published best-known solutions of the two test problems respectively. Such results imply that the CSP-CP model can be used to complement the MP in column generation effectively.

Because the CSP-CP model can effectively generate feasible pairings, we also applied the CSP-CP model to solve the crew scheduling problem in the “generate-and-optimize” framework. The generated CP includes a set of basic pairings (BP), each of which contains no deadheads, and a set of extended pairings, each of which contains exactly one deadhead. We found that the CP-based generate-and-optimize method yielded the same optimal costs as we obtained in the CP-based column generation method. Such results imply that the CSP-CP model may effectively reduce the problem size without losing solution quality in solving large-scale cabin crew scheduling problems.

Keywords : Crew Scheduling, Pairing Generation, Constraint Programming, Constraint Satisfaction Problem, Column Generation, Constrained Enumeration.

誌 謝

本碩士論文得以順利完成，首先要感謝恩師 韓復華教授兩年來之悉心指導與教誨，無論在研究方法論之啟迪、研究方向之指引以及嚴謹之治學態度，均讓學生受益匪淺，師恩浩蕩、永銘在心！

論文口試期間，承蒙 中華大學 張靖教授、交通大學 汪進財教授以及系上 王晉元教授不吝指教與斧正，所提供的種種寶貴意見使得本論文能夠更臻完備充實。論文審查時系上 卓訓榮教授與交大資管所 陳安斌教授所給予之建議，均使本論文更加嚴謹充實。此外，論文寫作期間，中華大學 卓裕仁助理教授、樹德科技大學 杜宇平助理教授所給予之指導與建議，均使本研究更加嚴謹充實。在此由衷地表示謝忱！

充實的碩士生活即將在此劃下句點，其中的箇中實在酸甜難以言喻，回顧在學期間之點點滴滴，由衷地感謝所上諸位師長們的循循善誘、學長姐的提攜與照顧、同儕的切磋與砥礪、學弟妹的同甘共苦、朋友們的精神鼓勵。

最後，我要將此論文獻給我最摯愛的雙親、姐姐與哥哥，感謝你們在我求學過程中持續不斷的給予支持與鼓勵，使我能克服種種困難並完成碩士學位。未來我將持續地向前邁進，希望能不辜負你們對我的期望。

能完成此篇論文，要感謝的人實在很多，在此僅將這份榮耀與你們一起分享。

王國琛 謹誌

記于 2002.6 新竹交大

目錄

| | |
|------------------------------|------|
| 中文摘要 | |
| 英文摘要 | |
| 誌謝 | |
| 目錄 | IV |
| 圖目錄 | VII |
| 表目錄 | VIII |
| 第一章 緒論 | 1 |
| 1.1 研究背景與動機 | 1 |
| 1.2 研究目的與範圍 | 3 |
| 1.3 研究方法與流程 | 3 |
| 第二章 文獻回顧 | 6 |
| 2.1 組員排班問題 | 6 |
| 2.1.1 勤務組合產生問題與求解方式 | 6 |
| 2.1.2 最小成本勤務組合產生問題 | 9 |
| 2.1.3 勤務組合產生問題之相關成計算 | 10 |
| 2.2 最小成本勤務組合產生問題特性之探討 | 12 |
| 2.3 最小成本勤務組合產生問題之求解演算法 | 15 |
| 2.3.1 啟發式解法 | 15 |
| 2.3.2 最佳解解法 | 16 |
| 2.4 變數產生法 | 18 |
| 2.4.1 以網路模式為基礎之變數產生法 | 19 |
| 2.4.2 以限制規劃為基礎之變數產生法 | 22 |
| 第三章 限制規劃文獻回顧 | 24 |
| 3.1 限制規劃簡介 | 24 |
| 3.2 限制滿足問題簡介 | 27 |
| 3.2.1 限制滿足問題之定義 | 27 |
| 3.2.2 限制滿足問題中之限制式種類 | 28 |
| 3.3 限制規劃中之求解演算法簡介 | 29 |
| 3.3.1 一致性檢驗技術介紹 | 29 |
| 3.3.2 空間搜尋演算法 | 32 |
| 3.3.2 空間搜尋演算法中之搜尋策略 | 37 |

| | |
|---------------------------------|----|
| 3.4 限制規劃之整體求解機制 | 40 |
| 3.5 構建限制規劃模式所需考慮之因素 | 42 |
| 3.6 模式化語言 OPL 之簡介 | 46 |
| 第四章 一般化勤務組合產生問題限制規劃模式 | 47 |
| 4.1 勤務組合產生問題 | 47 |
| 4.1.1 問題描述 | 47 |
| 4.1.2 勤務型態 | 50 |
| 4.1.3 勤務組合型態 | 50 |
| 4.2 勤務組合產生問題所考慮之勤務組合集合 | 53 |
| 4.3 勤務組合產生問題之相關法規限制 | 54 |
| 4.4 勤務組合產生問題之成本計算方式 | 57 |
| 4.5 一般化勤務組合產生問題限制規劃模式 | 58 |
| 4.5.1 模式輸入資料 | 59 |
| 4.5.2 模式參數與變數 | 59 |
| 4.5.3 模式限制式 | 64 |
| 4.6 勤務組合集合產生架構 | 72 |
| 4.6.1 TP 集合產生架構 | 72 |
| 4.6.2 CP 集合產生架構 | 73 |
| 4.7 一般化勤務組合產生問題限制規劃模式之類型 | 77 |
| 4.7.1 窮舉式的勤務組合產生模式 | 77 |
| 4.7.2 限制列舉式的勤務組合產生模式 | 77 |
| 第五章 後艙空勤組員排班模式與求解架構 | 78 |
| 5.1 八種後艙空勤組員排班模式 | 78 |
| 5.2 排班模式求解架構 | 81 |
| 5.2.1 以限制規劃為基礎之變數產生法 | 81 |
| 5.2.1.1 起始解建構模組 | 83 |
| 5.2.1.2 子問題求解模組 | 83 |
| 5.2.2 限制列舉式之產生-優化法 | 85 |
| 第六章 實證分析 | 87 |
| 6.1 測試例題簡介 | 87 |
| 6.2 排班模式之測試-單基地、單機型接續、單艙等 | 87 |
| 6.2.1 排班資料 | 87 |
| 6.2.2 問題規模與最佳解 | 90 |
| 6.2.2.1 問題規模-TP 集合 | 90 |
| 6.2.2.2 問題規模-CP 集合 | 91 |
| 6.2.2.3 問題之最佳解 | 91 |

| | |
|---------------------------------|-----|
| 6.2.3 測試結果與分析 | 92 |
| 6.3 排班模式之測試-多基地、多機型接續、多艙等 | 95 |
| 6.3.1 排班資料 | 95 |
| 6.3.2 問題規模與最佳解 | 98 |
| 6.3.2.1 問題規模-TP 集合 | 98 |
| 6.3.2.2 問題規模-CP 集合 | 99 |
| 6.3.2.3 問題之最佳解 | 99 |
| 6.3.3 測試結果與分析 | 100 |
| 第七章 結論與建議 | 105 |
| 7.1 結論 | 105 |
| 7.2 建議 | 107 |
| 參考文獻 | 108 |
| 附錄一 | 111 |
| 附錄二 | 114 |

圖目錄

| | |
|-----------------------------------|-----|
| 圖 1.1 研究流程圖 | 5 |
| 圖 3.1 GT 之運作方式 | 32 |
| 圖 3.2 BT 搜尋法–以 4-queens 為例 | 34 |
| 圖 3.3 BT、FC 與 LC 三種演算法之差異圖 | 35 |
| 圖 3.4 FC 搜尋法–以 4-queens 為例 | 35 |
| 圖 3.5 MAC 搜尋法–以 4-queens 為例 | 36 |
| 圖 3.6 限制規劃之整體求解機制 | 41 |
| 圖 4.1 可行勤務組合之相關名詞介紹示意圖 | 49 |
| 圖 4.2 $N_d = 4$ 時之勤務類型 | 50 |
| 圖 4.3 Pair 型態中最基本之勤務組合樣式 | 52 |
| 圖 4.4 Pair 型態中最基本之可行勤務組合樣式 | 52 |
| 圖 4.5 CSP 模式之求解概念圖 | 58 |
| 圖 4.6 TP 集合產生架構 | 72 |
| 圖 4.7 TP 集合所包含之子集合 | 73 |
| 圖 4.8 CP 集合產生架構 | 75 |
| 圖 4.9 CP 集合所包含之子集合 | 76 |
| 圖 5.1 以限制規劃為基礎之變數產生法求解架構 | 82 |
| 圖 5.2 限制列舉式之產生-優化法求解架構 | 85 |
| 圖 6.1 TPE-ANC-NYC 航線之排班網路 | 103 |

表目錄

| | |
|--|----|
| 表 2.1 運用網路模式產生可行勤務組合之缺點..... | 8 |
| 表 2.2 北美與歐洲區域航空公司排班問題之相異特性..... | 14 |
| 表 2.3 分枝定限法之四項求解步驟..... | 16 |
| 表 2.4 以網路模式為基礎之變數產生法文獻..... | 21 |
| 表 2.5 以限制規劃模式為基礎之變數產生法文獻..... | 23 |
| 表 3.1 OR 與 CP 方法所適用之問題類型..... | 26 |
| 表 4.1 勤務組合產生問題之相關名詞定義..... | 47 |
| 表 4.2 $N_p=6$ 、 $N_d=4$ 時之勤務組合類型與勤務接續類型..... | 51 |
| 表 4.3 排班問題規模之計算公式..... | 54 |
| 表 4.4 排班法規分類-依法規所屬之組織來區分..... | 55 |
| 表 4.5 排班法規分類-依法規所適用之對象來區分..... | 55 |
| 表 4.6 勤務組合產生問題限制規劃模式之輸入資料..... | 59 |
| 表 4.7 CSP 模式之航段資料參數與變數..... | 59 |
| 表 4.8 CSP 模式之成本資料參數..... | 60 |
| 表 4.9 CSP 模式之排班法規參數..... | 60 |
| 表 4.10 CSP 模式之排班因素參數..... | 61 |
| 表 4.11 CSP 模式之變數..... | 61 |
| 表 4.12 航線班次密集度分類準則..... | 74 |
| 表 4.13 窮舉式的勤務組合產生模式..... | 77 |
| 表 4.14 限制列舉式的勤務組合產生模式..... | 77 |
| 表 5.1 八種後艙空勤組員排班模式..... | 78 |
| 表 6.1 測試例題資料..... | 87 |
| 表 6.2 例題資料-例題 1..... | 87 |
| 表 6.3 勤務段產生規則-例題 1..... | 88 |
| 表 6.4 勤務組合產生規則-例題 1..... | 88 |
| 表 6.5 成本資料-例題 1..... | 88 |
| 表 6.6 問題規模-TP 集合-例題 1..... | 90 |
| 表 6.7 問題規模-CP 集合-例題 1..... | 91 |
| 表 6.8 測試結果-例題 1..... | 92 |
| 表 6.9 兩種求解架構間之求解結果與執行績效分析-例題 1..... | 93 |
| 表 6.10 本研究最佳解與文獻最佳解之比較分析-例題 1..... | 94 |
| 表 6.11 例題資料-例題 2..... | 95 |
| 表 6.12 勤務段產生規則-例題 2..... | 95 |
| 表 6.13 勤務組合產生規則-例題 2..... | 96 |

| | |
|--|-----|
| 表 6.14 成本資料-例題 2 | 96 |
| 表 6.15 航線資料-例題 2 | 96 |
| 表 6.16 機型資料-例題 2 | 97 |
| 表 6.17 各機型需各艙等組員之人數資料-例題 2 | 97 |
| 表 6.18 各基地、各艙等組員之平均飛時薪資成本資料-例題 2 | 97 |
| 表 6.19 問題規模-TP 集合(單艙等)-例題 2..... | 98 |
| 表 6.20 問題規模-CP 集合(單艙等)-例題 2 | 99 |
| 表 6.21 測試結果-例題 2 | 100 |
| 表 6.22 兩種求解架構間之求解結果與執行績效分析-例題 2 | 101 |
| 表 6.23 本研究最佳解與文獻最佳解之比較分析-例題 2 | 102 |
| 表 6.24 文獻與本研究所考慮之勤務類型-例題 2 | 104 |

第一章 緒論

1.1 研究背景與動機

在我國即將加入世界貿易組織(WTO)之際，開放天空政策壓力勢必將隨之而來，屆時國籍航空公司與外籍航空公司將處在較公平的市場環境下競爭，若國籍航空公司要維持其在航空市場的競爭優勢，除需提高其相關的服務水準與飛航安全外，亦需積極降低其營業成本。一般而言，航空公司的第一大營運成本分別為燃油成本，第二大營運成本為組員薪資成本(Crew Cost)，約佔總營運成本之 10-15% [31]。由於飛機燃油成本不易降低更動，所以航空公司僅能轉向有效運用組員資源以期降低營業成本、增加營業效率。以中華航空公司為例，其 2000 年營運支出成本約為 6 百七十億台幣，若以 10-15% 成本來計算，其組員薪資成本約在 67 億至 100 億五千萬之間，金額十分龐大。然而對於國外之大型航空公司而言，其組員薪資成本更高，Anbil et al. [2] 指出美國航空公司 (American Airlines, AA) 在 1991 年，其組員薪資成本高達 13 億美金之多，Graves et al. [17] 指出美國聯合航空公司 (United Airlines, UA) 在 1993 年，單駕駛員的薪資成本就高達 6 億美金之多。因此，國籍航空公司若能有效率地安排空勤組員之行程班表，以降低空勤組員在外站停留時間、減少外宿費用及零用金，便能降低營運成本、增加市場競爭力。

空勤組員行程班表排定問題為航空公司營運中必須經常處理之一項組合最佳化問題，然而，由於此問題之複雜度以及其對航空公司營運成本、組員士氣之重要性，所以在過去三、四十年間，已有相當多之國內外學者投入此問題之相關研究，其中，較著名之文獻如：Arabeyre [1] 等人在 1969 年開始針對此問題進行研究分析、Bodin [6] 等人在 1983 年開始嘗試將空勤組員行程班表排定問題分為「組員排班問題 (Crew Scheduling Problem)」與「組員派遣問題 (Crew Rostering Problem)」兩個階段來求解，由於 Bodin 等學者之求解方式，能大幅地降低空勤組員行程班表排定問題之複雜度，所以後來之學者皆將空勤組員行程班表排定問題分為上述兩階段來處理，因此，空勤組員行程班表排定問題又名為空勤組員排班與派遣問題 (Airline Crew Scheduling and Rostering Problem)。其中，由於組員排班問題之主要目的乃在於「求解一組成本最小且涵蓋所有航段之可行勤務組合集合」，於是文獻又稱此問題為最小成本勤務組合產生問題 (Min-Cost Crew Pairing Generation Problem)，此外，由於此一階段所求得之勤務組合與組員完全無關，所以此一階段所產生之勤務組合又稱為匿名可行勤務組合 (Anonymous Pairing)；組員派遣問題之主要目的乃在於「如何將組員排班問題所產生的那組成本最小且涵蓋所有航段之匿名可行勤務組合，公平地指派給每個組員並形成每個組員之行程班表」。

在排班問題所考慮之勤務組合集合 (Set of Pairings) 方面，文獻大多沒有明確地定義其所考慮之勤務組合集合範圍，而是從所有可行勤務組合所形成之集合中，直接產生品質較優良之勤務組合，有鑑於當問題規模變大時，所有可行勤務組合所形成之集合將變得相當龐大，所以，本研究將嘗試根據國際航線後艙空勤組員排班問題之特性，探討如

何定義出一個適合國際航線後艙組員排班問題之優良勤務組合集合，並運用限制規劃模式發展一套 CP 集合產生架構來產生此組優良勤務組合集合，以期能大幅度地縮小排班問題所考慮之勤務組合集合，並能在不失解品質之原則下加快求解排班問題之速度。

在勤務組合產生(Pairing Generation)方面，儘管過去三、四十年以來，國內外文獻紛紛不斷地對組員排班問題之求解方法進行廣泛地研究與探討，但直至目前為止，文獻中所提及之勤務組合產生方式大都是以檢驗-回溯法(Test-and-Backtrack Method)或資源受限制之之最短路徑法(Constrained Shortest Path Algorithm)為主，然而，有鑑於勤務組合產生問題所需考量的排班法規甚為繁多(如民航法規、工會規定、航空公司空勤組員派遣規則等)，而且上述兩種勤務組合產生模式具有模式彈性低、勤務組合產生效率低、法規限制透明度低、法規限制管理不易等缺點，所以，本研究將探討如何運用限制規劃方法(Constraint Programming, CP)來構建一個模式彈性高、勤務組合產生效率佳而且考慮各種排班法規與排班因素之一般化勤務組合產生問題限制規劃模式，以期提供未來相關研究一種新的勤務組合產生方法。

在勤務組合最佳化(Pairing Optimization)方面，文獻上最常使用之兩種方法為產生-優化法(Generate-and-Optimize)與變數產生法(Column Generation)。在產生-優化法之使用上，由於其必須先窮舉所有或部分品質較佳之勤務組合變數然後再求得最佳解，所以在以往之文獻中，大多只有在求解小規模之組員排班問題時，才使用產生-優化法來，然而，隨著近幾年來電腦運算速度與電腦記憶體容量之大幅提昇，以及 IP 求解演算法效率之大幅提昇，使得近年來文獻中所用產生-優化法之比率漸漸的提升；在變數產生法之使用上，由於此方法能在不窮舉所有勤務組合變數之情況下，仍求得一個近似最佳解，所以文獻中大多使用變數產生法來求解大規模的組員排班問題，在其解法架構中，大多以 Minoux[26]所使用之變數產生法求解架構為主，然而，由於此一變數產生法之架構是以網路模式為其勤務組合產生方法，所以這類的變數產生法又被稱為以網路模式為基礎之變數產生法求解架構。有鑑於上述兩種求解方法之優越性，所以本研究將以上述求解架構為勤務組合最佳化方法，以一般化勤務組合產生問題限制規劃模式為勤務組合產生方法，分別提出「限制列舉式之產生-求解法(Constrained Enumeration Based Generate-and-Optimize)」與「以限制規劃為基礎之變數產生法(Constraint Programming Based Column Generation)」兩種求解架構。

本研究希望藉由「CP 集合產生架構」與「一般化勤務組合產生問題限制規劃模式」之建立，來發展「限制列舉式之產生-求解法」與「以限制規劃為基礎之變數產生法」兩種求解架構，並透過此兩種求解架構對於實務問題之求解績效，實際探討「運用限制規劃方法於求解勤務組合產生問題之可行性」與「結合限制規劃與數學規劃方法於求解大規模後艙組員排班問題之可行性」，以期提供未來後續相關研究與國內航空公司建立其智慧型組員排班系統時之參考，最終之目的乃在期望能提高國內航空公司之營運效率與競爭力，以加強國內航空產業之競爭優勢。

1.2 研究目的與範圍

求解空勤組員排班問題之主要目的在於改善航空公司的營運績效、提高服務品質、減少營運成本以及增進其空勤組員資源之運用效率，然而由於實務問題的規模太過於龐大並且必須考慮種種複雜的法規限制，使得空勤組員排班問題的複雜度相當高，所以本研究之主要目的在於「探討如何有效地結合限制規劃與數學規劃方法於求解空勤組員排班問題，並建立一套符合國內航空公司現實營運需求且求解效率佳之一般化後艙組員排班架構，提供國內各大航空公司在未來開發空勤組員排班系統時之考慮」。

本研究乃以「國際航線之後艙組員排班問題(Cabin Crew Scheduling Problem)」為研究範圍，以「國內某家國際航空公司之國際航線資料」為測試例題，以最簡單之「單基地、單機型接續、單一艙等」排班模式與最複雜之「多基地、多機型接續、多艙等」排班模式作為測試本研究求解架構執行績效之測試模式。

由於本研究主要是規劃中長期後艙組員之排班問題，所以對於班表擾動所造成之空勤組員調度(Tracking)等相關問題，不在本研究範圍內。

1.3 研究方法與流程

本研究係在已知的草擬班次表、機隊排程、排班法規(如：民航法規、工會規定、航空公司空勤組員派遣規則等)條件下，將勤務組合產生問題定式成一般化勤務組合產生問題限制規劃模式，將最小成本勤務組合產生問題定式成集合涵蓋數學規劃模式，以最小化空勤組員營運成本為目標，以限制規劃方法與數學規劃方法為方法論之核心，發展「以限制規劃為基礎之變數產生法」與「限制列舉法」二種以限制規劃為基礎之求解架構，來求解大規模之後艙組員排班問題。

在模式建構方面，以 OPL 語法建構限制規劃與數學規劃模式，以 OPL Script 語法建構求解演算法；在模式求解方面，以 ILOG Solver 5.0 為求解限制規劃模式之函示庫、以 ILOG CPLEX 7.0 為求解數學規劃模式之函示庫；在模式測試平台方面，乃以 ILOG OPL Studio 3.1 為測試平台；在測試環境方面，以 AMD K7-800、512 MB、win2000 之 PC 為測試環境。

本研究之研究內容與流程，如下列所述：

(1) 空勤組員排班問題與限制規劃方法相關文獻之蒐集與回顧

首先需蒐集空勤組員排班之相關文獻，以進一步了解空勤組員排班問題之特性、排班限制種類與解題方法，並依據其問題特性與求解方法加以整理。接著，蒐集限制規劃方法之相關文獻，深入瞭解限制規劃方法對於限制規劃模式之求解方法與機制，以期建立較有效率之勤務組合產生問題限制規劃模式。

(2) 收集關於結合限制規劃方法與數學規劃方法於求解最小成本勤務組合產生問題之相

關文獻

針對運用限制規劃與數學規劃方法於求解最小成本勤務組合產生問題之相關文獻進行整理與探討，深入了解目前文獻中運用限制規劃方法在空勤組員排班問題上之發展現況，並進而探討如何才能有效地結合限制規劃與數學規劃方法來求解後艙組員之排班問題。

(3) 收集與整理測試例題之排班資料

蒐集並整理測試例題之排班資料(包括：班次表、住宿成本、空載成本、薪資計算方式、工會規定、法規限制、空勤組員派遣作業規程等)。

(4) 建立勤務組合產生問題限制規劃模式。

藉由實地訪談測試例題所屬之個案公司，並根據限制規劃方法之求解特性建立一個求解效率佳且能考慮各種排班限制(工時、休時、飛時)與排班因素(基地、機型接續、艙等)之一般化勤務組合產生問題限制規劃模式。

(5) 建立八種後艙空勤組員排班模式

根據基地、機型接續、艙等三項因素，建立出八種後艙空勤組員排班模式，其中，最簡單之排班模式為「單基地、單機型接續、單一艙等」，最複雜之排班模式為「多基地、多機型接續、多艙等」。

(6) 建立一套「以限制規劃為基礎之變數產生法」求解架構

根據限制規劃之優點與變數產生法理論，構建出一個執行效率較佳之「以限制規劃為基礎之變數產生法求解架構」，以期改善傳統「以網路模式為基礎之數產生法求解架構」之缺點(包含：模式之彈性、限制式之管理、限制式之透明度、求解效率等)。

(7) 建立一套「限制列舉式之產生-優化法」

充分利用限制規劃模式中可動態增加與刪除限制式之特性，在勤務組合產生問題限制規劃模式中，加入一條「限制航段間接續時間」之限制式，使得排班模式所考慮之勤務組合皆屬優良之勤務組合，一方面可大幅降低排班模式之規模，另一方面可確保在短時間內，此求解架構所求得之解品質不會太差。

(8) 模式之執行 / 修正 / 求解結果分析

根據測試例題資料之特性，確定其所屬之組員排班模式類型(即八種組員排班模式中之哪一種)，接著分別運用上述兩種求解架構進行實例測試，最後再根據測試結果修正勤務組合產生問題限制規劃模式或求解演算法架構。

(9) 結論與建議

提出本研究之具體結論與建議。

本研究之流程圖如圖 1.3 所示。

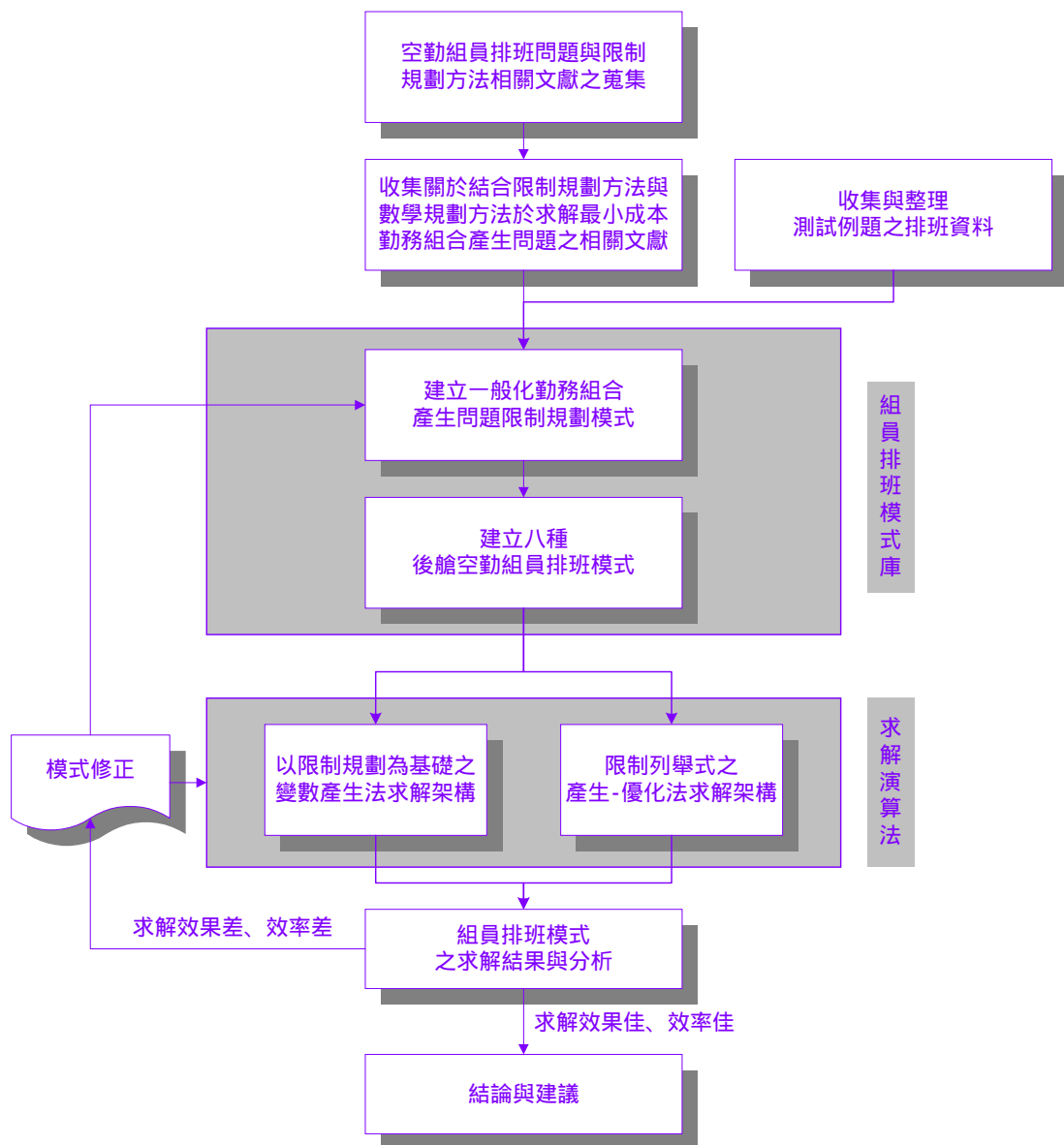


圖 1.1 研究流程圖

第二章 文獻回顧

空勤組員排班問題之複雜度屬於 NP-Hard [29]，而造成此問題如此難於求解之主要原因有二，一為「在產生可行勤務組合時，必須同時考慮複雜的排班法規(Complex Rule)與複雜的成本計算結構(Complex Cost Structure)」，二為「在求解最小成本勤務組合集合時，必須在大規模的可行勤務組合(Large-Scale Feasible Pairings)變數中，選出一組能滿足各航段各艙等空勤組員人數需求的限制且成本最小的勤務組合集合」。

由於空勤組員排班問題很難求解，所以在過去三、四十年來，國內、外一直有著相當多的文獻在探討此類問題之特性與求解方法。然而，詳細回顧此問題之求解方式，可發現，文獻上在求解此問題時，通常將其分解成一個子問題(勤務組合產生問題)與一個主問題(最小成本勤務組合產生問題)，其中子問題之功能乃在於「輔助主問題產生其需要的可行勤務組合」。在子問題之定式上，通常為一個檢驗-回溯模式或是(資源受限制之)最短路徑模式或是限制規劃模式；在主問題之定式方面，通常為一個集合分割(SPP)或集合涵蓋(SCP)之數學規劃模式。在子問題之求解方法方面，相對於子問題之定式包含有檢驗-回溯法或是(資源受限制之)最短路徑演算法或是限制規劃方法；在主問題之求解方法方面，通常方為啟發式解法與最佳解解法兩種，其中啟發式解法之產生主要是由於此類問題之規模過於龐大且早期之電腦運算能力不高，使得排班人員無法透過電腦之輔助，在合理之運算時間內求解大規模之組員排班問題，因而產生許多啟發式解法，而最佳解解法之產生，主要依據數學規劃之相關理論而來，其主要之目的在於求解問題之最佳解，在早期之文獻中，此方法通常只被運用來求解小規模之組員排班問題，然而有鑑於近年來電腦運算能力的大量提升與相關演算法之改進，近幾年來亦紛紛有許多學者開始運用最佳解解法來求解大規模之組員排班問題。

根據上述之相關概念，詳細回顧並整理空勤組員排班問題之相關文獻於下列各小節中。

2.1 組員排班問題

2.1.1 勤務組合產生問題與求解方式

勤務組合產生問題之基本定義為「在滿足種種複雜排班法規限制下，產生多組可行勤務組合，其中每個可行勤務組合是由一組連續的勤務所構成，且勤務組合之起點與迄點城市必須為同一個航空公司組員基地」[34]，由於此問題必須考慮甚多繁雜之排班法規限制，而且一般的法規很難實際用數學式來定式與量化，所以綜觀作業研究之相關文獻，可發現並不存在一個制式化之數學模式作為產生可行勤務組合時之參考，雖然如此，文獻仍然提供下列兩種傳統的列舉法(Enumeration Method)來產生可行勤務組合，此兩種方法之相關細節如下所述：[5]

1. 檢驗-回溯法(Test-and-Backtrack Method)：

運用一般之程式語言(如：C++/VB/Delphi 等)將「列舉策略(Enumeration Strategy)」與「法規限制」寫成電腦邏輯程式邏輯程式(Computer Logic Program)來列舉(Partial Enumeration)或窮舉(Complete Enumeration)可行勤務組組合，由於種求解方法之運作方式為「依序選取每個航段並將其加入至勤務組合中，檢查此航段的加入是否會違反部分法規限制，如果違反的話，則回溯(backtrack)至上一狀態進行另一航段的加入，如果符合的話，則將此一航段加入勤務組合；持續執行此一程序，直至找到一個可行勤務組合(若是要找尋所有可行勤務組合，則是根據上述之運作方式找尋所有可能之航段組合方式，直至所有可行勤務組合都被找出來為止)」。

就列舉效率(Efficiency of Enumeration)而言，此方法與傳統空間搜尋演算法中之換枝檢驗法(Back-Tracking, BT)同樣具有「無法事先避免未來之衝突產生」、「無法避免一樣的錯誤再度發生」等缺點，所以相當沒有效率，尤其在求解規模大且複雜度高之勤務組合產生問題時，最容易感覺到此一缺點之嚴重性；就列舉方法而言，此方法具有「法規限制式透明度低」、「電腦邏輯程式維護不易」、「列舉彈性低」等缺點，所以此方法無法有效地輔助排班模式產生或排班人員產生其所要的勤務組合類型。然而，由於此方法具有相當多的缺點，因此一般使用此一方法來產生可行勤務組合之文獻並不多(國內的文獻有 1991 年沈志展[40]、國外的文獻有 Crainic and Rousseau[11]等)。

2. 結構化網路模式(structured network model)：(如 Desrochers and Soumis [13])

根據相關之排班限制，將勤務組合產生問題構建成一個結構化網路模式 $G=\{N,A\}$ ，亦即所謂的「排班網路」，由於航段間之接續尤其時間之限制，所以排班網路 G 必為一個方向性(acyclic)路網。一般而言，排班網路可依據下列兩項因素來分類：

a. 依「節點與節線所代表之含意」來區分：

以 leg/duty 表示節點之排班網路

節點集合 N ：

起始、結束節點：起始(source node)、結束(sink node)兩個虛擬節點

leg/trip 節點：代表每個 leg/duty 之節點

節線集合 A ：

起始、結束節線：連接起始節點或結束節點至其 leg/duty 節點的節線

接續節線：連接兩個 leg/duty 節點之節線

以 leg/duty 表示節線之排班網路

節點集合 N ：

起始、結束節點：起始(source node)、結束(sink node)兩個虛擬節點

leg/trip 節線之端點節點：代表每個 leg/duty 節線之起點或終點節點

節線集合 A：

起始、結束節線：連接起始節點或結束節點至其 leg/duty 節點的節線

leg/duty 節線：表示每個表每個 leg/duty 之節線

接續節線：連接兩個 leg/duty 節點之節線

b. 依「是否符合所有法規限制」來區分：

符合全部法規限制之排班網路

一般而言，很難將一個人員排班問題構建成一個符合各項法規之排班網路，除非是在某些特殊結構之排班問題中，可試圖藉由忽略某些型態之可行勤務組合來促使其排班網路為一個完全符合所有法規限制之排班網路，如：嚴上堯、湯敦台[42]與嚴上堯、杜宇平[43]。此種排班網路最大之特性為「排班網路中之每條路徑，皆表示一個可行勤務組合」，拜此特性所賜，若要求解排班網路中成本最小之勤務組合時，可直接引用網路演算法中之最短路徑演算法來求解。

符合部分法規限制之排班網路

大部分文獻上之排班網路皆屬於這一類，如：王晉元、盧宗成[44]、[13、14、15、23、44]。此種排班網路最大之特性為「任何一個可行勤務組合，相對應至排班路網上即表示一個路徑(Path)；反之，排班路網上之任何一個路徑並不一定代表一個可行勤務組合，因為此路徑可能違反部分資源限制式(如：最大飛時、最大工時、最短休時等)」，由於此一排班網路特性，若要在排班網路中求得成本最小之可行勤務組合，就必須運用多重標籤最短路徑演算法(Multi-Labeling Correcting Method)或是限制性最短路徑演算法(Constrained Shortest Path Procedure)來求解，此種方式之缺點是它很難去表示(express)與維護(maintain)現實的複雜限制式，亦即當現實的複雜限制式改變時，限制性最短路徑演算法即必須重新設計。

雖然此種勤務組合產生方式，在作業研究領域當中已經被視為一種相當不錯的求解方式，尤其當此種以網路模式為基礎之勤務組合產生方式被運用在變數產生法求解架構中之子問題求解上，更是目前所有相關排班文獻最常使用之一種求解模式，然而仔細探討此種勤務組合產生方法之缺點，可將其綜合歸納如表 2.1 所示。

表 2.1 運用網路模式產生可行勤務組合之缺點

| 缺點 | 描述 |
|-----------|---|
| 模式之彈性差 | 無法有效地輔助排班人員產生其所需要之可行勤務組合類型 無法有效地輔助排班模式產生其所需要之可行勤務組合類型 |
| 排班限制完全不透明 | 將所有所需考慮之排班法規放置在排班網路中或是放在資源受限制之最短路徑演算法之中 |
| 限制式之管理不易 | 對於面臨多變的航空產業而言，若部分法規更動，則限制性之最短路徑演算法必須重新設計方能有效地求解子問題，無法隨時地、動態地對限制式進行管理(新增、刪除、修改等) |

2.1.2 最小成本勤務組合產生問題

所謂最小成本勤務組合產生問題乃意指「在滿足種種複雜的排班法規限制下，產生一組成本最小且涵蓋所有航段的可行勤務組合集合」[34]。目前文獻上(如 Arabeyre [1]、Marsten [28]等)大多將此問題建構成一集合分割(Set Partition)或集合涵蓋(Set Covering)之數學規劃模式。其基本之數學規劃模式如下【P₁】所示：

$$\text{【P}_1\text{】:} \quad \text{Min} \quad \sum_{j \in P} c_j \times x_j \quad (1) \quad \text{對偶變數}$$

st

$$\sum_{j \in P} a_{ij} \times x_j = (\geq) 1, \forall i \in F \quad (2) \quad \pi_i$$

符號說明：

$$x_j \in \{0, 1\} \quad (3)$$

$$a_{ij} = \begin{cases} 1, \text{表示勤務組合 } j \text{ 包含航段 } i \\ 0, \text{otherwise.} \end{cases} \quad (4)$$

$$c_j : \text{合法勤務組合 } j \text{ 之成本} \quad (5)$$

$$P \text{ 為所有可行勤務組合所形成之集合} \quad (6)$$

$$F \text{ 為所有航段所形成之集合} \quad (7)$$

數學規劃模式【P₁】中的每一個變數 x_j (或稱為 column)代表一可行的勤務組合(Feasible Pairing)，而所謂可行的勤務組合是指符合各種法規限制下產生的勤務組合。如果一個可行勤務組合 j 被選到，則表示 x_j 為 1，否則為 0。數學規劃模式【P₁】中之(2)式為集合分割限制式，其目的在限制每一個航段只能被涵蓋(服務)一次，亦即集合分割模式可禁止航段被重複涵蓋(over-cover)的情形發生，但就求解之困難度而言，集合涵蓋模式之求解較集合分割模式困難許多；若將其中之等式改為不等式()，則成為集合涵蓋問題模式，其表示允許重複涵蓋(over-cover)相同航段 1 次以上，意即允許空載的情況發生。然而，由於集合分割常無法求得最佳解，所以，大多文獻都建議採用集合涵蓋問題模式來表示最小成本之勤務組合產生問題。

運用集合分涵蓋(SCP)或集合分割模式(SPP)來求解最小成本勤務組合產生問題之優點，可歸納如下：

1. 將「排班法規」與「排班需求」兩項因素分開考慮，可以大幅低問題求解時的複雜度。其中「排班法規」因素必須要在產生可行勤務組合時考慮，完全不需要在 SCP 或 SPP 模式中考慮，而「排班需求」因素乃在 SCP 或 SPP 模式中考慮。

2. 當問題規模不大(小於 10 萬個變數)時,可直接使用整數規劃方法求得問題之最佳解;當問題規模太過於龐大(大於 10 萬個變數)時,可使用變數產生法、Branch-and-Cut-and-Price 等方法來求得近似解。

2.1.3 勤務組合產生問題之相關成計算

關於勤務組合成本之計算方式,文獻上通常將其分為「勤務成本之計算」與「勤務組合成本之計算」兩個階段,其計算方式如下所述:[34]

1. 勤務成本之計算:

由於勤務為空勤組員之最小工作單位,所以一般而言,不管此一勤務之時間是長或短,航空公司皆一定會給空勤組員一個基本保障薪資,亦即,空勤組員執勤任何一項勤務時,皆有一固定之最小保障時數以保障組員薪資,其計算公式如【式 4.1】所示。由其計算公式中可明顯發現,勤務成本一定大於等於勤務之總飛行時間成本。

$$C_d = \max \left\{ \text{dmg, ratio} * \text{duty_elapsed_time}, \sum_{j \in d} \text{fly_h}_j \right\} \quad \text{【式 4.1】}$$

符號說明:

C_d : 勤務 d 之時間成本(單位:小時)

dmg(duty minimum quaranteed hours): 組員執行一個勤務之最小保障時數(單位:小時)

ratio: 航空公司自訂之比率

duty_elapsed_time: 勤務之總歷時時間(單位:小時)

fly_h_j: 勤務中,航段 j 之飛行時間(單位:小時)

$\sum_{j \in d} \text{fly_h}_j$: 勤務之總飛行時間

2. 勤務組合成本之計算:

由於勤務為構成勤務組合之基本單位,所以勤務組合成本之計算方式乃以勤務成本之計算方式為基礎,其計算方式如【式 4.2】所示。由其計算公式可明顯發現,勤務組合成本一定大於等於勤務組合中之總飛行時間成本。

$$C_p = \max \left\{ \text{dmg} \times \sum_{d \in p} 1, \text{ratio} * \text{TAFB}, \sum_{d \in p} C_d \right\} \quad \text{【式 4.2】}$$

符號說明：

C_p ：勤務組合 p 之時間成本(單位：小時)

$\sum_{d \in p} 1$ ：勤務組合中所包含之勤務數

ratio：航空公司自訂之比率

TAFB：勤務組合之差旅時間(單位：小時)

$\sum_{d \in p} C_d$ ：勤務組合中，所有勤務之時間成本總和(單位：小時)

綜觀上述之勤務、勤務組合成本之計算方式，可發現下列幾項事實：

- 勤務組合之成本計算公式，一定包含「勤務組合之差旅時間(TAFB)」與「勤務組合之總飛行時間」兩項元素。
- 勤務組合成本之下限值乃以勤務組合中之總飛行時間成本為限。
- 勤務組合中之航段接續時間越長，表示組員閒置在外地之時間越長、勤務組合之差旅時間亦越長，所以航空公司所需花費之薪資成本亦越高
- 當勤務組合之總飛行時間越長，航空公司就必須支付更多之薪資成本。
- 當勤務組合之差旅時間越長，航空公司就必須支付更多之薪資成本與可能發生之住宿成本。

由於勤務組合成本之高低直接影響航空公司之成本支出，所以為了降低航空公司之成本支出，文獻上大多建議後續研究人員在產生勤務組合時，盡量產生優良之勤務組合，而其所謂之優良勤務組合乃指「勤務組合中之 差旅時間 TAFB / 總飛行時間 比值較低之勤務組合」，至於此比值要低至何種程度，則需視排班資料之特性而定。然而，為何勤務組合之 差旅時間 TAFB / 總飛行時間 之比值越低表示此勤務組合越優良呢，其主要之原因乃在於「由於勤務組合之總飛行時間成本為勤務組合成本之下限值，而且勤務組合之差旅時間為總報到時間與總報離時間、總休息時間與總休憩時間、總飛行時間之加總，所以 差旅時間 TAFB / 總飛行時間 之比值越低，則表示組員之總休息時間與總休憩時間越短，亦即此類型之勤務組合能以較低之薪資成本來提高組員之使用率」。

2.2 最小成本勤務組合產生問題特性之探討

空勤組員排班問題的類型，大至可依據排班問題之特性分為下列幾類[34]：

1. 以組員來區分：

a. 前艙組員排班(cockpit crew scheduling)：

前艙組員通常有正機師(Captains)、副機師(First Officers)及機械員，由於此三者之職務所受的專業飛行訓練不同，使得前艙之每位組員可值勤之機型受到嚴格的限制，而航空公司為了使前艙組員盡量熟悉特定之機型以提高飛航安全，前艙組員通常僅執勤某一特定之機型，所以在進行求解「前艙組員之最小成本勤務組合產生問題」時，通常會先將相同機型之航段群組起來，然後再依序根據各機型產生其單機型接續之最小成本勤務組合產生問題。

b. 後艙組員排班(cabin crew scheduling)：

後艙組員通常包括一位座艙長(Purser)及若干位男、女空服員，由於後艙組員在經過專業訓練之後大都能服務各機型，所以在進行「後艙組員之最小成本勤務組合產生問題」時，通常都會將所有各機型之航段一起考慮，而且由於不同機型之艙等配置不同、不同機型不同艙等中所需要之艙等組員數不同，因此，後艙組員之最小成本勤務組合產生問題會較前艙組員來的複雜。

2. 以國內航線與國際航線來區分：

a. 國內航線之組員排班問題(domestic crew scheduling problem)：

一般而言，國內航線之組員排班問題大多亦屬於短程航線之組員排班問題(如：台灣國內航線、日本國內航線等)，但是對於國家領土廣大之國家(如：美國、大陸等)而言，其國內航線就不一定屬於短程航線之組員排班問題。

b. 國際航線之組員排班問題(international crew scheduling problem)：

國際航線之組員排班問題亦屬於長程航線之組員排班問題。

3. 以航程長短來區分：

a. 短程航線之組員排班問題(short-haul crew scheduling problem)：

由於在此類問題中，每個航段之航程均較長程航線短，所以在同一工作時間之法規限制下，其每個勤務所能包含之航段數可較多，通常介於 4-8 個航段之間。

b. 長程航線之組員排班問題(long-haul crew scheduling problem)：

由於在此類問題中，每個航段之航程均較短程航線長，所以在同一工作時間之法規限制下，其每個勤務所能包含之航段數較少，通常介於 1-4 個航段之間，且大多之勤務只包含 1 個航段或 2 個航段，包含 3 個或 4 個航段之勤務較少見。

4. 以規劃時程(planning horizon)長短來區分：

a. 以天為基礎之組員排班問題(Daily-based crew scheduling problem)

每天或每週之航線班次大多不同且班次之異動頻率較高，不適合採用 1 週或 1 個月作為規劃時程之組員排班問題，然而由於此類形之排班問題需要排班規劃人員天天執行排班作業，所以其所花費在組員排班上之人力成本較高。

b. 以週為基礎之組員排班問題(Weekly-based crew scheduling problem)

對於每週的航線班次大致相同且航段航程較短之組員排班問題，航空公司通常會採用以 1 週作為其排班規劃之時程，以降低航空公司花費在組員排班上之人力成本。

c. 以月為基礎之組員排班問題(Monthly-based crew scheduling problem)

對於每週之航線班次大致相同且班次之異動頻率較小之組員排班問題，航空公司通常會採用以 1 個月作為其排班規劃之時程，以降低航空公司花費在組員排班上之人力成本。

5. 以區域來區分：[3]

a. 美國式之組員排班問題(American-Type crew scheduling problem)：

就「飛航網路結構」而言：北美區域屬軸輻式網路結構(Hub-and-Spoke)，所以每當某一航班飛抵至空運中心(Hub)後不久，就會有許多可接續之航班由此空運中心飛往其他城市，Graves et al. [17]指出每一航班抵達芝加哥之後不久就會有 100 多個航班從芝加哥起飛，這種飛航網路結構將使得航空公司中之勤務接續可能性組合增多，使得美國區域航空公司之最小成本勤務組合產生問題複雜度變的相當高。

就「排班法規」而言：美國區域之排班法規主要以美國聯邦航空總署(Federal Aviation Administration, FAA)所規定之排班規則與工會法規為主，其主要限制了勤務組合之長度(Length of Pairing)、勤務段之長度(Length of Duty Period)及勤務段間的最小休息時間，而每家航空公司之工會法規差異不大，所以對於北美區域之航空公司而言，其最小成本勤務組合產生問題的結構大致相同。

就「航班時刻之規則性」而言：北美區域之航空公司其週一至週五間每天之航班時刻皆一模一樣，而在週末時，只有減少部分航班而已，大致上航班時刻還是相同。

就「組員薪資結構」而言：北美區域之航空公司之組員薪資主要以空勤組員之飛行加給時數(Credit Hours)為基本計算單位，而每個勤務組合之飛行加給時數之計算乃是取「勤務組合之差旅時間(Time Away From Base, TAFB) * 一個比例、每勤務之最小保證時數(minimum guaranteed hours per duty)*勤務組合中之總勤務數、勤務組合中所有勤務段成本之總和」三者中最大值者，其中每個勤務段之成本計算則是取「每勤務之最小保證時數、勤務段之總歷時時間(包含組員報到與報離時間)、勤務段中之總飛行時間」三者中最大值者，由此可見，北美區域航

航空公司之最小成本勤務組合問題支解品質與總飛行加給時數除以總飛行時間的筆值($\text{Credit Hours} \div \text{Total Flying Time}$)有關，換句話說，當此比值越小表式解的品質越高，所以北美區域之航空公司可以根據此原則來求產生此比率越低之可行勤務組合。

b. 歐洲式之組員排班問題(European-Type crew scheduling problem)：

就「飛航網路結構」而言：由於歐洲區域之飛航網路結構並不是這種軸輻式網路結構，所以歐洲區域之航空公司之勤務接續可能性組合較少，所以歐洲區域航空公司之可行勤務組合數量會較北美區域航空公司少很多且問題也較來得簡單。

就「排班法規」而言：歐洲區域之航空公司，由於工會之規定比政府的排班法規還嚴格且變動性又高，而且每家航空公司之工會規定又不太相同，所以對於歐洲區域航空公司而言，每家航空公司之最小成本勤務組合產生問題的結構皆不同且差異性大。

就「航班時刻之規則性」而言：歐洲區域之航空公司，其每天的航班時刻皆不同，而且常常會有一週一、二、三次的航班出現。

就「組員薪資結構」而言：對於歐洲區域之航空公司，由於其薪水之給定採用固定薪水制(fixed salary)，所以較無一定準則可以產生合適的可行勤務組合。

綜合上述，將北美與歐洲區域航空公司排班問題之相異特性整理成如表 2.2 所示。

表 2.2 北美與歐洲區域航空公司排班問題之相異特性

| 相異特性 \ 區域 | 北美區域 | 歐洲區域 |
|-----------|---------------------------|--------------------------|
| 網路結構 | 軸輻路網之飛航網路結構 | 無任何特定之飛航網路結構 |
| 排班法規 | FAA 飛航管制規則最嚴格 | 工會的規定比政府的法規規定還嚴格，並且還時常更換 |
| 航班時刻之規則性 | 星期一至星期五之航班大致固定，而在週末只減部分航班 | 每天的航班都不太固定 |
| 組員薪資結構 | 依據空勤組員之 Credit Hours 給薪 | 組員皆給一固定之薪資 |

資料來源：[3]

2.3 最小成本勤務組合產生問題之求解演算法

有關最小成本勤務組合產生問題之求解演算法相關文獻數量十分龐大，但主要可區分為最佳解法(Exact Solution Methods)與啟發式解法(Heuristic Solution Methods)兩種，但是由於最小成本勤務組合產生問題屬於 NP-hard 的問題，因此，文獻上大多是採用啟發式解法。本節將針對此二類求解方法之相關文獻作一回顧。

2.3.1 啟發式解法

由於最小成本勤務組合產生問題屬於 NP-Hard 的問題，因此在實際的應用上，大多是使用啟發式解法來求解。而一般用於求解最小成本勤務組合產生問題的啟發式解法有：拉式啟發式解法(Lagrangian Heuristic)、基因演算法(Genetic Algorithm)。

1. 拉式啟發式解法

顏上堯與杜宇平[43]以拉式鬆弛法暨次梯度法(Lagrangian Relaxation with Subgradient Methods)的演算法架構進行求解，其作法是先將原問題之額外限制式鬆弛成為最小成本之純網路流動問題加以求解，此解即為原問題之下限解，並運用其自行發展之啟發式解法求得一符合原問題額外限制式之可行解，即為問題之上限解。然後以次梯度法修正拉式乘數之值，重複求解上、下限，直到上下限收斂至一許可值為止。

Baseley[7] 提出公車司機排班問題，將問題模式構建成集合分割形式並提出求解策略，也就是在以求解最佳解為基礎下發展演算法，求解的時候利用拉氏鬆弛法將集合分割的限制式放鬆，然後利用一個自行發展的啟發式策略，修正拉氏解使其搜尋到較優的可行解當成上界也就是此模式的解，再利用次梯度法法搜尋目標值的下界，經過多次的疊代，找到下限。此方法發展之啟發式策略所找到的解雖然不是最佳解，但是可以利用次梯度法得到的下界來評估與正確解的距離。若是配合一個有效率的樹狀搜尋 (tree search) 來做分支，可以加速分枝定限法求得最佳解。

2. 基因法

基因演算法為近十年來較流行的方法，其觀念為加入人工智慧的特性，使可行解的搜尋過程有機會能跳脫區域最佳解(Local Optimal)的情況。Levine[24]發展出 Hybrid Genetic Algorithm 用來求解航空公司人員排班問題，其概念主要是結合了基因法與區域搜尋法兩種方法，其運作方式為，先運用基因演算法在可接受的時間內找到一個較佳解後，再利用區域搜尋法來找尋更好的區域最佳解。

Wren[37]於 1995 年發展一基因演算法來求解集合分割的問題，其採用基因演算法的主要原因為：1.基因法所得到的解為一個可行解，故不像其他方式需要一些調整的策略；2.在處理額外的限制式上，處理調整模式的改變上比較方便，較具有彈性。3.由於其求解出的是一個群組的候選解，在對於實際問題上，數學模式很難得到滿足所有現實環境的變因，故演算法所得到的最適解未必能滿足現實環境所需，所以得到一群好的解

比得到一個最適解來的好。

2.3.2 最佳解解法

目前有關最小成本勤務組合產生問題之最佳解解法文獻，其作法通常是將整個問題構定式成集合分割(Set Partitioning)或集合涵蓋(Set Covering)模式(如【P₁】)，並配合最佳解的求解策略加以求解。然而，關於求解集合涵蓋(或集合分割)問題的最佳解求解方法，相關文獻中最常用的是分枝定限法(Branch-and-Bound)、分枝切面法(Branch-and-Cut)與變數產生法(Column Generation)，以下將針對此前二種方法作一簡單介紹，至於變數產生法之介紹則將在下一節中有更詳細且完整之介紹：

1. 分枝定限法(Branch-and-Bound)

分枝定限法(Branch-and-Bound)為傳統上最常用來求解整數規畫問題的最佳解解法。由 Hiller and Lieberman[19]的說明我們可知，分枝定限法的基本觀念是分割以及解決 (divide and conquer)。分枝定限法主要包含四項基本步驟：節點選擇 (Node Selection)、分枝 (Branching)、定限 (Bounding) 以及解決 (Fathoming)，其中各步驟之詳細內容如表 2.3 所示。

表 2.3 分枝定限法之四項求解步驟

| | |
|--------------------------|---|
| 節點選擇 (Node Selection) | 由於在分枝定限法的過程中，可能尚有許多未求得整數解的分枝端點，因此，如何從這些未選節點中進行求解工作即是節點選擇之主要工作。而一般常用的選擇方式則包含：最佳目標值法 (Best-Bound Search)、先深度搜尋法 (Depth-First Search)、先廣度搜尋法 (Breadth-First Search) 以及隨機搜尋 (Random Search) 等 |
| 分枝 (Branch) | 當處理二元 (Binary) 變數時，最直接的方法就是將合理解集合固定一個變數值 (假設為 x_1) 亦即此時將原本的大問題分割為 $x_1 = 0$ 以及 $x_1 = 1$ 的二個子集合所形成的次問題。故以分枝樹來看，即可發現其針對 x_1 此變數而分枝出二個分枝端點 |
| 定限 (Bound) | 對每一個次問題，我們找出它的線性最佳解，亦即該次問題的界限解，最標準的方法即是快速求解該次問題的鬆弛解 (Relaxation)。在大多數的情形，一個問題的鬆弛解是簡單的去掉導致問題難以求解的限制式而成。對整數規畫的問題而言，製造困難的就是要求各變數為整數的限制。因此，最常用的鬆弛解就是把這些整數限制去除 |
| 解決 (Conquer) | 當一個次問題 (分枝端點) 被解決後，它就不必再考慮，而分枝定限法之主要目的即是在於將所有的分枝端點加以解決。其有三種情形可以被解決：① 次問題之最佳解為整數解、② 次問題為不可行解 (Infeasible)、③ 次問題之最佳解較目前暫時解 (Incumbent) 為差 |

2. 分枝切面法(Branch-and-Cut)

分枝切面法可視為分枝定限法與切平面法之結合。根據 Young[1968]的定義，分支切面法是在計算分支端點前嘗試加入新的有效不等式，用以縮小放鬆整數限制後的可行解區域，而此即為分枝切面法用來修正分枝定限法之基本步驟，其主要目的是希望藉由縮小放鬆整數限制後的可行解區域，來加快分枝定限之求解速度。

對於分枝定限法而言，若計算分支端點後得到非整數解，則針對某一個變數繼續分支，並加以持續計算。但是對於分支切面法而言，若計算分支端點後得到非整數解，則在分支之前，會先設法得到一有效不等式（即所謂的切割（cut）），並且將其加入模式後再重新求解一次。若仍為非整數解，此時則有二種選擇方案可供繼續進行：第一，可再設法加入其他的有效不等式，並重新求解；第二，直接進行分支作業。

以 Hoffman and Padberg[20]提出之分支切面法為例，其將演算法分為四大步驟：① 前置處理（Preprocessing）② 模式鬆弛求解（LP Solver）③ 啟發式解法求取可行解（Heuristic）以及④ 產生有效不等式（Constraint Generator）。

作者首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。而有別於傳統的分枝定限法，其在分枝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。倘若仍然無法有效求解，則進入該演算法之核心部分 - Constraint Generator。作者將簡化後之模式的係數矩陣轉化成為一個路網圖，並利用該路網圖，獲得數個有效不等式，藉以去除線性鬆弛後之非整數解。同時，藉由有效不等式的產生，使模式的可行解區域不斷緊縮（tighten），以加速分枝定限法之求解效率。最後，以作者的例題測試結果而言，分支切面法應已能處理較為大型之問題。

在利用集合涵蓋(分割)方法求解最小成本勤務組合產生問題時，如果能在求解的過程中證明所有可行的勤務組合都被考慮到，且可以求得一最佳解，則表示此一解為整個問題的最佳解。然而，由於許多實務問題的可行勤務組合可能有數十億個且目前文獻上並無有效的分枝定限或分枝切面演算法來求解大規模的集合涵蓋(分割)問題，所以在求解實務問題時，實在不太可能實際考慮(explicit consideration)所有可行的勤務組合，因而轉而採用隱含考慮(implicit consideration)所有可能可行勤務組合之變數產生法(column generation)來求解。

2.4 變數產生法

變數產生法(Column Generation)或是稱為 Dantzig-Wolfe Decomposition, 是由 Dantzig 和 Wolfe[12]兩位學者於 1960 年時共同提出, 其求解概念為「將原來的大規模主問題 (Master Problem) 分為受限制主問題(Restricted Master Problem)與子問題(Sub Problem)兩部份, 然後運用線性規劃中的對偶理論(Dual Theory)將受限制主問題與子問題緊密的串連起來」, 其運作方式為「先透過人工方式或啟發式解法求得一組變數集合放入受限制主問題並確保此一受限制主問題存在一個基本可行解, 然後放鬆受限制主問題並求解得到一組對應每個限制式之對偶變數向量, 接著將此組對偶變數向量放入子問題中, 使子問題能運用對偶理論來產生對目前受限制主問題目標值有貢獻之新變數(column)並加入受限制主問題之中, 藉由受限制主問題與子問題之互動, 逐步改善受限制主問題之 LP 最佳解, 直至無法再改善為止, 最後所得之解即為放鬆後受限制主問題之最佳解, 若所得到的最佳解為整數解, 則此解即為目前受限制主問題之最佳解, 若不為整數解, 則需再採用分枝定限法以獲得目前受限制主問題之整數解」, 此方法論之優點有二, 一為「強而有力的理論基礎」, 二為「演算法求解效率佳而且可避免因問題規模過大而無法有效求解」。

由於有些實務問題之規模過於龐大(幾百萬或幾千萬個變數), 可能使得一般電腦硬體無法完全將模式記錄在電腦之中, 然而, 即使電腦硬體可以紀錄如此大規模之問題模式, 但目前亦尚無一個有效求解大規模問題之演算法可快速的求得最佳解。因此, 對於處理大規模之集合分割或集合涵蓋問題時, 一般文獻上大都建議使用變數產生法, 因為運用變數產生法來求解大規模之集合分割或涵蓋模式, 可大幅地避免浪費過多的時間去窮舉對問題目標式毫無貢獻的變數, 使得原來需要完全考慮的大規模主問題變成只要考慮規模比較小的受限制主問題, 然後再運用目前比較好的演算法來求解此一規模較小的受限制主問題之最佳解, 就求解效率而言, 變數產生法之求解效率較完全窮舉所有變數後再求解的方法有效率, 就求解品質而言, 雖然, 受限制主問題之最佳解未必等於原來之大規模主問題最佳解, 但是解的品質亦不會相差太多, 一般多為實務上應用可接受之成本誤差之內。

【P₂】為一個放鬆後之集合涵蓋問題, 而【P₃】為【P₂】之對偶問題。根據單體法 (simplex method) 之對偶可行性(dual feasibility)可知: 若可證明對所有尚未考慮的變數均滿足 Reduced Cost 大於等於 0 ($c_j - \sum_{i \in F} a_{ij} \pi_i \geq 0, \forall j \in P$) 之條件時, 則找到之解即為放鬆後主問題之最佳解, 但並不代表一定為主問題之最佳解, 因為在集合分割或集合涵蓋模式中之主問題大都為整數規劃模式; 否則, 我們可以從尚未考慮的變數中找到一個新的變數 j 或一組新的變數集合 $\{j\}$ 且符合 Reduced Cost 小於等於 0 ($c_j - \sum_{i \in F} a_{ij} \pi_i \leq 0$) 之條件, 並將此新的變數加入主問題中, 可以讓主問題的目標值更好。

$$\text{【P}_2\text{】: } \quad \text{Min } \sum_{j \in P} c_j x_j \quad (1) \quad \text{對偶變數}$$

st

$$\sum_{j \in P} a_{ij} x_j \geq 1, \quad \forall i \in F \quad (2) \quad \pi_i$$

$$0 \leq x_j \leq 1, \quad \forall j \in P \quad (3)$$

$$\text{【P}_3\text{】: } \quad \text{Max } \sum_{i \in F} \pi_i \quad (1)$$

st

$$\sum_{i \in F} a_{ij} \pi_i \leq c_j, \quad \forall j \in P \quad (2)$$

$$\pi_i \geq 0 \quad (3)$$

綜合歸納文獻上目前所提之變數產生法求解架構，可分為「以網路模式為基礎之變數產生法(Network Model Based Column Generation)」與「以限制規劃為基礎之變數產生法(Constraint Programming Based Column Generation)」兩大類，此兩種求解架構之最大不同點在於其子問題所使用的模式與求解演算法，亦即，前者使用最短路徑問題模式(資源受限制之最短路徑問題模式)與最短路徑演算法(資源受限制之最短路徑演算法)，而後者使用限制規劃模式與限制規劃方法。以下將分別簡單介紹此兩種變數產生法求解架構中之起始解產生方式、子問題求解方式以及其相關之排班文獻：

2.4.1 以網路模式為基礎之變數產生法

綜觀文獻上所提之各種以網路模式為基礎之變數產生法架構，其主要之不同點主要在於其求解架構中之起始解產生方式與子問題求解方式，經由歸納整理，其各可分為下列幾種方式：

1. 起始解產生方式

a.人工方法：

產生一組涵蓋每個航段、但其成本極大且不可行的勤務組合。

b.啟發式解法：

運用網路流量守恒觀念與最短路徑演算法，建構一個不錯的起始解，([42])。

運用程式邏輯，依序產生包含所有航段可行勤務組合。

2. 子問題求解方式

a. 最短路徑問題模式與最短路徑演算法：

將勤務組合產生問題構建成一個考慮各種法規限制之排班網路，使得排班網路中之每條路徑(Path)皆代表一個可行之勤務組合，然後透過受限制主問題中每個航段所對應之對偶變數值與最短路徑演算法，尋找出(Price Out)對目前受限制主問題最有貢獻之可行勤務組合(即目前排班網路狀態之最短路徑)。

b. 資源受限制之最短路徑問題模式與資源受限制之最短路徑演算法：

由於問題之特性與法規之複雜性，有時無法將勤務組合產生問題構建成一個考慮全部法規限制之排班網路，而相對的只能將勤務組合產生問題構建成一個資源受限制之排班網路，由於其中之每條路徑(Path)不一定能代表一個可行之勤務組合，所以必須藉由資源受限制之最短路徑演算法，來找出(Price Out)對目前受限制主問題最有貢獻之可行勤務組合。

在子問題的模式與時空網路構建上，Desrochers、Gilbert、Sauve and Soumis[14]有相當詳細的介紹與討論。

3. 相關文獻

有關於運用網路模式方法於子問題求解之相關文獻，茲整理如表 2.4 所示，並分述如下：

Gilmore and Gomory[18]曾利用變數產生法求解 Cutting Stock 問題，而變數產生法最早用於排班問題上，則可追溯至 Appelgren[4]應用在求解具時間窗限制的船舶排班問題上。

Minoux et al.[23]採用變數產生法發展求解空勤組員排班模式，在其求解過程中，先以人工方法產生足夠數量之可行變數，並以此作為起始解，其主問題為集合涵蓋問題，利用單體法自現有變數集合中找最佳解；子問題為最短路徑問題，利用標籤修正法(Label Correcting Algorithm)找到一個對主問題目標值有貢獻的變數，將其加入主問題中重新求解，以此程序改善目前解直到無法改善為止，在其 12 個測試問題中，有 9 個可以找到最佳解，充分證實變數產生法之求解效果。

Desrochers and Soumis[13, 14]曾利用變數產生法求解加拿大蒙特婁市(Montreal)之公車人員排班問題，其主問題為一集合涵蓋問題，利用單體法求解，子問題為限制性的最短路徑問題，利用動態規劃方法(Dynamic Programming)來求解對目前受限制主問題目標值有貢獻之新變數並加入受限制主問題中重新求解，依此步驟逐步改善以獲得最佳解，最後再配合分枝定限法以獲得最佳整數解。

Barnhart et al.[8]將長程航線之空服員排班問題(Long-Haul Crew Pairing Problem)定式為集合分割問題，其考慮將可能之空載(Deadhead)節線，以 price-out 方法加入次問題，期尋找更佳之組員行程以利主問題最佳解之求解。

國內顏上堯、林錦翌[40]針對單基地之組員排班問題，發展空勤組員勤務組合最佳化模式，利用變數產生法求解，其求解觀念為「將主問題定式為一集合涵蓋問題，子問

題定式為最短路徑問題，並利用單體法與標籤修正法分別求解主、次問題」。顏上堯、湯敦台[42]針對多基地空勤組員排班問題進行研究，其模式架構和求解流程與顏上堯、林錦翌[40]相似。

國內王晉元、盧宗成[44]針對捷運司機員之排班問題，發展一套捷運司機員排班模式，並運用變數產生法求解，其求解觀念為「將主問題定式為一集合涵蓋問題，子問題定式為資源受限制之最短路徑問題，並利用單體法與資源受限制之最短路徑演算法分別求解主、次問題」。

表 2.4 以網路模式為基礎之變數產生法文獻

| 作者 | 年代 | 問題 |
|--------------------|------|---------------|
| Appelgren | 1961 | 具時間窗限制的船舶排班問題 |
| Minoux et al. | 1988 | 空勤組員排班問題 |
| Desrochers, Soumis | 1989 | 公車排班問題 |
| 顏上堯、林錦翌 | 1995 | 單基地之空勤組員排班問題 |
| 顏上堯、湯敦台 | 1996 | 多基地之空勤組員排班問題 |
| Barnhart et al. | 1998 | 長程航線之空勤組員排班問題 |
| 王晉元、盧宗成 | 1999 | 捷運排班問題 |

2.4.2 以限制規劃為基礎之變數產生法

由於限制規劃方法是進幾幾年來才開始受到作業研究領域學者之重視，所以綜觀相關文獻可發現，運用以限制規劃為基礎之變數產生法來求解空勤組員排班或公車司機員排班問題之文獻並不多，而且此類文獻大多皆為概念性、敘述性文章，並沒有詳細針對限制規劃之特性，提出最能充分運用限制規劃特性之變數產生法求解架構。回顧以限制規劃基礎之變數產生法相關文獻，將其所提及之起始解產生方式與子問題求解方式，整理歸納如下所述：

1. 起始解產生方式

一般而言，皆直接使用文獻上所提之起始解產生方式

a. 人工方法：

產生一組涵蓋每個航段、但其成本極大且不可行的勤務組合。

b. 啟發式解法：

運用網路流量守恆觀念與最短路徑演算法，建構一個不錯的起始解。

運用程式邏輯，依序產生包含所有航段可行勤務組合。

2. 子問題求解方式

其子問題求解方式之基本概念為「使用勤務組合產生問題限制規劃模式與限制規劃方法來求產生一組對目前受限制主問題目標值有貢獻之勤務組合變數」。

3. 相關文獻

有關於運用限制規劃方法於子問題求解之相關文獻，茲整理如表 2.5 所示，並分述如下：

1996 年 Halatsis et al.[21]運用限制規劃於求解前艙組員之排班問題問題，並運用 ILOG Solver、ILOG CPLEX 與 ILOG Views 程式庫替歐洲的一家奧林匹亞航空公司 (Olympic Airways, OA)開發一套包含前艙組員與後艙組員之排班系統。作者將限制規劃方法運用於此系統的勤務產生模組 (Duty Construction Module)、勤務組合產生模組 (Pairing Construction Module)與公平性組員指派模組 (Fair Crew Assignment Module)。

2000 年 Amal[5]提出一個結合限制規劃 (Constraint Programming)與線性規劃 (Linear Programming)於求解最小成本勤務組合產生問題的成功個案，其應用範圍為以天為基礎 (daily-based)的公車駕駛員排班模式，(Bus Driver Duty Optimization)。在過去，限制規劃方法與數學規劃方法兩者被視為是兩種求解組合最佳化問題的不同方法，並且被認為是互相競爭 (competing)的方法，而每種方法都有其優點與缺點。這篇文章善用每一種方法的優點並完美的結合這兩種方法於變數產生法 (Column Generation)的求解程序中，並以日本的一個公車公司為案例，成功的開發結合限制規劃與線性規劃的排班系統，而此

應用系統是運用 ILOG 公司發行的限制規劃程式庫(ILOG Solver)與數學規劃程式庫(ILOG CPLEX)共同開發完成的。

2000 年 Talys et al.[33]結合限制規劃與數學規劃來求解大規模的公車駕駛員之排班問題，個案公司為一家巴西的公車業者，在其測試例題中，最大的規模高達 1.26×10^7 ，作者運用此混合式演算法(Hybrid Approach)，在約 22 小時求得最佳解。此外，作者亦比較運用限制規劃方法與 Desrocheds and Soumis [13]所提的限制性最短路徑演算法(constrained shortest path)來求解子問題的差異，作者發現限制性最短路徑演算法對於求解大規模 k 條具有資源限制性的最短路徑(k-shortest path)問題時，其求解效率相當差且需要大量記憶體，其主要原因為，隨著問題的規模變大，子問題的路網會變的相當大，而由於限制性最短路徑演算法必須在路網中之每個節點上，設定各種標籤來記錄各種限制性資源(如：總飛行時間、TAFB、總休息時數等)在不同路徑上的使用狀態，所以將使得演算法的總運算時間中，相當多的時間花費在這些標籤的紀錄與更新上，經過作者對於大規模例題的測試，發現大約有 99.2% 以上的演算法運算時間花在每個網路節點點上之不同路徑、不同限制資源之標籤記錄上。因此，作者建議當運用變數產生法來求解空勤組員之排班問題時，運用限制規劃方法來求解主問題所需求之變數較運用網路模式之限制性最短路徑演算法來的有效率，且模式之限制式表達方式也較有彈性、較簡單、較容易表達。

2000 年 Niklas[30]指出目前歐洲主要航空公司(如：Luthansa、British Airways、Air France、KLM、Alitalia、SAS、Iberia)皆使用 Carmen System AB 所開發出來之「空勤組員排班系統」，並強調此排班系統之核心求解架構是結合作業研究方法(Operation Research, OR)與限制規劃方法(Constraint Programming, CP)各自的優點所發展的求解架構，並簡單介紹此套系統的求解架構與其是如何有效地結合 OR 與 CP 兩種方法於求解空勤組員排班問題。

表 2.5 以限制規劃模式為基礎之變數產生法文獻

| 作者 | 年代 | 問題 | 方法 |
|-----------------|------|--------|------------------|
| Halatsis et al. | 1996 | 空勤組員排班 | 以限制規劃模式為基礎之變數產生法 |
| Amal | 2000 | 公車排班 | |
| Talys et al. | 2000 | 公車排班 | |
| Niklas | 2000 | 空勤組員排班 | |

第三章 限制規劃文獻回顧

限制規劃是從電腦科學(Computer Science)中的人工智慧領域所發展出來，而數學規劃是從傳統數學理論基礎發展而來，雖然此兩種方法皆可運用於求解組合最佳化問題，但此兩種方法求解問題的方式卻大相逕庭。由於大多 OR 領域之相關人員對於 OR 方法已有相當程度之瞭解，但對於限制規劃此一方法論卻相當地陌生，因此，在本章節中，本研究將透過對一些限制規劃相關文獻之回顧，為 OR 領域之相關人員作一略性之介紹，以期未來有更多有關於結合限制規劃與數學規劃方法論之研究。

3.1 限制規劃簡介

Brailsford et al.[9]將限制規劃定義為「運用電腦程式發展模式化語言，讓使用者能*簡單地、輕易地*描述(state)限制滿足問題，並透過精緻的電腦演算法(sophisticated computer algorithm)有*系統地、有效率地*求解(solve)限制滿足問題」，其主要的目的乃在於「簡化理論與實務問題解決的間距，讓一般對於程式語言(如：C 語言)不熟的使用者，亦能很快的發展其問題模式並進行求解」。這些精緻的電腦演算法最早是在邏輯語言(如：Prolog)中設計，但是由於此種邏輯求解程序(logical solution procedure)非常沒有效率，並且也沒有善用限制式系統中限制式與限制式間的關係來加快求解效率，針對此一缺點，部分科學家便開始針對用來求解限制滿足問題之邏輯語言作大幅度的修正，於是便產生如 CHIP(Constraint Handling in PROLOG)這類的限制式邏輯語言(Constraint Logic Programming, CLP)。近年來，由於電腦科技的發達與企業對於運用限制規劃方法來求解實務問題(如：組員排班、課表排定、工廠排程等)的需求，已經開始有物件導向式(C++/Java)的限制規劃程式庫被發展出來，常見的有 ILOG Solver(C++)、ILOG Jsolver(Java)與 COSYTECH 的 CHIPv5(C++)。

基於限制規劃之定義，可明顯地發現，限制規劃為一種二層式(Two-Tier)的求解架構，其中一層為能用來描述限制滿足問題的限制式元件，另一層為用來求解限制滿足問題的程式求解元件。各元件的相關內容，茲詳述如下：[36]

1. **限制式元件(Constraint Component)**：限制式元件又稱為限制式模式化元件(Constraint Modeling Component)，其功能乃在於提供一套親和的模式化語言，讓使用者能*直覺地(intuitive)、輕易地*宣告決策變數(decision viables)與限制式。然而，由於實務問題之限制式的種類相當多(如 logical constraint、arithmetic constraint、cardinality constraint, all-different constraint, atleast constraint, atmost constraint 等)，所以對於任何實務問題可能使用之限制式，限制規劃中皆存在一個其對應的限制式元件以供使用者進行問題之模式化(Modeling)動作，因此，就問題的描述方便性而言，限制規劃較傳統之數學規劃更容易讓使用者所使用。
2. **程式求解元件(Programming Component)**：為了使限制規劃能有效率地求解限制滿足問題，程式求解元件中必須包含一個智慧型的限制式系統邏輯推理元件；相對地，為

了使限制規劃能有系統地來求解限制滿足問題，程式求解元件中必須包含一個系統化的空間搜尋元件。然而，藉由此二元件的相互充分配合，促使限制規劃的求解效率相當的不錯。有關各元件之內部細節，茲簡述如下：

a. 限制式系統邏輯推理元件(Reasoning Component of Constraint System)

使用精緻的限制式求解演算法(Sophisticated Constraint Solving Algorithms)推理由限制滿足問題中所有限制式所構成之限制式系統(Constraint System)，相關的演算法主要包含決策變數值域與限制式間的一致性檢驗技術(Consistency Checking Techniques)與限制式遺傳機制(Constraint Propagation)兩類。

b. 空間搜尋元件：(Tree-Search Component)

使用 Tree-Search 方法中最基本的演算法-換枝檢驗法(Backtracking, BT)作為搜尋演算法之核心，並配合限制式系統邏輯推理元件中之變數值域縮減演算法，形成各種比 BT 演算法更有效率的智慧型空間搜尋演算法(如 Forward Checking 與 Look-Ahead Checking 等)。在空間搜尋策略方面，限制規劃中的空間搜尋元件除了提供使用者最常用的 Depth-first、Best-First 等搜尋策略外，還允許使用者根據其對問題領域知識(Problem Domain Knowledge)之瞭解，自行定義使用者認為最佳、最有效率的搜尋方式(User-Defined Search Procedure)，因此，就求解彈性而言，限制規劃之求解彈性相當地高。

由於限制規劃如同數學規劃一樣，皆可用來求解組合最佳化問題，使得許多才剛剛接觸限制規劃的 OR 相關人員，常對限制規劃發生以下兩點誤解：[9]

誤解 1：「限制規劃是數學規劃的延伸」

雖然限制規劃與數學規劃皆可被使用來求解組合最佳化問題的方法，但是此兩種方法論之求解思維模式卻迥然不同。數學規劃方法是從數學理論的角度來求解組合最佳化問題，此方法論之重心乃在於「運用歸類組合最佳化問題的種類並研究各類型問題的特性，以提出相關之數學理論來求解各類型之組合最佳化問題」；相反地，限制規劃方法是從電腦語言與電腦演算法的角度來求解組合最佳化問題，其方法論之重心在於「運用電腦語言與電腦演算法，使電腦能瞭解並且能系統化地、有效率地求解組合最佳化問題」，是一種電腦軟體工程思維。因此，限制規劃不是數學規劃的延伸，而是電腦科學領域人員用來求解組合最佳化之一種方法。

誤解 2：「限制規劃不能求解最佳化的問題」

限制規劃除了能求解限制滿足問題之外，亦能求解組合最佳化問題，其運作方式為將最佳化問題視為一個連續型的 CSP 問題(A Sequence of CSP)，意即當運用限制規劃在求解一個最大(小)化問題時，限制規劃視此最大(小)化問題中的限制式系統為一個限制滿足問題並進行空間搜尋求解，當在搜尋空間中找到一個可行解 Z^* 時，便加入「目標值變數 Z^* (目標值變數 Z^*)」的目標限制式(Objective Constraint)至此限制滿足問題中並繼續搜尋下一可行解，如此依此類推，直至所有可行解空間皆搜尋完為止，而最後的

Z^* 值，即為此最大(小)化問題的最佳解。然而，雖然可運用限制規劃來求解組合最佳化問題，但於規模很大且可行解相當連續的組合最佳化問題而言，文獻上大多不建議使用者使用限制規劃來求解此類問題，因為對於此類型的問題，限制規劃較無法發其快速、有效地求解能力。

對於離散式的組合最佳化問題而言，目前有作業研究(OR)、限制規劃(CP)兩種方法可以處理，然而由於此兩種方法之運作方式不同，使得此兩種方法較適用的問題類型亦不同。基於限制規劃中演算法之運作特性，可知限制規劃之僅較適合限制程度高的組合最佳化問題(如：含時間窗限制的車輛排程問題、含容量限制的車輛排程問題等)與限制滿足問題(如：排課問題、班表設計問題、勤務組合產生問題、排程問題等)，在求解此類型問題時，限制規劃之求解效率通常較作業研究來的佳；然而，對於限制程度低的組合最佳化問題而言，由於作業研究有其紮實之數學理論基礎，所以在求解此類型問題時，作業研究之求解效率通常較限制規劃來的佳。根據此一分析，本研究茲將作業研究與數學規劃之適用問題類型整理成表 3.1 所示。

表 3.1 OR 與 CP 方法所適用之問題類型

| 問題特性 | 求解方法 | 可行解空間特性 |
|--|-------|---------------------------|
| 限制程度高的問題 (Fully-constrained Problem) | CP | 可行解空間為斷斷 續續的(fragment) |
| 限制程度中的問題 (median-constrained Problem) | CP+OR | 介於上、下兩者之間 |
| 限制程度低的問題 (less-constrained Problem) | OR | 可行解空間相當 大且非常連續 |

綜合歸納應用限制規劃方法來求解限制滿足問題之優越特性，可發現限制規劃包含下列三項優點點：[9]

1. **有效率之求解機制**：由於在限制規劃的求解機制中，充分地結合了一致性檢驗技術技術(Consistency Checking Techniques)、限制式遺傳機制(Constraint Propagation)與智慧型的搜尋演算法（如：Forward Checking Algorithm、Look-Ahead Checking Algorithm），使得限制規劃的求解效率相當高。
2. **模式建立之方便性**：由於限制規劃語言屬於程式語言中之「宣告式語言(Declarative Language)」，使得使用者可以很容易地運用限制規劃語言來建構其限制滿足問題模式。
3. **可處理多樣之限制式**：限制規劃可以處理各種實務問題之限制式類型而其中有許多是數學規劃所無法直接使用的，如：布林限制式(Boolean Constraints)、邏輯限制式(Logical Constraints)、序列限制式(Sequence Constraints)等，所以對於不熟悉數學規劃理論的使用者，仍然可很直覺地(Intuitively)、容易地(Easily)運用限制規劃來就構其限制滿足問題或組合最佳化問題。

3.2 限制滿足問題簡介

3.2.1 限制滿足問題之定義

所謂限制滿足問題 (Constraint Satisfaction Problem, CSP) 係指「在給定一組決策變數 $\{X_1, \dots, X_n\}$ 、各變數相對應之有限值域 $\{D_1, \dots, D_n\}$ 、一組限制式 $\{C_1, \dots, C_m\}$ 之條件下，尋找滿足各項限制式之一組或多組可行解」[9]，其中， n 表示 CSP 中的決策變數個數， m 表示 CSP 中的限制式個數，有限值域(Finite Domain)表示決策變數之值域是由整數所構成且值域中存在一個上限值、一個下限值。由於限制滿足問題主要是由變數集合 X 、變數對應之值域集合 D 與限制式集合 C 所構成，所以文獻上常以 (C, X, D) 來表示一個 CSP(FD)問題。

在問題的本質上，限制滿足問題不同於的最佳化 (Optimization) 問題，其求解目的乃在於尋得一組或多組 *可行解*，而非最佳化問題所要求之 *最佳解*。這類問題中，最著名的有地圖塗色問題(Map Coloring Problem)和 n -皇后問題(N -queens Problem)，然而許多實務問題之特性亦屬於限制滿足問題，例如：班表設計問題 (Timetabling Problem)、勤務組合產生問題(Crew Pairing Generation Problem)、組員派遣 (Crew Rostering Problem) 等問題。

回顧限制滿足問題的發展，在 1970 年代，人工智慧(Artificial Intelligence, AI)領域的電腦科學家開始投入大量之研究與心思在建構限制滿足理論之研究上，其中最具貢獻的學者莫過於 Waltz(1972)、Montanari(1974)與 Mackworth(1977)等人，不巧的是，由於此一時期的電腦運算能力相當差，使得當時之電腦科學家尚無法有效地運用限制滿足理論來求解各種實務問題，並因而停滯了限制滿足理論之相關研究。直至 1990 年代，由於電腦的運算能力越來越強，使得限制滿足理論再度受到電腦科學家之重視，因而促使停滯已久的限制滿足理論再度被廣泛地研究。發展至今，文獻上已經存在許多極具效率的電腦演算法可被用來求解限制滿足問題。在此一研究領域之發展過程中，學術界為了促進相關領域學者間的學術交流，因而在 1996 年開始發行的 *Constraints* 期刊，以期提供一個專屬的學術論壇來促使此一研究領域之更加地活絡並藉此期望為此一研究領域開闢另一個新紀元。[9]

3.2.2 限制滿足問題中之限制式種類

一般而言，用來構建一個限制滿足問題時最常使用之限制式類型，包含下列幾類：

1. **算術限制式(arithmetic constraint)**：表示傳統數學理論中之算術運算限制式，如 $X[1]+X[2]+X[3]+X[4]+X[5]$ ($<$ 、 $>$ 、 $=$ 、 \neq) d ，表示此五個變數值之加總小於等於(大於等於、小於、大於、等於、不等於) d 。若依據限制式中變數個數之多寡來分，可區分為：

| 限制式名稱 | 範例 |
|-----------------------------------|--|
| uniary constraints (一元限制式) | 如 $X \leq 4$ (1 元限制式) |
| binary constraints (二元限制式) | 如 $X \leq Y$ 或 $X + Y \leq 5$ (2 元限制式) |
| n-ary constraints (多元限制式) | 如 $X + Y + Z \leq 10$ (3 元限制式) |

2. **邏輯限制式(logical constraint)**：通常包含下列三種邏輯限制式

| 限制式名稱 | 含意 |
|--|--|
| 及限制式 (and constraint) | 表示兩個或兩個以上之限制式必須同時成立，如： $(X[i+1]=5 \wedge X[i+2]=3 \wedge X[i+3]=9)$ 表示這三個限制式必須同時存在 |
| 或限制式 (or constraint) | 表示兩個或兩個以上之限制式只存在一個即可，如： $(X[i+1]=1 \vee X[i+2]=1 \vee X[i+3]=1)$ 表示這三個限制式只要一個成立即可 |
| 條件限制式 (if-then-else constraint) | 表示當每個限制式成立時，則另外一個限制式也必須同時成立，如： $(X[i]=1 \rightarrow X[i+1]=1)$ 表示當限制式 $X[i]=1$ 成立時，則限制式 $X[i+1]=1$ 也必須同時成立 |

3. **cardinality constraint**：用來計算「某個限制式」成立的次數，如 $\text{sum}(i \text{ in } 1..m, j \text{ in } 1..n) \text{ cardinality}(X[i,j]=d)=k$ 表示變數陣列 X 中，限制式 $X[i,j]=d$ 成立之總數需為 k 個。
4. **all-different constraint**：表示某個變數陣列 $X[]$ 中，每個變數 $X[i]$ ($i=1 \dots n$) 被指派之值都必須不一樣，如： $\text{alldifferent}(X)$ 表示 X 變數陣列中每個變數都被指派一個不同的值或 $\text{alldifferent}(\{X, Y, Z\})$ 表示 X 、 Y 、 Z 三個變數之值必須完全不同。
5. **atleast constraint**：表示「某個限制式」成立的次數必須至少幾次以上，如 $\text{sum}(i \text{ in } 1..m, j \text{ in } 1..n) \text{ atleast}(X[i,j]=d)=k$ 表示變數陣列 X 中，限制式 $X[i,j]=d$ 成立之總數至少需為 k 個。
6. **atmost constraint**：表示「某個限制式」成立的次數必須至少幾次以上，如 $\text{sum}(i \text{ in } 1..m, j \text{ in } 1..n) \text{ atmost}(X[i,j]=d)=k$ 表示變數陣列 X 中，限制式 $X[i,j]=d$ 成立之總數至多需為 k 個。

3.3 限制規劃中之求解演算法簡介

限制規劃中的求解演算法主要是由一致性檢驗技術、系統化之空間搜尋演算法(Branch-and-Bound)所構成，其運作機制是在每個分枝節點中運用一致性檢驗技術來縮減變數之值域(Variables' Domain Reduction)，以減少可行解之搜尋空間，以期達到運用電腦演算法系統化且有效地求得限制滿足問題可行解之目標，亦即限制規劃中的求解演算法為一種智慧型的空間搜尋演算法(Intelligent Tree Search Algorithm)。以下即分別介紹目前限制規劃中最基本的一致性檢驗技術與目前限制規劃中最常使用之智慧型空間搜尋演算法。

3.3.1 一致性檢驗技術介紹

限制規劃理論中最基本的一致性檢驗技術包括：節點一致性(Node Consistency)、節線一致性(Arc Consistency)、界限一致性(Bounds Consistency)、一般化之一致性(Generalized Consistency)四種。每種一致性檢驗技術的複雜度皆不同，越簡單的一致性檢驗技術意味著其所需花費之運算時間越短，反之，越複雜的一致性檢驗技術意味著其所需要花費之運算時間越長，使得每一種一致性檢驗技術皆有其優點與適用時機。

一致性檢驗技術，可以在 Branch-and-Bound 的根節點(root node)與任何一個分枝節點上使用，至於要在節點上使用何種一致性檢驗技術，就要視限制滿足問題之特性與演算法之設計有關，但最本地就是，在每個節點上一定會執行節點一致性與節線一致性這兩種最基本、最簡單的一致性檢驗技術。為了使 OR 相關人員對一致性檢驗技術有一初步之認識，本研究即針對上述四種的一致性檢驗技術作一簡單地介紹：[27]

1. 節點一致性(Node Consistency)：

| 符號說明 |
|--|
| primitive constraint：單純由變數、數學基本運算子(+、-、 \times 、 \div 、=、 \neq 、 \leq 、 \geq 、<、>等)以及一些常數值所構成之限制式。 |
| 令 c ：表示一個 primitive constraint。 |
| 令 $\text{vars}(c)$ ：表示由 primitive constraint c 中之決策變數所形成之集合。 |
| 令 $ \text{vars}(c) $ ：表示決策變數集合中之變數個數。 |
| 令 $D(x)$ ：表示決策變數 x 所對應之值域集合。 |
| 令 $\min D(x)$ ：表示決策變數 x 之值域 $D(x)$ 中的最小值。 |
| 令 $\max D(x)$ ：表示決策變數 x 之值域 $D(x)$ 中的最大值。 |
| 令 D ：為一個 Super Set，表示一個 CSP 問題中，由每個決策變數 x 所對應值域集合 $D(x)$ 所形成之集合。 |

| 定義 1.1 |
|---|
| a primitive constraint c is node-consistent with domain D , if $ \text{vars}(c) = 1$ or if $\text{vars}(c) = \{x\}$, for all x : each d in $D(x)$, $\{x = d\}$ is a solution of c |

| 定義 1.2 |
|---|
| A CSP with constraint $C_1 \wedge \dots \wedge C_m$ is node-consistent with Domain D , if each constraint C_i is node-consistent with D for $1 \leq i \leq m$ |

| 範例 1 |
|--|
| A primitive constraint $C : z \leq 5$, $\text{vars}(c) : D(z) = \{2 \dots 10\}$ constraint c is node-consistent with $D(z) = \{2, 3, 4, 5\}$ |

2. 節線一致性(Arc Consistency) :

| 定義 2.1 |
|--|
| a primitive constraint c is arc-consistent with domain D , if $ \text{vars}(c) = 2$ or if $\text{vars}(c) = \{x, y\}$, for all x : each d_x in $D(x)$ and there is some d_y in $D(y)$, $\{x = d_x, y = d_y\}$ is a solution of c |

| 定義 2.2 |
|---|
| A CSP with constraint $C_1 \wedge \dots \wedge C_m$ is arc-consistent with Domain D , if each constraint C_i is arc-consistent with D for $1 \leq i \leq m$ |

| 範例 2 |
|---|
| A primitive constraint $C : x < y$, $\text{vars}(c) : D(x) = \{1, 2, 3\}$, $D(y) = \{1, 2, 3\}$ constraint c is node-consistent with $D(x) = \{1, 2\}$ and $D(y) = \{2, 3\}$ |

3. 界限一致性(Bounds Consistency) :

| 定義 3.1 |
|--|
| An arithmetic primitive constraint c is bounds-consistent with domain D , if for each $x \in \text{vars}(c)$: 1. each $d_i \in \min D(x_i) \dots d_i \dots \max D(x_i)$, $\{x = \min D(x) : x_1 = d_1, x_2 = d_2, \dots, x_n = d_n\}$ is a solution of c 2. each $d_i \in \min D(x_i) \dots d_i \dots \max D(x_i)$, $\{x = \max D(x) : x_1 = d_1, x_2 = d_2, \dots, x_n = d_n\}$ is a solution of c |

| 定義 3.2 |
|---|
| A CSP with constraint $C_1 \wedge \dots \wedge C_m$ is bounds-consistent with Domain D If each constraint C_i is bounds-consistent with D for $1 \leq i \leq m$ |

| 範例 3 |
|--|
| $X=Y+Z$, $D(X)=\{4..8\}$, $D(Y)=\{0..3\}$, $D(Z)=\{2..2\}$ |
| <p>1. 首先根據 $X=Y+Z$ 這個簡單的限制式，運用 Constraint Propagation 將此一限制式變為下列三個限制式 $X=Y+Z$, $Y=X-Z$, $Z=X-Y$</p> <p>2. 依據 Bounds-Consistency 原則與上述之 Constraint Propagation，我們可以得知以下六個關係式：</p> $\begin{array}{ll} X & \min D(Y) + \min D(Z), X & \max D(Y) + \max D(Z) \\ Y & \min D(X) - \max D(Z), Y & \max D(X) - \min D(Z) \\ Z & \min D(X) - \max D(Y), Z & \max D(X) - \min D(Y) \end{array}$ <p>3. 根據以上六個限制式，可得知與限制式 $X=Y+Z$ Bounds-consistent 之值域範圍：</p> $\begin{array}{lll} 2 & X & 5 \\ 2 & Y & 6 \\ 1 & Z & 8 \end{array}$ <p>4. 根據上述 Bounds-Consistency 的技術，可以更新每個決策變數之值域並與限制式 $X = Y + Z$ 是 Bounds-Consistent</p> $D(X)=\{4..5\}, D(Y)=\{2..3\}, D(Z)=\{2..2\}$ |

4. 一般化之一致性(Generalized Consistency)：

綜合歸納上述三種一致性檢驗技術，可發現『每一種 Consistency 技術，皆有其最適合使用之限制式類型。當限制式為一元限制式時，適用節點一致性檢驗技術；當限制式為二元限制式時，適用節線一致性檢驗技術；當 $n(n \geq 3)$ 元限制式時，適用界限一致性檢驗技術』。舉例而言，某一限制式為 $X^2 = 1 - Y^2 \wedge X \geq 0 \wedge Y \geq 0$ ，其決策變數所對應之值域為 $D(X)=D(Y)=\{-1,0,1\}$ ，首先分析此一限制式中，最多為二元限制式，最少為一元限制式，所以最適合此一限制式之一致性檢驗技術同時包含節點與節線一致性檢驗技術，因此，若不先經過限制式之特性分析而一味地運用界限一致性檢驗技術來縮減變數值域時，由於此限制式和與其決策變數的值域為 bounds consistent，所以決策變數之值域完全不會被縮減，但若同時運用節點與節線一致性檢驗技術來縮減變數值域，由於此限制式和它的值域不為 arc consistent，所以此限制式之決策變數值域將被縮減至空集合，亦即 $D(X)=D(Y)=\{\}$ 。

對於實務上的限制滿足問題，由於其包括的限制式種類可能同時出現一元、二元及多元限制式，所以，設計一個整合性的一般化一致性檢驗技術來有效地處理各種限制式，實為目前限制規劃領域中的一個重要課題。

3.3.2 空間搜尋演算法

為了讓限制規劃中的空間搜尋演算法能更有系統、更有效率，電腦科學家經過評估目前文獻上所提的 Generate-and-Test(GT)及 Backtracking(BT)二種系統化搜尋演算法(Systematic Search Method)與結合一致性檢驗技術之可行性，最後決定採行 Backtracking演算法作為限制規劃之空間搜尋演算法。然而，GT 與 BT 兩種空間搜尋演算法的運作方式各為何，為何限制規劃會採行 BT 作為其空間搜尋演算法之核心機制，經過以下之介紹相信 OR 相關人員便能清楚地瞭解電腦科學家之用意。

一、系統化的空間搜尋演算法

系統化的空間搜尋演算法乃指「能系統化地(Systematically)、完整地(Completely)搜尋問題的可行解空間」。目前文獻上最廣為人知的系統化搜尋演算法為 GT 與 BT 二種，其間之差異性可從兩個角度來分析，從變數指派之角度來看，GT 是獨立且同時地一次指派給每個決策變數一個可能值(generating value to each variable independently and simultaneously)，而 BT 是獨立但依序地指派給每個決策變數一個可能值(generating value to each variable independently but orderly)；從限制式之角度來看，GT 的限制式檢驗時機為當所有變數皆被指派一個可能值時，才進行限制式之檢驗，而 BT 的限制式檢驗時機為每當一個變數被指派一個可能值，即進行限制式檢驗，所以就空間搜尋之搜尋效率而言，BT 演算法之執行績效較 GT 演算法較佳。

1. Generate and Test(GT)：

GT 演算法為系統化搜尋演算法中最原始也最容易懂懂演算法，其演算法中包含兩個重要之元件，一個是 Generator(解產生器)和 Tester(解檢查器)，其運作方式為「由解產生器產生所有的可能解(Possible Solutions)，然後將其丟入解檢查器中檢查看看是否滿足問題的所有限制式」，運作流程如圖 3.1 所示。其中，GT 演算法中的解產生器完全沒有依據變數間的相關性(限制式)來產生可能解，使得 GT 的產生器又可稱為 Blind Generator(盲目的解產生器)。由於此演算法具有「因盲目地產生可能解而無法事先避免未來可能發生之衝突」之缺點，使得文獻上亦存在某些改良式之 GT 演算法。

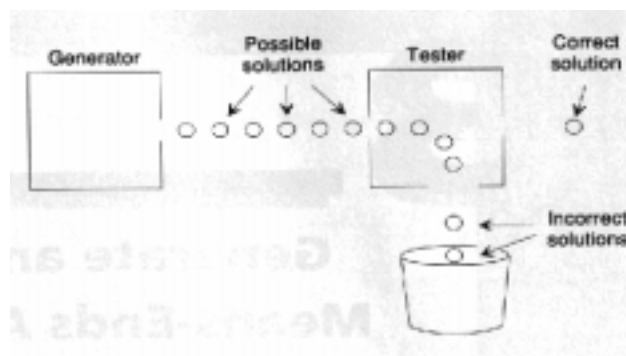


圖 3.1 GT 之運作方式

2. Backtracking(BT)：

換枝檢驗法(BT)為系統化搜尋演算法中最基本、最常使用之演算法，其演算法之運作概念為「以 Tree Search 為核心搜尋架構、以深度搜尋(Depth First Search, DFS)為搜尋策略、以初始化(initiate)決策變數(從決策變數所對應之值域中選擇一個值指派給此決策變數)作為分枝(Branching)之目的、以決策變數值域和限制式間之一致性檢驗與紀錄每個決策變數之目前值域狀態為每個分枝節點之必要工作、以遇到死巷節點(deadend node)為定限(Bounding)及換枝檢驗之準則」，其中由於每個分枝節點需負責記錄每個決策變數之目前值域狀態，所以分枝節點又可稱為一個狀態節點(State Node)，而在 BT 演算法中，所謂的決策變數值域和限制式間之一致性檢驗乃指「新初始化變數與已初始化變數間之一致性檢驗」；所謂的死巷節點乃指「在進行決策變數值域和限制式間之一致性檢驗時，若發生上述之不一致現象，即表示此時的分枝節點為一個死巷節點，而不能再進行分枝；反之，則為一個分枝節點，能再進行分枝」；所謂的換枝檢驗乃指「因遇到死巷節點而回溯至死巷節點之前一個分枝節點，並從此狀態節點選擇另一個分枝進行搜尋」。

在 BT 演算法之運作過程中，可依據「初始化」因素，將決策變數之類型分為已初始化變數(already initiated variables)、新初始化變數(newly initiated variable)/正初始化變數(currently initiated variable)、未初始化變數(not-yet initiated variables)三類，而所謂的已初始化變數乃指「在目前狀態節點之前，已被初始化之決策變數」，由於已初始化變數的值與問題的所有限制式皆具有一致性(Consistency)，所以由已初始化變數所形成的解集合可稱為「部分解(Partial Solution)」；所謂的新初始化變數乃指「在目前狀態節點上，所選擇要進行之分枝/初始化變數」；所謂的未初始化變數乃指「尚未進行任何初始化之變數」。

BT 演算法之運作流程為「根據 DFS 準則，依序地擴充(extend)一個部分解(partial solution)，直至找到一個可行解為止，若在擴充部分解的過程中遇到死巷節點，則換枝檢驗的動作立即發生並回溯至死巷節點之前一個分枝節點，以進行另一個分枝搜尋」，其中，所謂的擴充乃指「從未初始化變數中，選擇一個變數作為新初始化變數，並從此一新初始化變數的值域中，選擇一個值(Value)作為此變數的初始化值(Initiate Value)。

就演算法之執行效率而言，BT 演算法確實較 GT 演算法有效率，因為從 BT 演算法之運作流程中，可明顯地發現，當換枝檢驗發生時，即表示 BT 演算法刪除了一個不必要的搜尋空間，而此搜尋空間之大小即為所有未初始化變數值域的卡式積(Cartesian Product)。以運用 BT 演算法於求解 4-queens 問題為例，便可更清楚地瞭解之 BT 演算法之實際運作流程，其實際運作流程即如圖 3.2 所示：

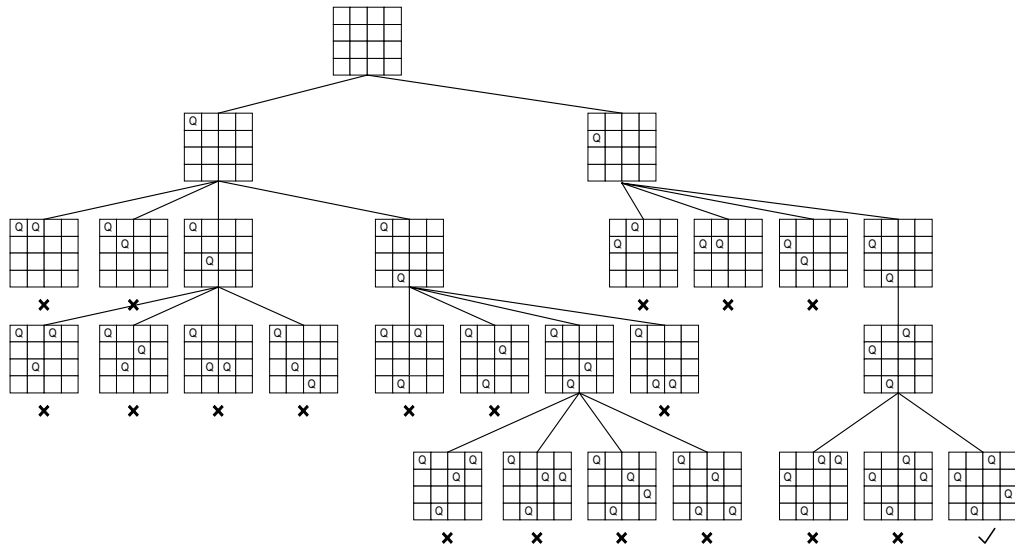


圖 3.2 BT 搜尋法-以 4-queens 為例

綜觀上述，可明確地發現，雖然 BT 演算法較 GT 演算法有效率，但由於 BT 演算法之運作方式只能透過死巷節點的發生而減少部分的搜尋空間，仍然無法事先避免未來會發生的衝突，尤其當我們使用 BT 演算法於求解複雜度高且問題規模大的組合最佳化或限制滿足問題時，便能更明顯地凸顯 BT 演算法的無效率性。為了改善 BT 演算法之缺點，電腦科學家則透過引進一致性檢驗技術於 BT 演算法中，構成一種新的智慧型空間搜尋演算法。

二、系統化的智慧型空間搜尋演算法

系統化的智慧型空間搜尋演算法乃指「能系統化地(Systematically)、完整地(Completely)、有效率地(Efficiently)搜尋問題的可行解空間」，最有名的有 Forward Checking (FC)與 Look-Ahead Checking(LC)兩種。理論上，BT、FC、LC 三種演算法之差異能以「演算法中，使用一致性檢驗技術於哪些變數類型之配對間」此原則來加以區分，根據此一原則並配合圖 3.3 即可充分顯示出此三種演算法之差異性，圖中明顯地顯示，BT 演算法使用一致性檢驗技術於「新初始化變數與已初始化變數間」，FC 使用一致性檢驗技術於「新初始化變數與未初始化變數間」，而 LC 演算法使用一致性檢驗技術於「新初始化變數與未初始化變數間、未初始化變數與未初始化變數間」。

在系統化的智慧型空間搜尋演算法(如：FC、LC 等)中，由於其最基本地一定有使用一致性檢驗技術於「新初始化變數與未初始化變數間」，使得在搜尋過程中之每個狀態節點中，未初始化變數之值域一定與已初始化變數之值域保持一致性(Consistency)，因此，每當從某個狀態節點進行分枝時，無須像 BT 演算法一樣要進行新初始化變數與已初始化變數間之一致性檢驗，此一優點除了可大幅降低智慧型空間搜尋演算法花費在一致性檢驗之執行時間而且還能使慧型空間搜尋演算法(如：FC、LC 等)透過事先避免未來會發生的衝突，來減少換枝檢驗次數、增加空間搜尋之效率，進而提高整體演算法之執行效率。

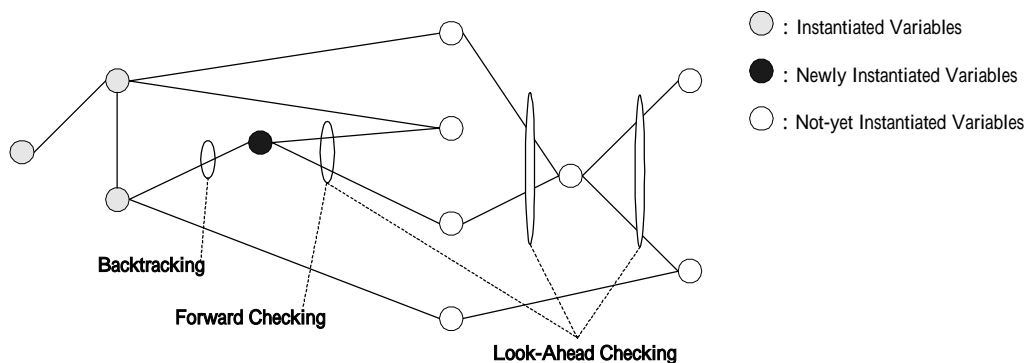


圖 3.3 BT、FC 與 LC 三種演算法之差異圖

資料來源：[10]

1. Forward Checking (FC)：

由 Haralick 與 Elliott 兩位學者於 1980 年所發展出來的 FC 演算法為智慧型空間搜尋演算法中最基本、最常使用之演算法，其運作概念與 BT 大致相同，差異性在於 FC 演算法是運用一致性檢驗技術於新初始化變數與未初始化變數間而不是運用在新初始化變數與已初始化變數間，藉此差異使得 FC 演算法能透過事先地刪除未初始化變數值域中與目前部分解具有不一致性的值域值(Domain Values)以避免目前部分解將面臨的未來衝突，進而大大縮減搜尋樹空間與搜尋過程中可能產生之換枝檢驗次數，大幅地提升傳統 BT 演算法之執行效率，成為一種智慧型空間搜尋演算法。同樣地，以運用 FC 演算法於求解 4-queens 問題為例，便可更清楚地瞭解之 FC 演算法之實際運作流程，其實際運作流程即如圖 3.4 所示：

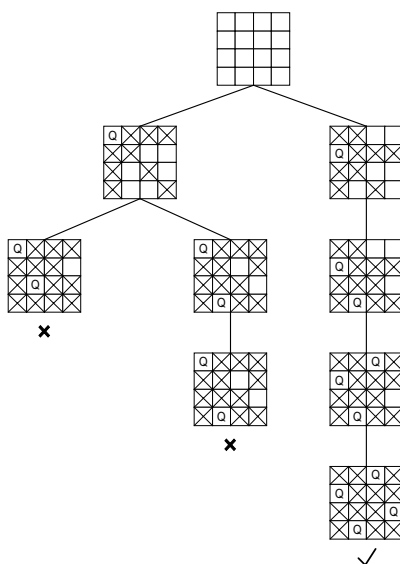


圖 3.4 FC 搜尋法—以 4-queens 為例

詳細比較 FC 與 BT 演算法之搜尋樹空間(searching tree space)(圖 3.3、圖 3.5)，可明顯地發現，FC 之分枝節點數(Branch Node)較 BT 少很多，且 FC 的分枝深度(Depth of Branch)亦較 BT 短，從此可得知，FC 演算法之執行效率較 BT 演算法好很多。

2. Look-Ahead Checking (LC) :

LC 演算法是一種比 FC 更智慧型(Intelligent)的演算法，其運作方式與 FC 皆相同，不同的是，LC 進一步地將一致性檢驗技術擴充運用至「未初始化變數與未初始化變數間」，使得 LC 較 FC 能更大幅度地縮減不必要的搜尋空間，然而此一額外之一致性檢驗使得 FC 在搜尋中的每個分枝節點上必須比 FC 花費更多的時間來執行一致性檢驗。

由 Sabin 與 Freuder 兩位學者於 1994 年所發展出來 MAC(Maintaining Arc Consistency)一致性檢驗技術為目前文獻上最常使用在未初始化變數與未初始化變數間之一致性檢驗技術，其運作方式乃「以節線一致性檢驗技術為基礎、以未初始化變數與其對應之值域所構成的子限制滿足問題(sub-CSP)為一致性檢驗之對象」，當 MAC 偵測到某個未初始化變數之所有值域與目前之 sub-CSP 呈現不一致(non-consistent)之現象時，即表示目前之 sub-CSP 不存在任何可行解，所以無須再從目前之分枝節點進行分枝。

以 4-queens 為例，運用 MAC 求解此問題之運作方式如圖 3.5 所示，其中有數字的格子表示「未初始化變數中與目前 sub-CSP 呈現不一致之值域值」，而數字的標示順序乃表示「當標示數字為 i 之未初始化變數值域值因與目前之 sub-CSP 不一致而被刪除時，便會進一步促使標示數字為 $i+1$ 之未初始化變數值域值與目前 sub-CSP 產生不一致之現象，其中 $i \geq 1$ 」，由此可知，標示數字為 1 的個格子表示，最開始與 sub-CSP 產生不一致之未初始化變數值域值，而標示數字為 2 的格子表示，當標示數字為 1 的值域值從其相對應之未初始化變數值域中移除之後，標示數字為 2 的未初始化變數值域值也必須因而從其對應之值域中移除，而其他的標示數字則依此類推。從圖 3.5 之左分枝來分析，由於經過 MAC 之一致性檢驗之後，發現最後一列之目前 queen 變數值域值完全與目前 sub-CSP 不一致，使得 LC 演算法能提早判定此一分枝完全不可能存在任何可行解而提止分枝和進行換枝檢驗，藉此減少空間搜尋之分枝數與提高整體空間搜尋演算法之執行效率。

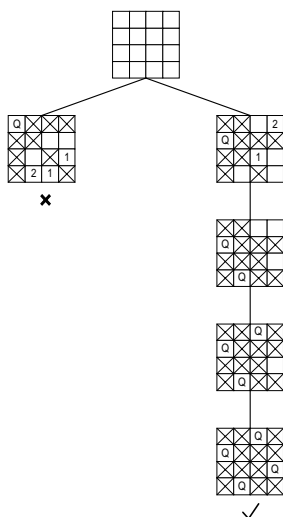


圖 3.5 MAC 搜尋法—以 4-queens 為例

3.3.2 空間搜尋演算法中之搜尋策略

基於空間搜尋演算法之運作方式，可明顯地發現「決策變數初始化順序(Variable Ordering)」與「決策變數值域值初始化順序(Value Ordering)」兩種搜尋策略對於整體空間演算法之執行效率有著重大之影響能力，然而，仔細觀察上述小節之空間搜尋演算法(如: BT、FC、LC)，可發現各演算法對於問題決策變數(Variable)及決策變數值域值(Value)之初始化順序準則並無任何之規定與限制，其主要原因為「對於所有之限制滿足問題，並沒有存在任何的決策變數及決策變數值域值之初始化順序準則可保證其搜尋效率是最好的」，雖然沒有最佳之初始化順序準則，但是文獻上還是提供一些基本的初始化順序準則可供參考，並且文獻上亦明確地指出藉由運用這些基本原則於實務問題之求解，通常可大幅地降低運用限制規劃求解實務問題之執行時間，因此，目前市面上之各種限制規劃軟體皆一定內建許多含式庫可供使用者能根據其本身問題之特性自行定義其問題特定之空間搜尋策略(User-Defined / Problem-Specific Search Strategy)，以期大幅地降低求解時間，提高模式運用之實用性。以下即分別進行簡述文獻上所提供之一般化初始化順序準則，其內容各分述如下：[10]

1. 決策變數初始化順序 (Variable Ordering)：分為靜態(Static)與動態(Dynamic)兩種

對於運用空間搜尋演算法求解限制滿足問題或組合最佳化問題而言，擁有正確而有效的決策變數值域值初始化順序準則之主要目的乃在於「期望在空間搜尋之過程中，盡量減少搜尋樹之分枝數並提升搜尋可行解之執行效率」。一般而言，決策變數值域值初始化順序亦分為靜態(Static)與動態(Dynamic)兩種，其內容各自分述如下：

a. 靜態的決策變數初始化順序：

又稱為確定性(Deterministic)之決策變數初始化順序，意指「在正式開始進行空間搜尋演算法之前，決策變數之初始化順序即已預先設定好，並在實際進行空間搜尋求解時，即完全依照此一既定之變數初始化順序進行變數的分枝」，由於此種決策變數初始化順序準則非常呆板、不聰明，所以此種變數初始化準則亦只有在無任何充分資訊下之 BT 演算法被採用而已。

b. 動態的決策變數初始化順序：

又稱為非確定性(Non-Deterministic)之決策變數初始化順序，此種搜尋策略乃意指「在空間搜尋過程中之每個分枝節點上，依據某準則動態決定要從未初始化變數之中選取哪個變數出來作為新初始化變數」，然而不管是遵循何種準則，基本上皆是根據搜尋樹之目前狀態而定(current state of search tree)，綜觀目前文獻上所提之相關準則，大可分為下列兩種：

準則 1：選取目前狀態中，值域值個數最少的未初始化變數作為新初始化變數

即為 Haralick 與 Elliott 兩位學者於 1980 年所提出之最先失敗原則(fail-first principle)[22]，雖然此原則之名字有點違反人類之正常思維，但其主要之觀念乃在於「若

是目前之部分解完全沒有任何機會可擴充成一個完整的可行解，則及早發現即可避免花費多餘的分枝搜尋時間，所以若選取目前之狀態中取擇值域值個數最少的未初始化變數作為新初始化變數，即可提早判斷目前之部分解是否有機會擴充成一個完整解，若是無機會則可減少許多不必要之分枝數，若是有機會則由於每個未初始化變數皆一定要被初始化才能獲得一個完整解，所以若是選擇值域值個數最少的未初始化變數作為新初始化變數，則其分枝數可能較選擇值域值個數較多的未初始化變數作為新初始化變數還來的少，對整體之空間搜尋效率而言，較有可能有直接之幫助。」。

準則 2：選取目前狀態中，限制程度最高的未初始化變數作為新初始化變數

所謂限制程度最高的未初始化變數(most constrained variable)乃指「與最多個未初始化變數有直接限制式關係的未初始化變數」，此原則之主要概念乃在於「由於限制程度最高的未初始化變數是在所有未初始化變數之中與最多其他未初始化變數有直接限制式關係之變數，所以當選取此值作為新初始化變數之標的時，可以保證此初始化動作將可觸動(Trigger)更大範圍之值域空間縮減(Domain Reduction)，使得空間搜尋之分枝數大幅減少、增加空間搜尋效率」。然而，使用此準則時必須有一個先決條件，才較能發揮此一準則之策略功效，此一條件為「此一限制程度最高的未初始化變數與限制式系統中最難滿足之限制式(可能一個或多個)，必須要有直接之限制式關係」，因為當限制程度最高的未初始化變數與限制式系統中最難滿足之限制式沒有存在任何之直接關係時，此一初始化原則可能導致無解之機率將會很大。

綜觀上述二種動態之決策變數初始化順序原則，由於使用原則 2 必須要符合其先決條件才能發揮其較大之求解功效，所以文獻上通常是以原則 1 為主，以原則 2 作為當同時含有(含)兩個以上值域值個數最少之未初始化變數變數時之解結原則(Tie-Breaking Rule)。此外，根據上述二種原則之策略概念，可發現動態的決策變數初始化順序並不是可以運用於所有的空間搜尋演算法中，例如最簡單的 BT 演算法，就不能適用，最主要的原因乃在於 BT 演算法無法提供充分的資訊作為決策變數初始化順序之原則依據。

雖然對於 BT 演算法而言，無法使用準則 1 與準則 2 來改善其空間搜尋效率，但文獻上仍存在一個簡單的準則可供 BT 演算法採用，此準則為「選取目前狀態中，與最多個已初始化變數有直接限制式關係的未初始化變數作為新初始化變數」，例如在地圖圖色問題(Graph Coloring Problem)中，從地圖之所有尚未著色之區塊中選擇一個相連最多個已著完色之區塊作為下一個著色之目標區塊，這樣便能提早發現無解之情況。

2. 決策變數值域值初始化順序(Value Ordering)：

對於運用空間搜尋演算法求解限制滿足問題而言，擁有正確而有效的決策變數值域值初始化順序準則之主要目的乃在於「期望在空間搜尋之過程中，盡量避免遇到死巷(deadend)之次數，甚至是希望能免回溯換枝(backtrack-free)就能找到一個可行解」。然而，對於運用空間搜尋演算法求解組合最佳化問題或無任何解之限制滿足問題而言，由於必須要將整個解空間完整搜尋過才能找到最佳解或確定此問題無解，在此種情況下，

不管使用任何之決策變數值域值初始化順序準則，對於整體搜尋效率皆無任何重大影響。一般而言，決策變數值域值初始化順序亦分為靜態(Static)與動態(Dynamic)兩種，其內容各自分述如下：

a.靜態的決策變數值域值初始化順序：

又稱為確定性(Deterministic)之決策變數值域值初始化順序，意指「在正式開始進行空間搜尋演算法之前，每個決策變數之直域值初始化順序即已預先設定好，並在實際進行空間搜尋求解時，即完全依照此一既定之變數值域值初始化順序進行變數值域值的之初始化」，由於此種決策變數值域值初始化順序準則非常呆板、不聰明，所以此種變數值域值初始化順序準則亦只有在無任何充分資訊下之 BT 演算法被採用而已。

b.動態的決策變數值域值初始化順序：

又稱為非確定性(Non-Deterministic)之決策變數值域值初始化順序，此搜尋策略乃意指「在空間搜尋過程中之每個分枝節點上，依據某準則來動態決定要從新初始化變數之值域中選取哪個值域值來指派給此新初始化變數」。然而，目前文獻上皆以「最先成功準則(succeed-first principle)」作為 Dynamic Value Ordering 之最高指導原則，其意指「從新初始化變數之目前值域中，依據使用者自訂之最先成功準則來選擇最有可能成功找到可行解的值域值並將其指派給此新初始化變數」，由於目前並沒有一般化的準則來判斷哪個值會先求到可行解，因此，大部分的作法則是根據問題的特性而定來設計此種空間搜尋策略。

綜觀上述兩種空間搜尋策略，可明顯地發現，對於一個好的空間搜尋演算法而言，除了必須包含一些智慧型之空間搜尋技巧之外，還必須兼具好的 Variable Ordering 與 Value Ordering 之搜尋策略加以輔助，方能發揮整體空間搜尋演算法之最大執行績效。

3.4 限制規劃之整體求解機制

限制規劃之整體求解機制為「根據 Variable Ordering 策略，依序決定在每個狀態節點上之新初始化變數，並同時依據 Value Ordering 策略，從新初始化變數之目前值域中選取一個值域值指派給此新初始化變數，然後根據一致性檢驗技術(Consistency Checking Techniques)或限制式遺傳演算法(Constraint Propagation)來執行值域縮減(Domain Reduction)之動作，接著在目前狀態中檢查是否存在一個或一個以上之變數值域成為空集合，若存在空集合之值域狀態則表示此指派為不可行之指派(unfeasible assignment)，並重新回溯至前一狀態，重新進行指派，若不存在空集合之值域狀態，則表示此指派為可行之指派(feasible assignment)；然後重複執行此步驟直至所有變數都已初始化為止」，其運作流程如圖 3.6 所示。

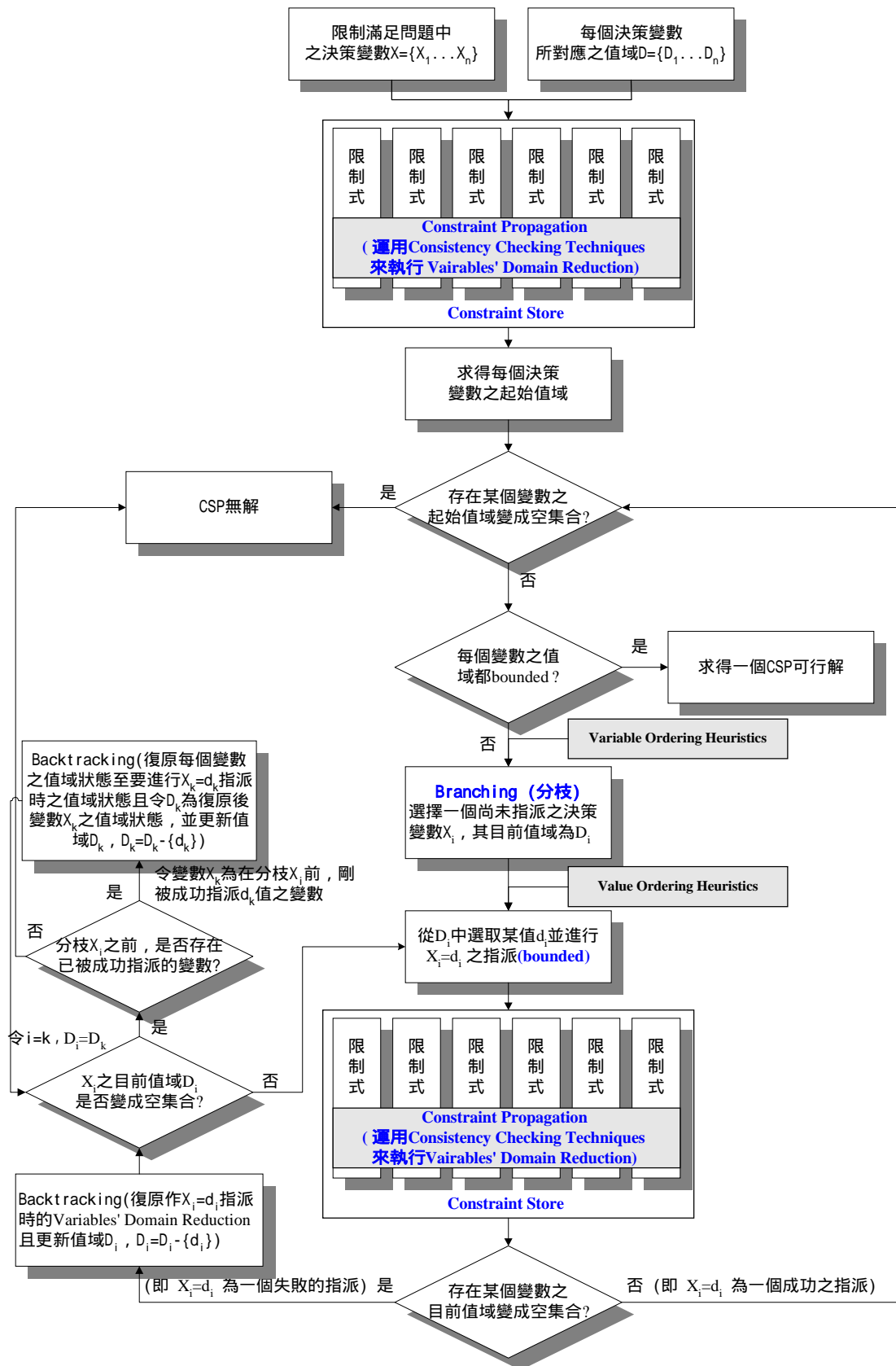


圖 3.6 限制規劃之整體求解機制

3.5 構建限制規劃模式所需考慮之因素

運用限制規劃方法於求解限制滿足問題時，首先必須面對的第一個問題就是「要如何建構限制滿足問題模式」，亦即「要如何表達限制滿足問題的變數與限制式」，雖然有很多種變數與限制式之表達方式可以構建出同樣的限制滿足問題，但是由於不同之變數表達方式將進一步會影響限制式之表達方式，而不同之限制式表示方式亦將進一步影響一致性檢驗技術所能發揮之執行績效。以下茲就一致性檢驗技術之特性，歸納以下三點構建限制規劃模式時所需注意的事項：[9]

注意 1：盡量使用二元限制式而避免使用多元限制式來構建限制規劃模式

由於當一個限制式中所包含之變數個數越多(較多元之限制式)，一致性檢驗技術越無法對於其值域空間進行有效之縮減，然而綜觀所有之一致性檢驗技術，其中最基本亦最有效率之一致性檢驗技術為節線一致性檢驗技術，且因為節線一致性檢驗技術最適合運用於二元限制式中，所以如果能盡可能地將一個限制滿足問題模式化成一個具有較多個二元或較低元限制式之限制規劃模式，便可充分地提高限制規劃求解此限制滿足問題之求解績效。以下將以 Cryptarithmic Puzzles 問題為例，進行說明與分析：

| 範例：Cryptarithmic Puzzles 問題 | |
|---|--|
| <p>問題描述：</p> <p>10 個英文字母 D、O、N、A、L、G、E、R、B、T 分別代表值域值為{0,...,9}之變數，其限制式包含下列三項：</p> <p>限制式 1：alldifferent(D,O,N,A,L,G,E,R,B,T)</p> <p>限制式 2：D ≠ 0 & G ≠ 0 & R ≠ 0</p> <p>限制式 3：滿足下列的編碼運算</p> $ \begin{array}{r} \text{D O N A L D} \\ + \text{G E R A L D} \\ \hline = \text{R O B E R T} \end{array} $ | |
| <p>限制規劃模式 1：使用較多元之限制式型態來建構限制式 3</p> $ \begin{aligned} &100,000 * D + 10,000 * O + 1000 * N + 100 * A + 10 * L + D \\ &+ 100,000 * G + 10,000 * E + 1000 * R + 100 * A + 10 * L + D \\ &= 100,000 * R + 10,000 * O + 1000 * B + 100 * E + 10 * R + T \end{aligned} $ | |
| <p>限制規劃模式 2：使用較低元之限制式型態來建構限制式 3</p> <p>新增五個值域為{0,1}之進位變數 C1、C2、C3、C4、C5</p> $ \begin{aligned} 2 * D &= 10 * c1 + T \\ 2 * L + c1 &= 10 * C2 + R \\ 2 * A + C2 &= 10 * C3 + E \\ N + R + C3 &= 10 * C4 + B \end{aligned} $ | |

$$O + E + C4 = 10 * C5 + O$$

$$D + G + C5 = R$$

$$D \geq 0, G \geq 0, R \geq 0$$

限制規劃模式分析：

限制規劃模式 2 使用較低元限制式型態所建構出來之限制式 3，包含 6 個子限制式且其中之子限制式包含最多個變數的為 5 個，而在限制規劃模式 1 中，直接使用 10 元限制式表示限制式 3，雖然此兩種限制規劃模式皆可用來表示同一個 Cryptarithmic Puzzles 問題，但是由於限制規劃模式 2 包含較多較低元之限制式，所以就求解效率而言，限制規劃方法對於限制規劃模式 2 之求解效率較限制規劃模式 1 有效率。

注意 2：盡量使用多餘限制式(redundant constraint)。

所謂的多餘限制式乃意指「不管此限制式是否存在模式之中，都不會影響整個模式的解」，然而，使用多餘限制式模式化(redundant modeling)技巧之主要目的乃在於「藉由在限制式系統中增加一個或多個有意義之多餘限制式(redundant constraint)，以期增加限制式遺傳演算法(Constraint Propagation)之功效並提高限制規劃之求解績效」。

注意 3：盡量使用少變數多值域(few variables with large domain)而避免多變數少值域(many variables with small domain)之變數定義方式，且盡量避免使用大量的二元變數(binary variable)。

假設在一個限制規劃模式中，有 n 個變數且每個變數的相對值域值恰好皆為 m 個，則其搜尋空間大小為 m^n 。當 $n < m$ 時，表示此模式為一個少變數多值域的模式；相對的，當 $n > m$ 時，表示此模式為一個多變數少值域的模式，相較此兩種模式的搜尋空間大小，可以很清楚的發現，少變數多值域模式的搜尋空間較多變數少值域模式來的小，尤其當 n 大於 m 的程度越高時，兩模式搜尋空間大小的差值也就越大。由此可知，對於構建任何一個限制滿足問題之限制規劃模式時，若能將其相對應之限制規劃模式構建成一個少變數多值域模式時，便能大幅地縮減模式求解時所需花費之時間。以下將以 n -queens 問題為例，進行說明與分析：

範例：n-queens 問題

問題描述：

假設現有一 $N \times N$ 之方形棋盤，上面總共須要擺 N 個棋子，而擺棋之規則為：

規則(限制式)1：同一行不能出現兩顆棋子

規則(限制式)2：同一列不能出現兩顆棋子

規則(限制式)3：對角線不能出現兩顆棋子

規則(限制式)4：\對角線不能出現兩顆棋子

限制規劃模式 1：多變數少值域之模式建構方式

決策變數定義：

$$X_{ij} = \begin{cases} 1, & \text{if there is a queen in row } i \text{ and column } j \\ 0, & \text{otherwise} \end{cases}, \forall i \in 1 \sim n, \forall j \in 1 \sim n$$

限制式表示：

$$\text{限制式 1 : } \sum_{i=1}^n X_{ij} = 1, j = 1 \sim n$$

$$\text{限制式 2 : } \sum_{j=1}^n X_{ij} = 1, i = 1 \sim n$$

$$\text{限制式 3 : } \sum_{(i,j) \in S} X_{ij} = 1, S = \{i+j \mid i+j = 2 \sim 2n\}$$

$$\text{限制式 4 : } \sum_{(i,j) \in S} X_{ij} = 1, S = \{i-j \mid i-j = n-1 \sim 1-n\}$$

搜尋空間大小： 2^{n^2}

限制規劃模式 2：多變數少值域之模式建構方式

決策變數定義：

queen[i], if there is a queen in row i, column queen[i],

$$\forall i = 1 \sim n \text{ and queen}[i] = 1 \sim n$$

$$\text{限制式 1 : } \text{queens}[i] \neq \text{queens}[j], \forall i \neq j$$

$$\text{限制式 2 : } \text{queen}[i] \text{ and queen}[i] = 1 \sim n, \forall i = 1 \sim n \text{ (每個 row 只能對應一個 column)}$$

$$\text{限制式 3 : } \text{queens}[i] + i \neq \text{queens}[j] + j, \forall i \neq j$$

$$\text{限制式 4 : } \text{queens}[i] - i \neq \text{queens}[j] - j, \forall i \neq j$$

搜尋空間大小： n^n

限制規劃模式分析：

就變數之定義方式而言，限制規劃模式 1 屬多變數少值域之變數定義方式(n^2 個變數、每個變數之值域皆為{0,1})，而限制規劃模式 2 屬少變數多值域之變數定義方式(n個變數、每個變數之值域皆為{1,...,n}個)；就模式所使用之限制式類型而言，限制規劃模式 1 皆用多元限制式，限制規劃模式 1 皆用二元限制式；就模式之搜尋空間大小而言，限制規劃模式 1 之搜尋空間為 2^{n^2} ，限制規劃模式 1 之搜尋空間為 n^n 。綜觀上述，可明顯發現，變數之定義方式，深深地影響模式所使用之限制式類型與搜尋空間之大小，進而影響至整個空間搜尋之求解效率。

注意 4：儘量減少限制規劃模式中，存有任何對稱性解(Symmetric Solution)之結構

對於許多實務問題而言，其問題結構中存有對稱性解的特性，而所謂的對稱性解乃意指「當某一組解為可行解，則存在另外一組或多組結構類似的可行解；相對的，當某一組解為不可行，則不存在另外一組或多組結構類似的不可行解」，當一個問題存有對稱性解特性之問題結構時，將有下列二個缺點，缺點一為「使問題之解空間因為存在許多對稱性解而擴大整個問題之實際複雜度與實際解空間」，缺點二「在空間搜尋之過程中，將浪費許多時間在搜尋類似的對稱性解」，所以，若能在構建限制規劃模式時即刪

除此種問題之對稱性解結構，即可大量縮小解空間並加快搜尋非對稱性可行解之速度。

詳細觀察各類型之實務問題，可明顯地發現，班表排定問題(Timetable Problem)與人員派遣問題(Rostering Problem)這兩類問題，具有對稱性解特性之程度最高，因為，在班表排定問題中，我們可以從某一可行解中，交換某些資源(Resource)或活動(Activity)得到另外一組或多組解，而在人員派遣問題中，我們可以將具有相同能力技術(Skill)的人員先群組起來，並相互交換其所派遣的任務得到另外一組或多組可行解。因此，當我們在建構班表排定問題之限制規劃模式或人員派遣問題之限制規劃模式時，我們必須要儘可能地消除此種對稱性解結構，以便減少問題之搜尋空間與求解時間。

3.6 模式化語言 OPL 之簡介

組合最佳化問題在現實世界的問題中到處存在，舉凡規劃(Planning)、排班排程(Scheduling)、排序(Sequencing)、資源規劃(Resource Allocation)等都是這類型的問題，而綜觀目前一些較強而穩健的求解系統(Robust Solver)雖然都能對大規模的線性規劃與各類型整數規劃問題提供效能不錯的求解演算法，但由於實際的實務問題規模非常大也非常複雜，往往使得 OR 人員在求解實務問題時，花很多時間在模式化問題(Modeling Problem)並且轉換成這種求解系統可以讀的檔案格式與設計問題的求解演算法，這意味著「模式構建(Model Formulation)」與「求解演算法」之間的間距相差非常大，因此，對於 OR 人員而言，若能擁有一套緊密連接「模式構建」與「求解演算法」的模式化語言，將促使 OR 人員更快速地解決企業所面臨之實務問題，並大幅地提高 OR 人員的生產力。

OPL(Optimization Programming Language)為一種專門求解組合最佳化問題的模式化語言(Modeling Language)，不同於一般命令式語言(Imperative Language)(如：C、Basic 等)，OPL 為一種宣告式語言(Declarative Language)，由於繼承宣告式語言的特性，使得限制式在模式中的順序並不會影響模式之求解結果，所以使用者可以隨時地加入、修改、刪除與管理問題之限制式。模式化語言最大的貢獻在於，它拉近了『電腦程式語法』和『數學表示法』之距離，讓使用者能簡便地描述並解決問題。

就*模式化語言特性*之角度來看，OPL 乃是從 AMPL 和 GAMS 這兩種支援傳統代數表示法的模式化語言延伸而來，它除了提供如 AMPL 和 GAMS 兩種模式化語言對於數學規劃模式的支援，還支援限制規劃模式，可以大量縮減 OR 人員的模式建構時間；就*求解效率*之角度來看，OPL 直接與目前求解效率最佳(state-of-the-art)之線性規劃、整數規劃、限制規劃演算法進行連接，可以大量縮減 OR 人員的模式求解時間。

OPL 與傳統模式化語言(如：AMPL、GAMS)不同的功能可歸納成下列三項：

1. 支援限制規劃模式：支援限制規劃(Constraint Programming, CP)模式是 OPL 與傳統模式化語言最大的不同處。
2. 使用者可以自訂搜尋演算法(Search Procedure)：OPL 允許使用者針對其面臨的組合最佳化問題(如：線性規劃問題、整數規劃問題、限制滿足問題)特性，自行訂定搜尋演算法；而傳統的模式化語言並不提供這個功能。
3. 提供更多種的限制式表達方式：針對我們實務問題最常見的排班排程問題(Scheduling Problem)與資源配置問題(Resource Allocation)，OPL 提供專門的語法來表示活動(Activity)與資源(Resource)之間的關係，使得使用者模式這類問題時，能更容易，此外，OPL 提供比傳統模式化語言更多的限制式表達方式，如：Higher-Order Constraint、Logical Constraint 等。

第四章 一般化勤務組合產生問題限制規劃模式

4.1 勤務組合產生問題

4.1.1 問題描述

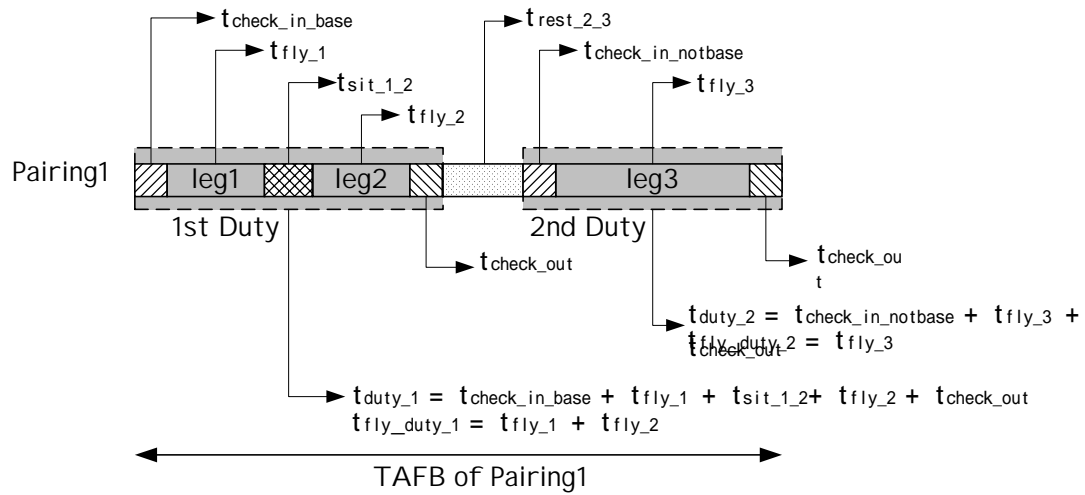
勤務組合產生問題之基本定義為「在滿足種種複雜排班法規限制下，產生多組可行勤務組合，其中每個可行勤務組合是由一組連續的勤務所構成，每個勤務是由一組連續的航段所組成，且勤務組合之起點與迄點城市必須為同一個航空公司組員基地」[34]。此問題之相關名詞定義如表 4.1 所示：

表 4.1 勤務組合產生問題之相關名詞定義

| 名詞 | 定義 |
|--|---|
| 航段(flight leg) | 一個直飛(nonstop)的航班 |
| 勤務(duty) | 空勤組員之最小工作單位，即為組員能在一天內合法完成值勤的任務工作，其種類有二，一種為「只包含單一航段(通常為長程航段)之勤務」，另一種為「包含一組短程航段接續之勤務」，其中任兩連續航段間的休憩時間必須介於法定之最短與最長修休憩時間之間 |
| 勤務組合(pairing) | 由一組連續的勤務段所構成，其中任兩連續勤務段間相隔一段較長的休息時間(如組員外宿過夜時間)，而且勤務組合之起點與迄點城市必須要在相同的航空公司組員基地(HomeBase) |
| 報到時間 (brief time/check-in time) | 從組員報到至開始值勤之時間，通常又可分為 <i>基地報到時間</i> 與 <i>外站報到時間</i> |
| 報離時間 (debrief time/check-out time) | 從組員結束值勤至正式報離之時間 |
| 總報到時間 (total brief time/check-in time) | 勤務組合中，所有報到時間之總和(由於一個勤務組合之中可能包含很多個勤務，所以其可能包含很多次之組員報到次數) |
| 總報離時間 (total debrief time/check-out time) | 勤務組合中，所有報到時間之總和(由於一個勤務組合之中可能包含很多個勤務，所以其可能包含很多次之組員報離次數) |
| 休息時間(rest times) | 又稱為勤務間之休息時間，此休息時間一定較休憩時間長，意指組員在勤務終止報離後至下一勤務開始報到時，中間之間隔時間，其間一定包含組員在旅館住宿之過夜時間(overnight rest times) |
| 休憩時間(sit times) | 包含兩個以上航段之勤務中，兩連續航段間的休憩時 |

| | |
|--|--|
| | 間 |
| 最短休憩時間 (minimum sit times) | 組員值勤兩連續航段時，其間最少需有的休憩時間，此休憩時間能提供組員短暫休息時間與下一航段之飛航準備時間 |
| 最長休憩時間 (maximum sit times) | 最長的休憩時間，航空公司不能讓組員在機場休憩或等候下一執勤航段太久，若是超過此一最長休憩時間，航空公司必須負責送組員到旅館休息 |
| 勤務組合之總休息時間 (total rest times in pairing) | 勤務組合中，所有休息時間之加總 |
| 勤務之總休憩時間 (total sit times in pairing) | 勤務中，所有休憩時間之加總 |
| 勤務之總飛行時間 (total flying time in duty) | 勤務中，所有航段飛行時間之加總 |
| 勤務之總歷時時間 (elapsed time of duty) | 勤務之組員報到與報離時間、勤務之總飛行時間、勤務之總休憩時間之時間加總 |
| 勤務組合之總飛行時間 (total flying time in pairing) | 勤務組合中，所有勤務總飛行時間之加總 |
| 勤務組合之差旅時間 (Time Away From Base, TAFB) | 勤務組合所花費之總時間，為勤務組合中，總報到時間與總報離時間、總休息時間與總休憩時間、總飛行時間之加總 |
| 空載(deadhead) | 為一個空勤組員不需執勤的航段，空勤組員在此航段中為一位乘客不需值勤，其主要目的是為了要運送一位空勤組員至另外一個城市，使其服務下一個航段 |

圖 4.1 為一可行勤務組合之相關名詞介紹示意圖。圖中橫軸代表時間，可行勤務組合 Pairing1 包含勤務 1、2 兩個勤務，且此兩勤務皆符合飛時、工時、休時、航段接續、組員報到報離時間等相關排班法規之限制。其中勤務 1 包含航段 1、2 兩短程航段，勤務 2 只包含航段 3 一個長程航段，勤務 1 之工作時間為基地報到時間、報離時間、航段 1 之飛行時間、航段 2 之飛行時間、航段 1 與航段 2 間之休憩時間之總和，勤務 1 之飛行時間為航段 1 與航段 2 之飛行時間總和；勤務 2 之工作時間為外站報到時間、報離時間、航段 3 之飛行時間之總和，勤務 2 之飛行時間為航段 3 之飛行時間。勤務組合 Pairing1 之差旅時間 TAFB 為勤務 1 與勤務 2 工作時間之總和。



$t_{check_in_base}$: 基地報到時間
 $t_{check_in_notbase}$: 外站報到時間
 t_{check_out} : 外站報到時間
 t_{sit_min} : 法定最短休憩時間
 t_{sit_max} : 法定最長休憩時間
 $t_{sit_i_j}$: 航段i與航段j間之休憩時間，介於 t_{sit_min} 與 t_{sit_max} 之間
 $t_{rest_i_j}$: 航段i與航段j間之休息時間
 t_{fly_i} : 航段i之飛行時間
 t_{duty_i} : 勤務i之總歷時時間
 $t_{fly_duty_i}$: 勤務i之總飛行時間

圖 4.1 可行勤務組合之相關名詞介紹示意圖

4.1.2 勤務型態

本研究所定義之勤務型態乃是根據「勤務中所包含之航段數 X 」來區分，並以 $dutyX$ 符號表示其所代表之勤務類型，其中，當 $X=0$ 時表示此勤務為虛擬勤務。一般而言，在一個勤務組合中，可能包含之勤務類型數等於排班法規對於勤務最多能包含之航段數 N_d 加上一個只包含虛擬航段之虛擬勤務，舉例而言，當 $N_d = 4$ 時，表示勤務組合中可能發生之勤務類型共有 5 種，細節如下圖 4.2 所示。

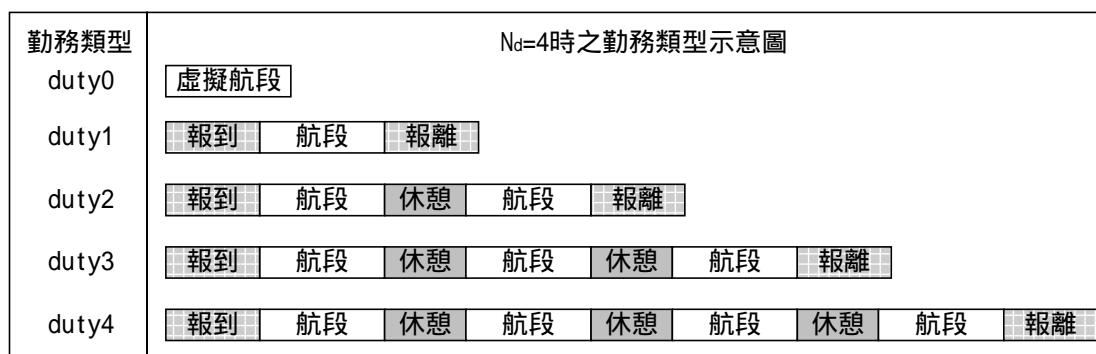


圖 4.2 $N_d = 4$ 時之勤務類型

在國際航線之空勤組員排班問題中，由於每個航段之航行時間均較長，所以基於成本效益與安全因素之考量，其排班法規通常會規定「一個勤務中所能包含之航段數不能超過 4 個」；相對的，在國內航線之空勤組員排班問題中，由於每個航段之航行時間均較短，所以基於成本效益與安全因素之考量，其排班法規通常會規定「一個勤務中所能包含之航段數不能超過 8 個」。根據此一特性，可明顯地得知，在國際航線之空勤組員排班問題中，其勤務型態之種類會較國內航線之空勤組員排班問題少。

4.1.3 勤務組合型態

大致而言，勤務組合型態可依據下列項四因素來做區分：

1. 依「勤務組合是否考慮空載因素」來區分：

a. 基本勤務組合(Basic Pairings)：

乃指「不包含空載之所有可行勤務組合」。

b. 延伸勤務組合(Extended Pairings)：

乃指「以基本可行勤務組合為基礎且包含空載之所有可行勤務組合」，亦即一個包含 n 個航段之基本可行勤務組合，其相對應之延伸可行勤務組合之種類有包含 1 個空載之 C_1^n 種可行勤務組合、包含 2 個空載之 C_2^n 種可行勤務組合、...、包含 $n-1$ 個空載之 C_{n-1}^n 種可行勤務組合，共計 $\sum_{i=1}^{n-1} C_i^n$ 種。由此可知，國內航線組員排班問題之延伸可行勤務組合之數目較國際航線組員排班問題多。

2. 依「勤務組合中所包含之實際航段數」來區分：

用 PairN，表示包含 N 個實際航段之勤務組合類型，對於一個限制勤務組合中最多只能包含 N_p 個航段數之勤務組合產生問題中，其可能之勤務組合類型包含 Pair2、Pair3、...、Pair N_p 。以 $N_p=6$ 為例，其可能之勤務組合類型包含 Pair2、Pair3、Pair4、Pair5、Pair6 五種。

3. 依「勤務組合中所包含之勤務型態」來區分：

由於可行勤務組合是由一連續之勤務所構成，所以，一個勤務組合中可能發生之勤務接續型態會跟「勤務組合中最多能包含之航段數 N_p 」與「勤務中能包含之最多航段 N_d 」兩項因素有關。例如：對於一個 $N_p=6$ 、 $N_d=4$ 之勤務組合產生問題而言，其可行勤務組合中之勤務接續類型種類如表 4.2 所示：

表 4.2 $N_p=6$ 、 $N_d=4$ 時之勤務組合類型與勤務接續類型

| 勤務組合類型 | | 可能之勤務接續類型 |
|--------|---------|---------------------------------|
| Pair2 | Pair2-1 | 2 個 duty1+4 個 duty0 |
| | Pair2-2 | 1 個 duty2+4 個 duty0 |
| Pair3 | Pair3-1 | 3 個 duty1+3 個 duty0 |
| | Pair3-2 | 1 個 duty2+1 個 duty1+3 個 duty0 |
| | Pair3-3 | 1 個 duty3+3 個 duty0 |
| Pair4 | Pair4-1 | 4 個 duty1+2 個 duty0 |
| | Pair4-2 | 2 個 duty1+1 個 duty2+2 個 duty0 |
| | Pair4-3 | 2 個 duty2 + 2 個 duty0 |
| | Pair4-4 | 1 個 duty4+2 個 duty0 |
| Pair5 | Pair5-1 | 5 個 duty1+1 個 duty0 |
| | Pair5-2 | 3 個 duty1+1 個 duty2+1 個 duty0 |
| | Pair5-3 | 1 個 duty1 + 2 個 duty2+1 個 duty0 |
| | Pair5-4 | 1 個 duty3+1 個 duty2 + 1 個 duty0 |
| | Pair5-5 | 1 個 duty4 + 1 個 duty1+1 個 duty0 |
| Pair6 | Pair6-1 | 6 個 duty1 |
| | Pair6-2 | 4 個 duty1+1 個 duty2 |
| | Pair6-3 | 3 個 duty1 與 1 個 duty3 |
| | Pair6-4 | 2 個 duty3 |
| | Pair6-5 | 3 個 duty2 |

| | | |
|--|---------|-------------------------------|
| | Pair6-6 | 1 個 duty1+1 個 duty2+1 個 duty3 |
| | Pair6-7 | 2 個 duty1+2 個 duty2 |
| | Pair6-8 | 2 個 duty1+1 個 duty4 |

4. 依「是否允許勤務組合之中間航段經過基地」來區分：

a. 允許勤務組合之中間航段經過基地：Pair

Pair 類型之勤務組合乃指「允許中間航段經過基地之可行勤務組合」，此類型之勤務組合大多發生在國內航線之勤務組合產生問題中，其發生之主要原因為國內航線中每個航段之航行時間均較短以及善用組員人力資源兩項因素。下圖 4.3 為 Pair 類型中，最基本、最簡單之可行勤務組合樣式。

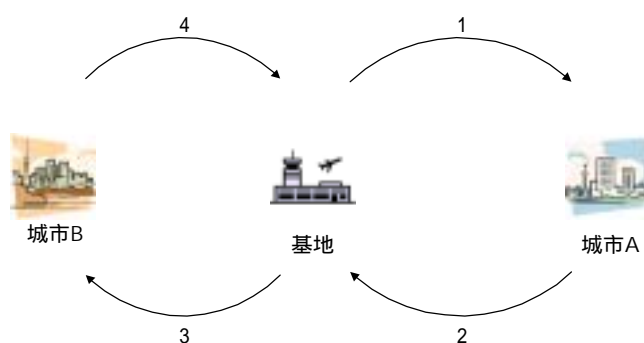


圖 4.3 Pair 型態中最基本之勤務組合樣式

b. 不允許勤務組合之中間航段經過基地：Pair

Pair 類型之勤務組合乃指「不允許中間航段經過基地之可行勤務組合」，此類型之勤務組合大多發生在國際航線之勤務組合產生問題中，其發生之主要原因為國際航線中每個航段之航行時間均較長以及組員之外宿成本兩項因素。下圖 4.4 為 Pair 類型中，最基本、最簡單之可行勤務組合樣式。

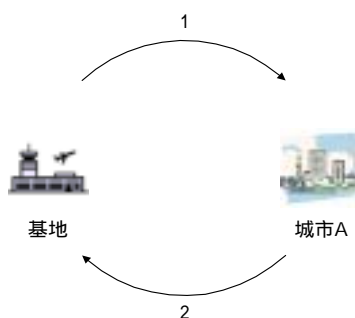


圖 4.4 Pair 型態中最基本之可行勤務組合樣式

4.2 勤務組合產生問題所考慮之勤務組合集合

綜合歸納文獻中各種排班模式所考慮之勤務組合集合類型，大致可依下列三項因素來區分：

1. 依「考慮之範圍」來區分：

a. TP 集合：全部可行勤務組合所形成之集合(Total Pairing Set)

由考慮各種排班因素(基地、機型接續、艙等、空載)之可行勤務組合所構成之一個集合，亦即，越複雜之排班模式其所考慮之 TP 集合規模將越龐大，而產生此集合之方式在文獻中被稱為 Explicit/Complete Enumeration Method。

b. PP 集合：經由篩選過之部分可行勤務組合所形成之集合(Partial Pairing Set)

經由一些篩選準則，從 TP 集合中所選出的一個子集合，而此子集合必須要符合「能涵蓋所有航段需求」之條件，而產生此子集合之方式在文獻中被稱為 Implicit/Partial Enumeration Method。若使用限制規劃方法來產生 PP 集合，則此集合又可稱為 CP 集合(Constrained Pairing Set)。

2. 依「包含空載與否」來區分

仔細區分 TP 或 CP 集合中之勤務組合型態，可明顯發現其包括基本可行勤務組合(Basic Pairing, BP)與延伸性勤務組合(Extended Pairing, EP)兩種。

a. BP 集合：由 BP 所構成之勤務組合集合(Basic Pairing Set)

所謂之基本可行勤務組合(BP)乃意指「不含空載之勤務組合」。

b. EP 集合：由 EP 所構成之勤務組合集合(Extended Pairing Set)

所謂之延伸性可行勤務組合(EP)乃意指「含空載之勤務組合」，其中又可依據勤務組合所包含之空載數分成 N_p-1 個子集合，即包括 EP_1 、 EP_2 、...、 $EP_{(N_p-1)}$ ，其中 EP_i 表示含 i 個空載數之延伸性勤務組合。EP 之產生主要是來自 BP，亦即，一個 BP 可依據其包含之航段數衍生成多個 EP；反之，一個 EP 只能對應至一個 BP。因此，當一個排班問題所包含之 BP 越多(BP 集合越大)，則其包含的 EP 將越多(EP 集合越大)。

對於一個排班問題的 TP 集合而言，可以充分運用 BP 集合與 EP 集合之關係來計算出 TP 集合之規模，例如對於一個排班問題而言，可以運用【式 4.1】之來計算，而此公式之形成，可參照表 4.3 所示，其中 $N_p=6$ 。

$$\sum_{k=2}^{N_p} N_k (1 + C_1^k + C_2^k + \dots + C_{k-1}^k)$$

符號說明：

【式 4.1】

N_p ：表示一個勤務組合中，最多能包含之航段數

N_k ：表示 Pair k 型態之勤務組合集合中，BP 集合之規模

表 4.3 排班問題規模之計算公式

| | BP | EP ₁ | EP ₂ | EP ₃ | EP ₄ | EP ₅ | 總計 |
|-------|----------------|--|--|--|--|--|---|
| Pair2 | N ₂ | C ₁ ² × N ₂ | N/A | N/A | N/A | N/A | N ₂ (1 + C ₁ ²) |
| Pair3 | N ₃ | C ₁ ³ × N ₃ | C ₂ ³ × N ₃ | N/A | N/A | N/A | N ₃ (1 + C ₁ ³ + C ₂ ³) |
| Pair4 | N ₄ | C ₁ ⁴ × N ₄ | C ₂ ⁴ × N ₄ | C ₃ ⁴ × N ₄ | N/A | N/A | N ₄ (1 + C ₁ ⁴ + C ₂ ⁴ + C ₃ ⁴) |
| Pair5 | N ₅ | C ₁ ⁵ × N ₅ | C ₂ ⁵ × N ₅ | C ₃ ⁵ × N ₅ | C ₄ ⁵ × N ₅ | N/A | N ₅ (1 + C ₁ ⁵ + C ₂ ⁵ + C ₃ ⁵ + C ₄ ⁵) |
| Pair6 | N ₆ | C ₁ ⁶ × N ₆ | C ₂ ⁶ × N ₆ | C ₃ ⁶ × N ₆ | C ₄ ⁶ × N ₆ | C ₅ ⁶ × N ₆ | N ₆ (1 + C ₁ ⁶ + C ₂ ⁶ + C ₃ ⁶ + C ₄ ⁶ + C ₅ ⁶) |
| 總計 | | | | | | | $\sum_{k=2}^6 N_k (1 + C_1^k + C_2^k + \dots + C_{k-1}^k)$ |

3. 依「勤務組合品質」來區分

a. GP 集合：品質優良之勤務組合集合(Good Pairing Set)

意指比值(TAFB/勤務組合總飛行時間)較小之勤務組合所形成之集合。

b. NGP 集合：品質較差之勤務組合集合(Non-Good Pairing Set)：

意指比值(TAFB/勤務組合總飛行時間)較大之勤務組合所形成之集合。

由於許多實務問題之規模皆很龐大，所以大部分之排班模式皆會運用一些隱含限制規則(Implicit Restriction Rule)以刪除某些品質較差的勤務組合，進而有效地降低排班模式之規模、加快問題求解速度且不失求解之品質。

一般而言，由於很難明確訂定「何謂品質優良之勤務組合」，所以本研究乃將以 CP 集合(Constrained Pairing Set)代表本研究所認為之優良勤務組合集合，此目的主要有二，一為「減少不同排班模式對於優良勤務組合之認知差異」，另一為「彰顯本研究所認定之優良勤務組合集合是使用限制規劃方式來產生的」。

4.3 勤務組合產生問題之相關法規限制

為維持飛行安全與空勤組員服務水準之要求，空勤組員被指派至飛機上服務時，需

符合各項排班法規對於空勤組員飛行時間(飛時)、工作時間(工時)與休息時間(休時)之相關限制。綜合文獻上之各種排班法法規，可依下列兩項因素來區分：

1. 依排班法規所屬之組織來區分：

分為民航法規、工會規定、航空公司空勤組員派遣規則三大類，而每類法規可能包含之法規規定如表 4.4 所示。

表 4.4 排班法規分類-依法規所屬之組織來區分

| 法規名稱 | 相關規定 |
|--------------|---|
| 民航法規 | 規定每種機型所需之最少服勤空服組員數 規定空服組員之工作時數、飛行時數、休息時數限制 |
| 工會規定 | 防止空服員因為過度疲倦而導致意外發生之派遣規則 領取合理報酬之薪資訂定規則 |
| 航空公司空勤組員派遣規則 | 防止空服員因為過度疲倦而導致意外發生之派遣規則 公平性之組員派遣規則等 |

2. 依排班法規所適用之對象來區分：

分為勤務之相關法規與勤務組合之相關排班法規兩大類，而每類法規可能包含之法規規定如表 4.5 所示。

表 4.5 排班法規分類-依法規所適用之對象來區分

| 勤務之相關排班法規 | 勤務組合之相關排班法規 |
|---------------------------------|---|
| 1. 限制每個勤務中最多可包含的航段數。 | 1. 限制每個勤務組合最多可包含的航段數。 |
| 2. 每個勤務的勤務總飛行時間必須小於等於某一個最大的法定值。 | 2. 限制每個勤務組合最多可包含的勤務數。 |
| 3. 每個勤務的值勤時間必須小於等於某一個最大的法定值。 | 3. 在每個勤務組合當中，任兩個勤務間的銜接時間必須大於等於一法定之最小休息時間。 |
| | 4. 每個勤務組合之的總旅程時間必須小於等於某一最大的法定值。 |

雖然每個國家之民法規不同、每個國家或航空公司之空勤組員工會規定不同、每個航空公司之空勤組員派遣規則不同，使得每個航空公司之勤務組合產生問題所考慮之排班法規會有些許之不同，但基本上皆必須考慮到表 4.4 所列之各項法規。以本研究之個案航空公司為例，綜合其實務運作時所考慮之各項排班法規對於國際航線之空勤組員飛時、工時、休時之相關規定，整理如下所述：

一、名詞定義：

1. 工作時間：空勤組員值勤某一勤務所需花費之時間，其計算標準乃自組員報到起至報離止，包含組員報到與報離時間、勤務中之總飛行時間、勤務中之總休憩時

間，亦稱作勤務之總歷時時間。

2. 飛行時間：空勤組員值勤某一勤務之總飛行時間，其計算航段飛行時間之標準乃自飛機起飛前起動發動機引擎至飛機落地後關閉發動機引擎之時間區段。
3. 休息時間：前次勤務工作時間結束至下次勤務工作時間開始之間隔時間。
4. 休憩時間：包含兩個以上航段之勤務中，兩連續航段間之間隔時間。

二、勤務中飛行時間、工作時間與勤務間休息時間之限制：

1. 連續 24 時內，工作時間不超過十四小時，飛行時間應不超過 9 小時。惟單趟落地航次可酌予延長，但需保留客位或床位不得少於 4 個。
2. 若工作時間超過 12 小時，則休息時間不得低於 24 小時。
3. 若工作時間超過 8 小時，不到十二小時，休息時間不得低於 12 小時。
4. 工作時間 8 小時以內，休息時間不得低於 8 小時。
5. 班機因天災事變或然發事件得延長工作時間，但應於延長開始後 24 小時內通知工會，延長之工作時間得於事後予以適當之休息時間。

三、勤務中航段接續之限制：

1. 兩連續值勤航段間之最短休憩時間必須大於 40 分鐘，最長休憩時間不得超過 4 小時。
2. 國際線，當天勤務之總起(降)次數以 4 次為限，即當天勤務最多可飛行四個航段。

四、勤務組合之限制：

1. 勤務組合中之航段數必須小於等於 6 個。
2. 勤務組合中之差旅時間必須小於 15 天。
3. 勤務組合中，除了起始航段之起點城市與結束航段之迄點城市為航空公司的組員基地外，其他航段之起點或迄點城市皆不能為航空公司的組員基地。

五、組員報到與報離時間之限制：

| | 國內航線 | 國際航線 |
|------|-------------------|--|
| 報到時間 | 首次任務於起飛前 60 分鐘報到 | 起飛前 140 分鐘報到 |
| | 非首次任務於起飛前 45 分鐘報到 | 連續假日提早 20 分鐘(越洋航線於 12 時至 17 時間，提早 10 分鐘) |
| 報離時間 | 飛機落地後 1 小時報離 | 飛機落地後 1 小時報離 |

4.4 勤務組合產生問題之成本計算方式

由於航空公司之勤務成本計算方式會受到其公司內部規定或工會規定等因素影響，使得文獻上並不存在一個一般化勤務組合成本計算公式，然而因為本研究之測試例題皆來自同一家航空公司，所以在本研究所構建之勤務組合產生問題限制規劃模式中，其勤務組合成本計算公式乃先依據個案公司之成本計算公式來訂定。

本研究在勤務組合成本之計算上，參酌個案航空公司之實際營運，將勤務組合成本區分為「飛行成本、空載成本、住宿成本、零用金成本」四大項來計算，其中，飛行成本之計算方式乃以單一空勤組員之總飛行時間乘上組員時薪成本而得，空載成本之計算方式乃以單一空勤組員之總空載時間乘上組員時薪成本並除以 2 而得，住宿成本之計算方式乃以單一空勤組員之住宿費乘以過夜天數而得，休息成本之計算方式乃以勤務組合之差旅時間 TAFB(時)乘以每小時之零用金費用而得。

相較於國外文獻之勤務組合成本計算方式，可明確地發現，個案公司沒有最低保障之勤務工資而文獻中有，但相對的個案公司有零用金之成本支出而文獻上沒有，由於此一不同點，使得個案公司之勤務組合成本計算方式較文獻所提及之計算方式更為簡單、更容易計算。

4.5 一般化勤務組合產生問題限制規劃模式

限制規劃最主要的特點就是其「問題模式之建構(Modeling)」與「問題模式之求解(Solving)」是互相獨立的，意即每當現實環境的改變使得限制式改變時，我們只要稍微修改問題模式之相關限制式，即可運用相同的求解演算法重新求解；相較於網路模式之求解架構，若是問題限制式改變時，可能要同時修改求解演算法或重新建立新的求解演算法，這種求解架構不僅使得程式開發者不易維護其撰寫的程式，並且對於面臨多變環境的公司而言，亦較無法採行運用網路模式所開發的系統。因此，不同於傳統將勤務組合產生問題建構成一個排班網路模式[37]，本研究將勤務組合產生問題構建成一個能同時考慮各種排班限制(如：工時、飛時、休時)與排班因素(如：組員基地、機型接續、艙等、空載)之一般化勤務組合產生問題限制規劃模式並將其命名為 CSP，而模式中之每個可行解即表示一個可行之勤務組合，此模式之求解概念如圖 4.5 所示。

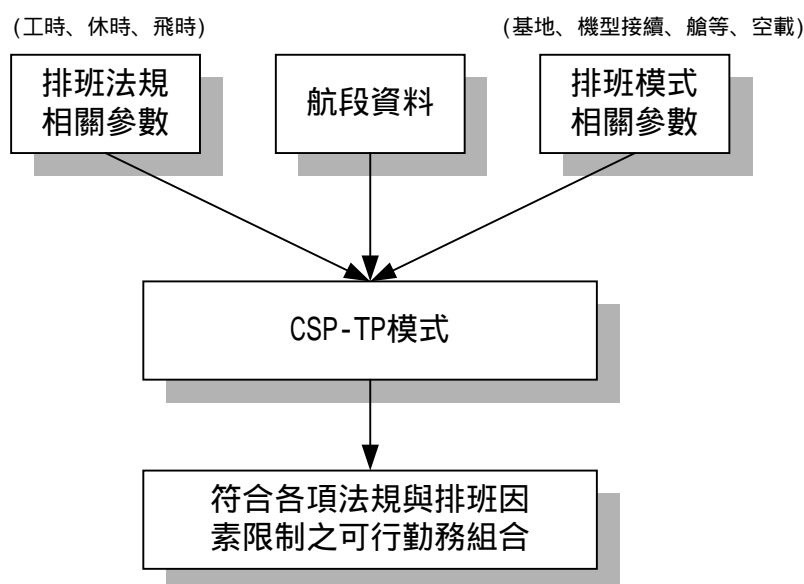


圖 4.5 CSP 模式之求解概念圖

本研究建構一般化勤務組合產生問題限制規劃模式之主要目的有三，一為「藉由限制規劃之求解機制，輔助排班人員或各種排班模式有效率地產生各類型之可行勤務組合」，二為「藉由限制規劃模式之模式化語言優點，輔助排班人員有效率地、系統化地管理其各項排班法規」，三為「藉由限制規劃模式之模式彈性，輔助排班人員或各種排班模式有效地產生其所需要之可行勤務組合類型」。以下將一一詳述本研究所構建之一般化勤務組合產生問題限制規劃模式。

4.5.1 模式輸入資料

茲將 CSP 模式所需之輸入資料，依據其資料特性分為航段資料、法規資料、成本資料、組員基地資料四大類，其細節資料如表 4.6。

表 4.6 勤務組合產生問題限制規劃模式之輸入資料

| 資料類型 | 資料內容 |
|--------|---|
| 航段資料 | 航段之起站、航段之起飛日、航段之起飛時間 航段之迄站、航段之到達日、航段之到達時間 航段之飛航時間、航段之使用機型 |
| 法規資料 | 勤務中飛行時間、工作時間與勤務間休息時間之限制 勤務中航段接續之限制、勤務組合之限制 組員報到與報離時間之限制 |
| 成本資料 | 零用金成本、各基地組員之平均時薪成本 各基地中各艙等組員之平均時薪成本 |
| 組員基地資料 | 目前航空公司所擁有之組員基地數與基地名稱 |

4.5.2 模式參數與變數

茲將勤務組合產生問題限制規劃模式之相關參數與變數分類成「航段資料參數與變數、成本資料參數、排班法規參數、排班因素參數、模式變數」四大類，並將其內容依序分別整理成表 4.7、表 4.8、表 4.9、表 4.10、表 4.11。

表 4.7 CSP 模式之航段資料參數與變數

| 航段資料參數與變數 | 變數說明 |
|-----------------------------------|--|
| N_f | 航段資料中之總航段數 |
| Hub | 航空公司之組員基地代碼 |
| $Org_i, \forall i=0 \sim N_f$ | $\forall i=1 \sim N_f$, Org_i 表示實際航段 i 之起站城市代碼 |
| | Org_0 表示虛擬航段之起站代碼 |
| $Dept_t_i, \forall i=0 \sim N_f$ | $\forall i=1 \sim N_f$, $Dept_t_i$ 表示實際航段 i 之起飛時間 |
| | $Dept_t_0$ 表示虛擬航段之起飛時間 |
| $Dept_d_i, \forall i=0 \sim N_f$ | $\forall i=1 \sim N_f$, $Dept_d_i$ 表示實際航段 i 之起飛日(星期 1-7) |

| | |
|----------------------------------|---|
| | $Dept_d_0$ 表示虛擬航段之起飛日(星期 0) |
| $Dst_i, \forall i=0 \sim N_f$ | $\forall i=1 \sim N_f$, Dst_i 表示實際航段 i 之迄站代碼 |
| | Dst_0 表示虛擬航段之迄站代碼 |
| $Arr_t_i, \forall i=0 \sim N_f$ | $\forall i=1 \sim N_f$, Arr_t_i 表示實際航段 i 之到達時間 |
| | Arr_t_0 表示虛擬航段之到達時間 |
| $Arr_d_i, \forall i=0 \sim N_f$ | $\forall i=1 \sim N_f$, Arr_d_i 表示實際航段 i 之到達日(星期 1-7) |
| | Arr_d_0 表示虛擬航段之到達日(星期 0) |
| $fly_h_i, \forall i=0 \sim N_f$ | $\forall i=1 \sim N_f$, fly_h_i 表示實際航段 i 之飛行時間 |
| | fly_h_0 表示虛擬航段之飛行時間, 且 $fly_h_0 = 0$ |
| $flt_T_i, \forall i=0 \sim N_f$ | $\forall i=1 \sim N_f$, flt_T_i 表示實際航段 i 所使用之機型 |
| | flt_T_0 表示虛擬航段所使用之機型, 且 $flt_T_0 = 0$ |

表 4.8 CSP 模式之成本資料參數

| 成本參數 | 參數說明 |
|---------------------|---------------|
| Fly_Cost | 組員的時薪成本(元/分) |
| $Petty_Cash_Cost$ | 組員的零用金成本(元/分) |
| $Resident_Cost$ | 組員的外宿成本(元/次) |

表 4.9 CSP 模式之排班法規參數

| 排班法規參數 | 參數說明 |
|---------------------------|--|
| N_p | 一個勤務組合中, 最多能包含之航段數 |
| N_d | 一個勤務中, 最多能包含之航段數且 $N_d \leq N_p$ |
| $nWBH$ | 排班法規對於組員勤務工時之限制, 可分為 $nWBH$ 個勤務工時限制區間 |
| $WBH[1..nWBH, 1..2]$ | 每個工時限制區間 i 之最小勤務工時 $WBH[i, 1]$ 與最大勤務工時 $WBH[i, 2]$ |
| $WBH_min_Rest[1..nWBH]$ | 若是組員之勤務工時介於勤務工時限制區間 i 之間, 則 |

| | |
|--------------------------|---|
| | 組員之休時必須大於等於 $WBH_minRest[i]$ |
| $max_duty_fly_h$ | 組員之勤務飛時，一定要小於等於 $max_duty_fly_h$ ，惟單趟落地可酌予延長 |
| min_sit_h | 法定最短休憩時間 |
| max_sit_h | 法定最長休憩時間 |
| $T_{check_in_base}$ | 組員在基地時，從報到至飛機起飛，所需花費之時間 |
| $T_{check_in_nonbase}$ | 組員在外站時，從報到至飛機起飛，所需花費之時間 |
| T_{check_out} | 組員飛機落地至報離，所需花費之時間 |

表 4.10 CSP 模式之排班因素參數

| 排班因素參數 | 參數說明 |
|---|---|
| $Deadhead_Open$ | 表示開啟何種形式之空載產生方式 |
| | 若 $Deadhead_Open=0$ ，表示模式不產生任何空載航段 |
| | 若 $Deadhead_Open=1$ ，表示模式只產生含 $nbDeadhead$ 個空載航段之勤務組合 |
| | 若 $Deadhead_Open=2$ ，表示模式只產生含 min_DH 至 max_DH 個空載航段之勤務組合 |
| $nbDeadhead$ | 表示要產生包含 $nbDeadhead$ 個空載航段之勤務組合 |
| min_DH | 表示要產生至少包含 $nbDeadhead$ 個空載航段之勤務組合 |
| max_DH | 表示要產生至多包含 $nbDeadhead$ 個空載航段之勤務組合 |
| $pair_sameT$ | 表示勤務組合中，每個航段所使用之機型必須相同 |
| $PairingType$ | 表示要產生何種型態之勤務組合 |
| max_Rest | 表示勤務組合中連續兩航段間之休息時間不能超過 max_Rest (可用來減少組員在外站之休息時間、提升勤務組合之品質) |
| N_c | 模式考慮之組員艙等數 |
| $Cabin_Class$ | 模式所考慮之組員艙等(值域只能介於 1 至 N_c 之間) |
| $Dual_Cost_{ij}$ $, \forall i \in 1 \sim N_c, j \in 1 \sim N_p$ | $Dual_Cost_{ij}$ ，表示 i 艙等組員值勤 j 航段之對偶成本 (i 值越小，表示組員之艙等等級越高) |

表 4.11 CSP 模式之變數

| 模式變數 | 變數值域 | 變數說明 |
|------|------|------|
|------|------|------|

| | | |
|--------------------------------------|--------------|---|
| $X_i, \forall i=1 \sim (N_p+1)$ | $0 \sim N_f$ | X_i , 表示勤務組合 X 中 , 第 i 個航段的值 |
| | | 變數值域 $1 \sim N_f$ 分別代表一個真實航段 , 變數值域 0 代表一個虛擬航段 |
| | | $\forall i=1 \sim N_p$, X_i 之下一航段為 X_{i+1} ; 反之 , X_{i+1} 之前一航段為 X_i |
| | | $X = (X_1, X_2, \dots, X_{N_p}, X_{N_p+1})$ 表示一個完整之勤務組合 , 且令 $X_{N_p+1} = 0$ |
| | | $\forall i=1 \sim N_p$, 若 $X_i = 0$, 則表示勤務組合 X 中 , 第 i 個航段為虛擬航段 |
| $T_i, \forall i=1 \sim N_p$ | 自然數 N | T_i , 表示勤務組合 X 中 , i 航段落地後至 $i+1$ 航段起飛前之相隔時間 |
| | | 勤務組合 X 中 , 若 i 航段與 $i+1$ 航段皆為虛擬航段時 , 則 $T_i = 0$ |
| | | 勤務組合 X 中 , 若 i 航段為真實航段且 $i+1$ 航段為虛擬航段時 , 則 $T_i = 0$ |
| $DayDiff_i, \forall i=1 \sim N_p$ | 自然數 N | $DayDiff_i$, 表示勤務組合 X 中 , i 航段之抵達日與 $i+1$ 航段之起飛日間之日差 |
| | | 勤務組合 X 中 , 若 i 航段與 $i+1$ 航段皆為虛擬航段時 , 則 $DayDiff_i = 0$ |
| $Deadhead_i, \forall i=1 \sim N_p$ | {0,1} | $Deadhead_i$, 表示勤務組合 X 中之第 i 個航段是否為一個空載航段 |
| | | 若其值為 1 , 表示為空載航段 若其值為 0 , 表示為非空載航段 |
| $Check_In_i, \forall i=1 \sim N_p$ | {0,1} | $Check_In_i$, 表示勤務組合 X 中 , 組員在值勤第 i 個航段前 , 是否有執行報到手續 |
| | | 若其值為 1 , 表示有執行報到手續 若其值為 0 , 表示沒有執行報到手續 |
| $Check_Out_i, \forall i=1 \sim N_p$ | {0,1} | $Check_Out_i$, 表示勤務組合 X 中 , 組員在值勤第 i 個航段後 , 是否有執行報離手續 |
| | | 若其值為 1 , 表示有執行報離手續 若其值為 0 , 表示沒有執行報離手續 |

| | | |
|------------------------------------|-----------|---|
| $Resident_i, \forall i=1 \sim N_p$ | $\{0,1\}$ | $Resident_i$, 表示勤務組合 X 中, 組員在值勤第 i 個航段後, 是否有在外站住宿 若其值為 1, 表示有外宿 若其值為 0, 表示沒有外宿 |
| $Total_Fly_Time$ | 自然數 N | 勤務組合之總飛行時間 |
| $Total_Duty_Time$ | 自然數 N | 勤務組合之總勤務時間 |
| $Total_Sit_Time$ | 自然數 N | 勤務組合之總休憩時間 |
| $Total_Rest_Time$ | 自然數 N | 勤務組合之總休息時間 |
| $Total_Check_In_Time$ | 自然數 N | 勤務組合之總報到時間 |
| $Total_Check_Out_Time$ | 自然數 N | 勤務組合之總報離時間 |
| $Total_Resident_Day$ | 自然數 N | 勤務組合之總外宿天數 |
| $TAFB$ | 自然數 N | 勤務組合之差旅時間 |
| Pay | 自然數 N | 勤務組合之總成本 |
| $Reduced_Cost$ | 自然數 N | 勤務組合之 Reduced_Cost |

計算兩實際航段間到達
日與起飛日之日差陣列

$$Day_Diff[i][i+1] =$$

| | | $i+1$ | | | | | | | |
|-----|---|-------|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| i | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 1 | 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| | 2 | 0 | 6 | 0 | 1 | 2 | 3 | 4 | 5 |
| | 3 | 0 | 5 | 6 | 0 | 1 | 2 | 3 | 4 |
| | 4 | 0 | 4 | 5 | 6 | 0 | 1 | 2 | 3 |
| | 5 | 0 | 3 | 4 | 5 | 6 | 0 | 1 | 2 |
| | 6 | 0 | 2 | 3 | 4 | 5 | 6 | 0 | 1 |
| | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 0 |

4.5.3 模式限制式

由於大部分之排班法規皆屬於條件限制式(邏輯限制式)之類型，所以本研究在構建勤務組合產生問題限制規劃模式中之法規限制式時，大都是運用邏輯限制式來定義與量化每一項排班法規限制。茲將勤務組合產生問題限制規劃模式之限制式，整理成下列五大類，並詳述各項限制式之表達方式。

【限制式集合 1】：勤務中飛時、工時、休時與勤務間休時之相關限制式

必須考慮所有之勤務類型：duty1 至 duty N_d ，以下依序列出 duty1 至 duty N_d 之相關限制式。

勤務類型：duty1

| 若勤務類型為 duty1，則需符合下列條件 | |
|---|---------------------------|
| $X_i \in \text{duty1}$ $\Rightarrow (\text{Check_In}_i = 1) \wedge (\text{Check_Out}_i = 1)$ | $\forall i = 1 \quad N_p$ |

| 若勤務類型為 duty1，則勤務之總飛行時間必須小於 max_duty_fly_h 且勤務之總歷時時間必須小於等於 WBH[nWBH,2]，惟單趟落地可酌予延長 | |
|---|---------------------------|
| $X_i \in \text{duty1}$ $\Rightarrow (\text{fly_h}_{X_i} \leq \text{max_duty_fly_h}) \vee (\text{fly_h}_{X_i} > \text{max_duty_fly_h})$ | $\forall i = 1 \quad N_p$ |

| 若勤務類型為 duty1 且其工時介於 WBH[j,1]-WBH[j,2]小時、飛時 max_duty_fly_h 小時，則休時 WBH_minRest[j]小時 | |
|---|---|
| $(X_i \in \text{duty1}) \wedge (\text{WBH}[j,1] < (\text{Org}_{X_i} = \text{Hub}) * T_{\text{check_in_base}}$ $+ (\text{Org}_{X_i} \neq \text{Hub}) * T_{\text{check_in_nonbase}} + \text{fly_h}_{X_i} + T_{\text{check_out}} \quad \text{WBH}[j,2])$ $\wedge (\text{fly_h}_{X_i} \leq \text{max_duty_fly_h})$ $\Rightarrow (T_i \leq \text{WBH_minRest}[j] + (\text{Dst}_{X_i} = \text{Hub}) * T_{\text{check_in_base}}$ $+ (\text{Dst}_{X_i} \neq \text{Hub}) * T_{\text{check_in_nonbase}} + T_{\text{check_out}}) \vee (T_i = 0)$ | $\forall i = 1 \quad N_p,$ $\forall j = 1 \quad n\text{WBH}$ |

勤務類型：duty2

| 若勤務類型為 duty2，則需符合下列條件 | |
|---|-----------------------------------|
| $X_i, X_{i+1} \in \text{duty2}$ $\Rightarrow (\text{Check_In}_i = 1) \wedge (\text{Check_In}_{i+1} = 0) \wedge (\text{Resident}_i = 0)$ $\wedge (\text{Check_Out}_i = 0) \wedge (\text{Check_Out}_{i+1} = 1)$ | $\forall i = 1 \quad (N_p - 1)$ |

| 若勤務類型為 duty2，則勤務中之航段接續時間 min_sit_h 與 max_sit_h 之間 | |
|--|-------------------------------|
| $X_i \ X_{i+1} \in duty2$ $\Rightarrow (min_sit_h < T_i < max_sit_h)$ | $\forall i=1 \quad (N_p - 1)$ |

| 若勤務類型為 duty2，則勤務之總飛行時間必須小於 $max_duty_fly_h$ 且勤務之總歷時時間必須小於等於 $WBH[nWBH,2]$ ，惟單趟落地可酌予延長 | |
|---|-------------------------------|
| $X_i \ X_{i+1} \in duty2$ $\Rightarrow (fly_h_{X_i} + fly_h_{X_{i+1}} \leq max_duty_fly_h)$ $\wedge ((Org_{X_i} = Hub) * T_{check_in_base}$ $+ (Org_{X_i} \neq Hub) * T_{check_in_nonbase} + fly_h_{X_i} + T_i + fly_h_{X_{i+1}}$ $+ T_{check_out} \leq WBH[nWBH,2])$ | $\forall i=1 \quad (N_p - 1)$ |

| 若勤務類型為 duty2 且其工時介於 $WBH[j,1]$ - $WBH[j,2]$ 小時、飛時 $max_duty_fly_h$ 小時，則休時 $WBH_minRest[j]$ 小時 | |
|--|--|
| $(X_i \ X_{i+1} \in duty2) \wedge (WBH[j,1] < (Org_{X_i} = Hub) * T_{check_in_base}$ $+ (Org_{X_i} \neq Hub) * T_{check_in_nonbase} + fly_h_{X_i} + T_i + fly_h_{X_{i+1}}$ $+ T_{check_out} \leq WBH[j,2]) \wedge (fly_h_{X_i} + fly_h_{X_{i+1}} \leq max_duty_fly_h)$ $\wedge (min_sit_h < T_i < max_sit_h)$ $\Rightarrow (T_{i+1} \geq WBH_minRest[j] + (Dst_{X_{i+1}} = Hub) * T_{check_in_base}$ $+ (Dst_{X_{i+1}} \neq Hub) * T_{check_in_nonbase} + T_{check_out})$ $\vee (T_{i+1} = 0)$ | $\forall i=1 \quad (N_p - 1),$ $\forall j=1 \quad nWBH$ |

.....

勤務類型：duty N_d

| 若勤務類型為 duty N_d ，則需符合下列條件 | |
|--|---------------------------------------|
| $X_i \ X_{i+1} \dots X_{i+(N_d-1)} \in dutyN_d$ $\Rightarrow (Check_In_i = 1) \wedge (\bigwedge_{j=1}^{(N_d-1)} (Check_In_{i+j} = 0))$ $\wedge (\bigwedge_{j=0}^{(N_d-2)} (Resident_{i+j} = 0)) \wedge ((\bigwedge_{j=0}^{(N_d-2)} (Check_Out_{i+j} = 0))$ $\wedge (Check_Out_{i+(N_d-1)} = 1)$ | $\forall i=1 \quad (N_p - (N_d - 1))$ |

| 若勤務類型為 $duty N_d$, 則勤務中之航段接續時間 min_sit_h 與 max_sit_h 之間 | |
|--|---|
| $X_i, X_{i+1}, \dots, X_{i+(N_d-1)} \in duty N_d$ $\Rightarrow (\bigwedge_{j=0}^{(N_d-2)} (min_sit_h \leq T_{i+j} \leq max_sit_h))$ | $\forall i = 1 \quad (N_p - (N_d - 1))$ |

| 若勤務類型為 $duty N_d$, 則勤務之總飛行時間必須小於 $max_duty_fly_h$ 且勤務之總歷時時間必須小於等於 $WBH[nWBH,2]$, 惟單趟落地可酌予延長 | |
|---|---|
| $X_i, X_{i+1}, \dots, X_{i+(N_d-1)} \in duty N_d$ $\Rightarrow (\sum_{j=0}^{j=(N_d-1)} fly_h_{X_{i+j}} \leq max_duty_fly_h)$ $\wedge ((Org_{X_i} = Hub) * T_{check_in_base} + (Org_{X_i} \neq Hub) * T_{check_in_nonbase}$ $+ \sum_{j=0}^{j=(N_d-1)} fly_h_{X_{i+j}} + \sum_{j=0}^{j=(N_d-2)} T_{i+j} + T_{check_out} \leq WBH[nWBH,2])$ | $\forall i = 1 \quad (N_p - (N_d - 1))$ |

| 若勤務類型為 $duty N_d$ 且其工時介於 $WBH[j,1]-WBH[j,2]$ 小時、飛時 $max_duty_fly_h$ 小時、則休時 $WBH_minRest[j]$ 小時 | |
|---|---|
| $(X_i, X_{i+1}, \dots, X_{i+(N_d-1)} \in duty N_d) \wedge (WBH[j,1] <$ $(Org_{X_i} = Hub) * T_{check_in_base} + (Org_{X_i} \neq Hub) * T_{check_in_nonbase}$ $+ \sum_{k=0}^{k=(N_d-1)} fly_h_{X_{i+k}} + \sum_{k=0}^{k=(N_d-2)} T_{i+k} + T_{check_out} \leq WBH[j,2])$ $\wedge (\sum_{k=0}^{k=(N_d-1)} fly_h_{X_{i+k}} \leq max_duty_fly_h)$ $\wedge (\bigwedge_{k=0}^{(N_d-2)} (min_sit_h < T_{i+k} < max_sit_h)$ $\Rightarrow (T_{i+(N_d-1)} \geq WBH_minRest[j]$ $+ (Dst_{X_{i+(N_d-1)}} = Hub) * T_{check_in_base}$ $+ (Dst_{X_{i+(N_d-1)}} \neq Hub) * T_{check_in_nonbase} + T_{check_out})$ $\vee (T_{i+(N_d-1)} = 0)$ | $\forall i = 1 \quad (N_p - (N_d - 1)) ,$ $\forall j = 1 \quad nWBH$ |

【限制式集合 2】：航段接續之相關限制式

時間方面：

| 若發生兩實際航段接續，則其中間之休息時間一定要大於 min_sit_h | |
|--|---------------------------|
| $(fly_h_{X_i} \neq 0) \wedge (fly_h_{X_{i+1}} \neq 0)$ $\Rightarrow T_i \geq min_sit_h$ | $\forall i = 1 \quad N_p$ |

| 若兩實際航段間之接續時間大於 max_sit_h , 則必須安排組員至旅館中休息 | |
|--|--------------------------|
| $T_i > max_sit_h$ $\Rightarrow T_i = WBH[1] + (Dst_{X_i} = Hub) * T_{check_in_base}$ $+ (Dst_{X_i} \neq Hub) * T_{check_in_nonbase}$ | $\forall i = 1 \sim N_p$ |

地點方面：

| 勤務組合中，起始航段之起站與結束航段之迄站皆需為同一個組員基地 | |
|-------------------------------------|--|
| $Org_{X_1} = Dst_{X_{N_p+1}} = Hub$ | |

| 若發生兩實際航段接續，則前一航段之迄站即為後一航段之起站 | |
|------------------------------|--------------------------|
| $Org_{X_{i+1}} = Dst_{X_i}$ | $\forall i = 1 \sim N_p$ |

虛擬航段之接續方面：

| 限制虛擬航段只能出現在勤務組合中之尾端 | |
|--|--|
| $X_1 \neq 0$ | |
| 若航段 i 為虛擬航段，則其後續之航段(j>i)亦皆為虛擬航段 ($X_i = 0$) \Rightarrow ($X_j = 0$) | $\forall i, j = 1 \sim N_p \ \& \ 1 < i < j$ |

【限制式集合 3】：報到、報離、外宿之相關限制式

| 組員在值勤第一個航段前，一定要執行報到手續 | |
|-----------------------|--|
| $Check_In_1 = 1$ | |

| 組員在值勤最後一個航段後，一定要執行報離手續 | |
|------------------------|--|
| $Check_Out_{N_p} = 1$ | |

| 勤務組合中，若航段 i 為虛擬航段，則不需執行報到、報離手續且也不會發生外宿 | |
|--|--|
| $(X_i = 0) \Rightarrow (Check_In_i = 0) \wedge (Check_Out_i = 0) \wedge (Resident_i = 0) , \forall i = 1 \sim N_p$ | |

| 若兩實際航段間之接續時間屬於休憩時間，則前一航段不需執行報離手續、後一航段不需執行報到手續且前一航段亦不會發生外宿 | |
|---|--|
|---|--|

| |
|--|
| $(min_sit_h \leq T_i \leq max_sit_h)$ $\Rightarrow (Check_Out_i = 0) \wedge (Check_In_{i+1} = 0) \wedge (Resident_i = 0), \forall i=1 \sim N_p$ |
|--|

若兩實際航段間之接續時間屬於休息時間，則前一航段需執行報離手續、後一航段需執行報到手續

| |
|--|
| $(T_i > max_sit_h)$ $\Rightarrow (Check_Out_i = 1) \wedge (Check_In_{i+1} = 1), \forall i=1 \sim N_p$ |
|--|

| | |
|--|------------------------|
| 若勤務類型為 duty1 且勤務之報離地點為外站，則組員必須外宿 | |
| $(X_i \in duty1) \& (Dst_i \neq Hub) \Rightarrow (Resident_i = 1)$ | $\forall i=1 \sim N_p$ |
| 若勤務類型為 duty1 且勤務之報離地點為基地，則組員不必須外宿 | |
| $(X_i \in duty1) \& (Dst_i = Hub) \Rightarrow (Resident_i = 0)$ | $\forall i=1 \sim N_p$ |

| | |
|---|------------------------------|
| 若勤務類型為 duty2 且勤務之報離地點為外站，則組員必須外宿 | |
| $(X_i, X_{i+1} \in duty2) \& (Dst_{i+1} \neq Hub) \Rightarrow (Resident_{i+1} = 1)$ | $\forall i=1 \sim (N_p - 1)$ |
| 若勤務類型為 duty2 且勤務之報離地點為基地，則組員不必須外宿 | |
| $(X_i, X_{i+1} \in duty2) \& (Dst_{i+1} = Hub) \Rightarrow (Resident_{i+1} = 0)$ | $\forall i=1 \sim (N_p - 1)$ |

.....

| | |
|--|--------------------------------------|
| 若勤務類型為 duty N_d 且勤務之報離地點為外站，則組員必須外宿 | |
| $(X_i, X_{i+1}, \dots, X_{i+(N_d-1)} \in dutyN_d) \& (Dst_{i+(N_d-1)} \neq Hub)$ $\Rightarrow (Resident_{i+(N_d-1)} = 1)$ | $\forall i=1 \sim (N_p - (N_d - 1))$ |
| 若勤務類型為 duty N_d 且勤務之報離地點為基地，則組員不必須外宿 | |
| $(X_i, X_{i+1}, \dots, X_{i+(N_d-1)} \in dutyN_d) \& (Dst_{i+(N_d-1)} = Hub)$ $\Rightarrow (Resident_{i+(N_d-1)} = 0)$ | $\forall i=1 \sim (N_p - (N_d - 1))$ |

| | |
|--|--|
| 勤務組合中，組員值勤報到手續之總次數需等於值勤報離手續之總次數 | |
| $\sum_{i=1}^{N_p} (Check_In_i = 1) = \sum_{i=1}^{N_p} (Check_Out_i = 1)$ | |

【限制式集合 4】：排班因素之相關限制式

機型接續因素：

若 $pair_sameT=1$ ，則勤務組合中每個實際航段所使用之機型必須皆相同

$$(pair_sameT = 1)$$

\Rightarrow

空載因素：

若 $Deadhead_Open=0$ ，則勤務組合中之每個航段皆為非空載航段

$$(Deadhead_Open = 0) \Rightarrow \bigwedge_{i=1}^{N_d} (Deadhead_i = 0)$$

若 $Deadhead_Open \neq 0$ ，且航段 i 為虛擬航段，則航段 i 一定為非空載航段

$$(Deadhead_Open \neq 1)$$

$$\Rightarrow ((X_i = 0) \Rightarrow (Deadhead_i = 0)), \forall i = 1 \sim N_p$$

若 $Deadhead_Open \neq 0$ ，且航段 i 為實際航段，則航段 i 可能為空載航段，亦可能為非空載航段

$$(Deadhead_Open \neq 1)$$

$$\Rightarrow ((X_i \neq 0) \Rightarrow (Deadhead_i = 1) \vee (Deadhead_i = 0)), \forall i = 1 \sim N_p$$

若 $Deadhead_Open \neq 0$ ，則勤務組合中所含空載航段之個數一定要小於勤務組合中所含實際航段個數

$$(Deadhead_Open \neq 1)$$

$$\Rightarrow \sum_{i=1}^{N_p} (Deadhead_i = 1) < \sum_{i=1}^{N_p} (X_i \neq 0)$$

若 $Deadhead_Open=1$ ，則勤務組合中一定包含 $nbDeadhead$ 個空載航段

$$(Deadhead_Open = 1)$$

$$\Rightarrow (\sum_{i=1}^{N_p} (Deadhead_i = 1) = nbDeadhead)$$

若 $Deadhead_Open=2$ ，則勤務組合中所包含之空載航段數一定介於 min_DH 至 max_DH 之間

$$(Deadhead_Open = 2)$$

$$\Rightarrow (min_DH \leq \sum_{i=1}^{N_p} (Deadhead_i = 1) \leq max_DH)$$

勤務組合型態因素：

若 $PairingType=1$ ，則產生 Pair 型態之勤務組合

($PairingType = 1$)

$$\Rightarrow ((Dst_{x_i} = Hub) \Rightarrow (X_j = 0)), \forall i, j=1 \sim N_p \& 1 < i < j$$

若 $PairingType=2$ ，則產生 Pair 型態之勤務組合

($PairingType = 2$)

$$\Rightarrow ((Dst_{x_i} = Hub) \Rightarrow (min_sit_h \leq T_i \leq max_sit_h) \vee (X_{i+1} = 0)), \forall i=1 \sim N_p$$

【時間計算公式】

航段間日差數之計算公式

$$DayDiff_i = Day_Diff[Arr_d_{x_i}][Dep_d_{x_{i+1}}]$$

航段間相隔時間之計算公式

$$T_i = DayDiff_i \times 24 \times 60 + (X_{i+1} \neq 0) \times Dep_t_{x_{i+1}} - (X_{i+1} \neq 0) \times Arr_t_{x_i}$$

總飛行時間之計算公式

$$Total_Fly_Time = \sum_{i=1}^{N_p} fly_h_{x_i}$$

總勤務時間之計算公式

$$Total_Duty_Time = \sum_{i=1}^{N_p} fly_h_{x_i} + Total_Sit_Time$$

總休憩時間之計算公式

$$Total_Sit_Time = \sum_{i=1}^{N_p} (min_sit_h \leq T_i \leq max_sit_h) \times T_i$$

總休息時間之計算公式

$$Total_Rest_Time = \sum_{i=1}^{N_p} (T_i > max_sit_h) \times T_i \\ - \sum_{i=1}^{N_p} (Resident_i = 1) \times (T_{check_out} + T_{check_in_nonbase})$$

總報到時間之計算公式

$$\begin{aligned} Total_Check_In_Time &= \sum_{i=1}^{N_p} (Check_In_i = 1) \times (Org_{x_i} = Hub) \times T_{check_in_base} \\ &+ \sum_{i=1}^{N_p} (Check_In_i = 1) \times (Org_{x_i} \neq Hub) \times T_{check_in_nonbase} \end{aligned}$$

總報離時間之計算公式

$$Total_Check_Out_Time = \sum_{i=1}^{N_p} (Check_In_i = 1) \times T_{check_out}$$

差旅時間之計算公式

$$\begin{aligned} TAFB &= Total_Fly_Time + Total_Sit_Time + Total_Rest_Time \\ &+ Total_Check_In_Time + Total_Check_Out_Time \end{aligned}$$

總外宿次數之計算公式

$$\begin{aligned} Total_Resident_Day &= \sum_{i=1}^{N_p} (Resident_i = 1) \times (DayDiff_i \neq 0) \\ &+ \sum_{i=1}^{N_p} (Resident_i = 1) \times (DayDiff_i = 0) \end{aligned}$$

【成本計算公式】

勤務組合之成本計算公式(可根據航空公司之需求，自行定義自己的成本計算公式)

$$\begin{aligned} Pay &= Total_Fly_Time \times Fly_Cost \\ &+ TAFB \times Petty_Cash_Cost + Total_Resident_Day \times Resident_Cost \end{aligned}$$

【TAFB 限制式】

兩連續航段間之休息時間，一定要小於等於 max_Rest

$$T_i \leq max_Rest, \forall i = 1 \sim N_p$$

【Reduced_Cost 限制式】

勤務組合之 Reduced_Cost

$$Reduced_Cost = Pay - \sum_{i=1}^{Cabin_Class} \sum_{j=1}^{N_p} (X_i \neq 0) \times Dual_Cost_{ij}$$

4.6 勤務組合集合產生架構

一般化勤務組合產生問題限制規劃模式(CSP)為一個最基本、最核心的勤務組合產生問題限制規劃模式，若要使用 CSP 模式來產生 TP 或 CP 集合，則必須搭配適當的集合產生架構方能產生 TP 或 CP 集合。以下即針對本研究所提出之兩種集合產生架構作一詳細之介紹。

4.6.1 TP 集合產生架構

TP 集合產生架構之主要功能在於「輔助 CSP 模式，產生包含於 TP 集合中之所有可行勤務組合」，然而由於不同的排班模式其所需考慮之 TP 集合範圍將有所不同，所以本研究嘗試建立一個能同時考慮各種排班因素(基地、機型接續、艙等、空載)之 TP 集合產生架構，使得各種排班模式能夠過此一集合產生架構考慮其模式所對應之 TP 集合。

TP 集合產生架構之流程細節如圖 4.6 所示，在排班因素方面，其中將基地與艙等因素放在外層之主要原因在於「不同基地組員能值勤之航段集合不太相同，而且不同基地、不同艙等組員之薪資計算方式不太相同」，而把機型接續與空載因素放在內層考慮之主要原因在於「此兩種排班因素可模式化成排班模式限制式，並放置在 CSP 模式中」。

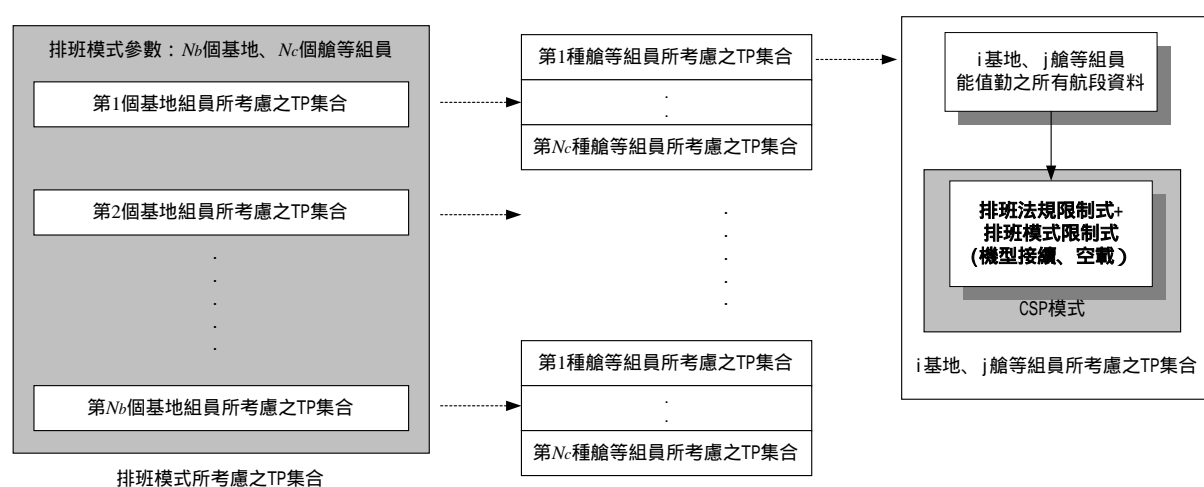


圖 4.6 TP 集合產生架構

根據 BP 與 EP 之概念，將 TP 集合作適度的切割，形成 BP 集合與 EP 集合兩大類，並且再針對 EP 集合進行切割成 EP₁、EP₂、...、EP_(N_p-1) 共 N_p-1 個子集合，其集合示意圖如圖 4.7 所示。



圖 4.7 TP 集合所包含之子集合

4.6.2 CP 集合產生架構

所謂的 CP 集合(Constrained Pairing Set)乃意指「由航段接續時間較短之勤務組合所構成之勤務組合集合」，為了透過限定排班模式只考慮品質優良之勤務組合，來縮減各種排班模式之規模與加快模式求解效率，本研究根據國際航線後艙組員排班問題之特性(稀疏之航線班次、長程航段)，構建出一個能同時考慮各種排班因素(基地、機型接續、艙等、空載)之 CP 集合產生架構，透過此一架構，每一種排班模式皆能找到其對應之 CP 集合。

CP 集合產生架構之流程細節如圖 4.8 所示，其架構大至上與 TP 集合產生架構相同，其間之主要差異在於「各基地、各艙等組員之勤務組合集合產生程序」，對於 CP 集合產生架構而言，由於其必須根據不同的航線班次密集度資料，來加入不同的航段接續限制式(TAFB 限制式)來產生 CP 集合，所以在 CP 集合產生架構中，其必須先對航段資料作分群，然後在 CSP 模式中加入適當的 TAFB 限制式，方能產生 CP 集合。

對於各基地、各艙等組員之 CP 集合產生程序而言，其運作原理為「透過航段資料分群之概念，配合 CSP 中所定義之 TAFB 限制式，來產生 (TAFB/勤務組合總飛行時間) 比值較低之優良勤務組合」，而詳細之運作細節則可歸納成下列五大步驟：

步驟 1：將所有航段資料，依組員基地分成各基地組員能值勤之航段集合

步驟 2：將基地組員能值勤之所有航段依據航線班次密集度分類準則進行分群

由於個案航空公司之航段資料特性，本研究定義一項航線班次密集分類度準則來分群每個基地組員所能值勤之航段資料，而航線班次密集度準則之主要目的在於「依據每週之航線班次頻率與航線中航段間之最短接續時間兩項因素，將航段資料分割成航線班次密集度高、中、低三類」，其準則之詳細內容如表 4.12 所示，而將航線航段資料分群之主要目的在於「促使勤務組合產生問題限制規劃模式能針對不同航線班次密集度之航段資料，快速而有效地地產生優良之可行勤務組合」。

表 4.12 航線班次密集度分類準則

| | |
|-------------------|---|
| 若航線航段資料 | 每週之航線班次 = 6 天且其航線中航段間之最短接續時間在 2 天之內 |
| 則屬「航線班次密集度高之航段資料」 | |
| 若航線航段資料 | 每週之航線班次 = 6 天且其航線中航段間之最短接續時間在 2 天之外且每週之航線班次 = 4 天、5 天且其航線中航段間之最短接續時間在 4 天之內 |
| 則屬「航線班次密集度中之航段資料」 | |
| 若航線航段資料 | 每週之航線班次 = 4 天且其航線中航段間之最短接續時間在 4 天之外且每週之航線班次 = 3 天且其航線中航段間之最短接續時間在 7 天之內 |
| 則屬「航線班次密集度低之航段資料」 | |

步驟 3：分別針對分群後航段資料，進行適度的合併成不同的群組航段資料

將步驟 1 分群出來之三種航段資料，群組成「包含航線班次密集度高、中、低之航段資料」、「包含航線班次密集度中、低之航段資料」、「包含航線班次密集度低之航段資料」三種不同的群組航段資料，此一群組之目的在於「防止不同航線班次密集度之航線航段資料有互相重疊之現象，以避免勤務組合產生問題限制規劃模式無法找到跨航線航段之可行勤務組合」。

步驟 4：分別針對每個群組航段資料，加入一項對應其航線班次密集度之 TAFB 限制式

所謂的 TAFB 限制式乃意指「透過限制勤務組合中之航段接緒時間，來控制(TAFB/勤務組合總飛行時間)比值之限制式」，本研究在 TAFB 限制式之構建上與一般文獻相同，同樣是「限制勤務組合中之航段間接續時間」來控制勤務組合之品質。對於**航線班次密集度高**之航段資料而言，由於其航段間之接續時間通常較短，所以可設定較短之航段間接續時間限制來控制其所產生之勤務組合品質；對於**航線班次密集度中**之航段資料而言，由於其航段間之接續時間不會太長亦不會太短，所以可設定中等長度之航段間接續時間限制來控制其所產生之勤務組合品質；對於**航線班次密集度低**之航段資料而言，由於其航段間之接續時間通常較長，所以必須設定較長之航段間接續時間限制來控制其所產生之勤務組合品質。亦即，對於不同航線班次密集度之航段資料而言，由於其最短之航段接緒時間長短不太相同，所以其相對應之 TAFB 限制式亦會不同。

步驟 5：運用勤務組合產生問題限制規劃模式求解各種型態之優良勤務組合

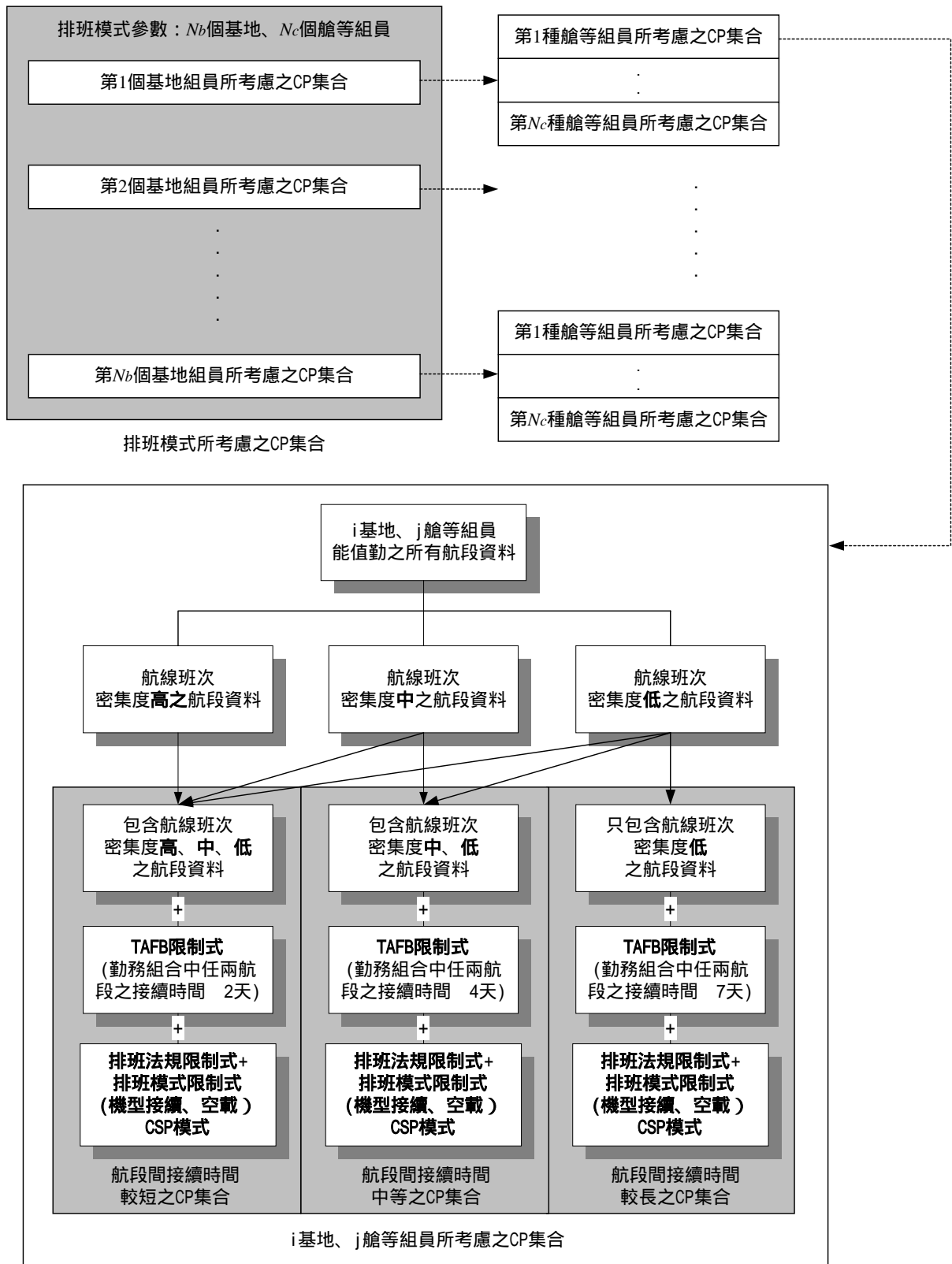


圖 4.8 CP 集合產生架構

根據 BP 與 EP 之概念，將 CP 集合作適度的切割，形成 CBP 集合(Constrained Basic Pairing Set)與 CEP 集合(Constrained Extended Pairing Set)兩大類，並且再針對 CEP 集合

進行切割成 CEP_1 、 CEP_2 、...、 $CEP_{(N_p-1)}$ 共 N_p-1 個子集合，其集合示意圖如圖 5.9 所示。

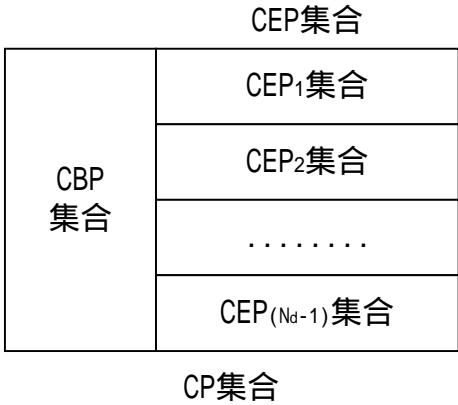


圖 4.9 CP 集合所包含之子集合

4.7 一般化勤務組合產生問題限制規劃模式之類型

為了後續文章敘述上之方便與模式的區別，此小節將定義與介紹本研究所使用到的勤務組合模式。

4.7.1 窮舉式的勤務組合產生模式

窮舉(Complete Enumeration)式的勤務組合產生模式乃意指「使用 CSP 模式+TP 集合產生架構的勤務組合產生模式」，若依模式所要產生的集合範圍來區分，可分為四種窮舉式的勤務組合產生模式，其相關細節如表 4.13 所示。

表 4.13 窮舉式的勤務組合產生模式

| 模式名稱 | 模式內容 |
|------------------------|------------------------------------|
| CSP-TP 模式 | CSP 模式 + TP 集合產生架構 |
| CSP-BP 模式 | CSP 模式 + TP 集合產生架構 + 不含空載限制式 |
| CSP-EP 模式 | CSP 模式 + TP 集合產生架構 + 只含空載限制式 |
| CSP-EP _i 模式 | CSP 模式 + TP 集合產生架構 + 只含 i 個空載空載限制式 |

4.7.2 限制列舉式的勤務組合產生模式

限制列舉(Constrained Enumeration)式的勤務組合產生模式乃意指「使用 CSP 模式+CP 集合產生架構的勤務組合產生模式」，此類型模式之主要概念在於「運用限制規劃來產生接續時間較短的優良勤務組合」，若依模式所要產生的集合範圍來區分，可分為四種窮舉式的勤務組合產生模式，其相關細節如表 4.14 所示。

表 4.14 限制列舉式的勤務組合產生模式

| 模式名稱 | 模式內容 |
|-------------------------|------------------------------------|
| CSP-CP 模式 | CSP 模式 + CP 集合產生架構 |
| CSP-CBP 模式 | CSP 模式 + CP 集合產生架構 + 不含空載限制式 |
| CSP-CEP 模式 | CSP 模式 + CP 集合產生架構 + 只含空載限制式 |
| CSP-CEP _i 模式 | CSP 模式 + CP 集合產生架構 + 只含 i 個空載空載限制式 |

第五章 後艙空勤組員排班模式與求解架構

5.1 八種後艙空勤組員排班模式

空勤組員排班模式之基本定義為「產生一組成本最小且涵蓋所有航段之可行勤務組合集合」[34]，亦即為文獻上所提之「最小成本勤務組合產生問題(Min-Cost Pairing Generation Problem)」，其定式通常為一個集合涵蓋或集合分割之數學規劃模式。然而，關於空勤組員排班模式之種類，又可根據航空公司之實際運作需求來分類，綜合歸納文獻上所有空勤組員排班模式，大致上可根據「組員之基地數(單基地或多基地)、機型接續限制考慮與否(單機型接續或多機型接續)、艙等考慮與否(單艙等或多艙等)、空載考慮與否」四項因素來區分，若先不考慮空載因素，基本上可將排班模式歸納為表 5.1 中所列的八種排班模式。

表 5.1 八種後艙空勤組員排班模式

| 模式 | 組員基地 | 機型 | 艙等 |
|-----|------|-------|-----|
| 模式一 | 單基地 | 單機型接續 | 單艙等 |
| 模式二 | 單基地 | 多機型接續 | 單艙等 |
| 模式三 | 單基地 | 單機型接續 | 多艙等 |
| 模式四 | 單基地 | 多機型接續 | 多艙等 |
| 模式五 | 多基地 | 單機型接續 | 單艙等 |
| 模式六 | 多基地 | 多機型接續 | 單艙等 |
| 模式七 | 多基地 | 單機型接續 | 多艙等 |
| 模式八 | 多基地 | 多機型接續 | 多艙等 |

其中，模式一為最基本之空勤組員排班模式，其數學規劃模式如【P₄】所示，模式八為最複雜之空勤組員排班模式，其數學規劃模式如【P₅】所示。其中，若是模式五能採用整組指派之排班方式，則模式五則可以簡化至模式一。

$$\text{【P}_4\text{】: } \quad \text{Min} \quad \sum_{j=1}^{NP} c_j \times x_j \quad (1) \quad \text{對偶變數}$$

st

$$\sum_{j=1}^{NP} a_{ij} \times x_j = 1, \forall i \quad (2) \quad \pi_i$$

$$x_j \in \{0,1\}, \forall j \quad (3)$$

$$a_{ij} \in \{0,1\}, \forall i, j \quad (4)$$

where $i = 1, 2, \dots, NF$ and $j = 1, 2, \dots, NP$

符號說明如下：

NP：可行勤務組合數目

NF：航段數目

i：第 i 個航段

j：第 j 個可行勤務組合

c_j ：空服員值勤可行勤務組合 j 之成本

x_j ：1,最後的排班班表包含可行勤務組合 j；0,其他

a_{ij} ：1,可行勤務組合 j 涵蓋航段 i；0, 其他

π_i ：航段 i 所對應之對偶變數(dual variable)

$$\text{【P}_5\text{】: } \quad \text{Min} \quad \sum_{k=1}^{NB} \sum_{j=1}^{NP(k)} [c_{jF}^k \times x_{jF}^k + c_{jC}^k \times x_{jC}^k + c_{jY}^k \times x_{jY}^k] \quad (1) \quad \text{對偶}$$

變數

st

$$\sum_{k=1}^{NB} \sum_{j=1}^{NP(k)} a_{ijF}^k \times x_{jF}^k \geq b_{iF}, \quad \forall i \quad (2) \quad \pi_{iF}$$

$$\sum_{k=1}^{NB} \sum_{j=1}^{NP(k)} [a_{ijF}^k \times x_{jF}^k + a_{ijC}^k \times x_{jC}^k] \geq b_{iF} + b_{iC}, \quad \forall i \quad (3) \quad \pi_{iC}$$

$$\sum_{k=1}^{NB} \sum_{j=1}^{NP(k)} [a_{ijF}^k \times x_{jF}^k + a_{ijC}^k \times x_{jC}^k + a_{ijY}^k \times x_{jY}^k] = b_{iF} + b_{iC} + b_{iY}, \quad \forall i \quad (4) \quad \pi_{iY}$$

$$x_{jF}^k, x_{jC}^k, x_{jY}^k \geq 0 \quad x_{jF}^k, x_{jC}^k, x_{jY}^k \in \text{Integer}, \quad \forall j, k \quad (5)$$

$$a_{ijF}^k, a_{ijC}^k, a_{ijY}^k \in \{0,1\}, \quad \forall i, j, k \quad (6)$$

where $i = 1, 2, \dots, NF$ and $j = 1, 2, \dots, NP(k)$

符號說明如下：

NB：航空公司所擁有之組員基地數

NP(k)：由第 k 個基地產生之可行勤務組合數目

NF：航段數目

i：第 i 個航段

j：第 j 個可行勤務組合

c_{jF}^k 、 c_{jC}^k 、 c_{jY}^k ：第 k 個基地中，頭等艙(F)、商務艙(C)、經濟艙(Y)空服員值勤可行勤務組合 j 之成本

x_{jF}^k 、 x_{jC}^k 、 x_{jY}^k ：第 k 個基地中，值勤可行勤務組合 j 所需之頭等艙(F)、商務艙(C)、經濟艙(Y)空服員人數

a_{ijF}^k 、 a_{ijC}^k 、 a_{ijY}^k ：1,第 k 個基地中頭等艙(F)、商務艙(C)、經濟艙(Y)空服員值勤之可行勤務組合 j 涵蓋航段 i ；0,其他

b_{iF} 、 b_{iC} 、 b_{iY} ：第 i 航段所需要之頭等艙(F)、商務艙(C)、經濟艙(Y)空服員人數

π_{iF} 、 π_{iC} 、 π_{iY} ：航段 i_F 、 i_C 、 i_Y 所對應之對偶變數(dual variable)

若排班模式屬單基地之排班模式(如：模式一、二、三、四)，則【P₅】中之組員基地數 NB=1；反之，組員基地數為 NB。若模式屬單機型接續之排班模式(如：模式一、三、五、七)，則【P₅】中之每個可行勤務組合 j 中的每一個航段所使用之機型必需相同；反之，每個航段所使用之機型可以不同。若模式屬單一艙等之排班模式(如：模式一、二、五、六)，則【P₅】中之限制式(2)、(3)不存在，並可簡化成【P₆】；反之，則存在。

$$\text{【P}_6\text{】: } \quad \text{Min} \quad \sum_{k=1}^{NB} \sum_{j=1}^{NP(k)} c_j^k \times x_j^k \quad (1)$$

ST

$$\sum_{k=1}^{NB} \sum_{j=1}^{NP(k)} a_{ij}^k \times x_j^k = b_i, \quad \forall i \quad (2)$$

$$x_j^k \geq 0, x_j^k \in \text{Integer}, \quad \forall j, k \quad (3)$$

$$a_{ij}^k \in \{0,1\}, \quad \forall i, j, k \quad (4)$$

where $i=1,2,...,NF$ and $j=1,2,...,NP(k)$ and $k=1,2,...,NB$

實務上，對於一個大型之國際航空公司而言，由於其複雜的飛航網路與機隊組合，通常使得國際航空公司基於經濟、服務品質等相關因素之考量下，採用模式八為實務之排班模式。然而，由於個案航空公司之排班模式亦採用模式八為其實務運作之排班模式，因此，本研究之主要核心模式即採用模式八這種排班模式。

5.2 排班模式求解架構

本研究提出之排班模式求解架構，係在已知草擬班次表、機隊排程、民航法規、工會規定、航空公司人事派遣規則等條件下，以最小化空服員營運成本為目標，發展結合限制規劃與數學規劃方法之執行架構，來求解航空公司之大規模後艙組員排班問題。

當航空公司之航班越密集、飛航網路越大越複雜，則其基地組員可值勤勤務組合種類與數目亦將越多，若一次產生所有之可行勤務組合再加以求解，將因問題規模過大而會耗費相當長之求解時間，因此，本研究根據限制規劃方法之運作特性，提出兩種結合限制規劃與數學規劃方法之求解架構，亦即以限制規劃為基礎之變數產生法與以限制規劃方法為基礎之限制列舉法，此兩種求解架構在運作上皆有一個共同之特性，即「在產生可行勤務組合時，只考慮本研究所定義之優良可行勤務組合集合」，而此集合之定義方式如圖 5.1 所示。以下就分別針對每一種求解架構作更深入之描述。

5.2.1 以限制規劃為基礎之變數產生法

本研究所提出之變數產生法求解架構與一般文獻所使用之變數產生法求解架構有下列幾點相異處：

1. 本研究之起始解建構模組與子問題求解模組，皆是以勤務組合產生問題限制規劃模式 CSP 作為核心的勤務組合產生模組，而不是以網路模式作為勤務組合產生模組。
2. 以 CP 集合中之 CBP 集合與 CEP_1 子集合作為勤務組合產生之對象，亦即本研究考慮實務之運作狀況只考慮含一個空載之可行勤務組合，而且所考慮之可行勤務範圍完全侷限在所定義之 CP 集合中。
3. 以 CSP-CBP 作為起始解建構模組，亦即本研究不像傳統文獻使用人工方式或是啟發式解法找到一個較不好的 RMP 起始解，而是直接窮舉所有 CBP 集合，找到一個非常好的解作為 RMP 起始解。
4. 以 CSP- CEP_1 +Reduced_Cost 限制式作為子問題求解模組，亦即本研究不像傳統文獻在子問題中一起考慮無空載與含空載之可行勤務組合，而是將子問題視為一個空載選擇模組(Deadhead Selector Module)，透過對偶理論來仔細地、系統地從含空載之勤務組合集合中，選取部分對於整體目標值有貢獻的含空載勤務組合。

本研究所發展出之變數產生法求解架構，除了與文獻所使用之變數產生法求解架構有上述幾點不同外，基本上，其運作方式仍是以變數產生法理論為核心，其整體之求解架構如圖 5.1 示

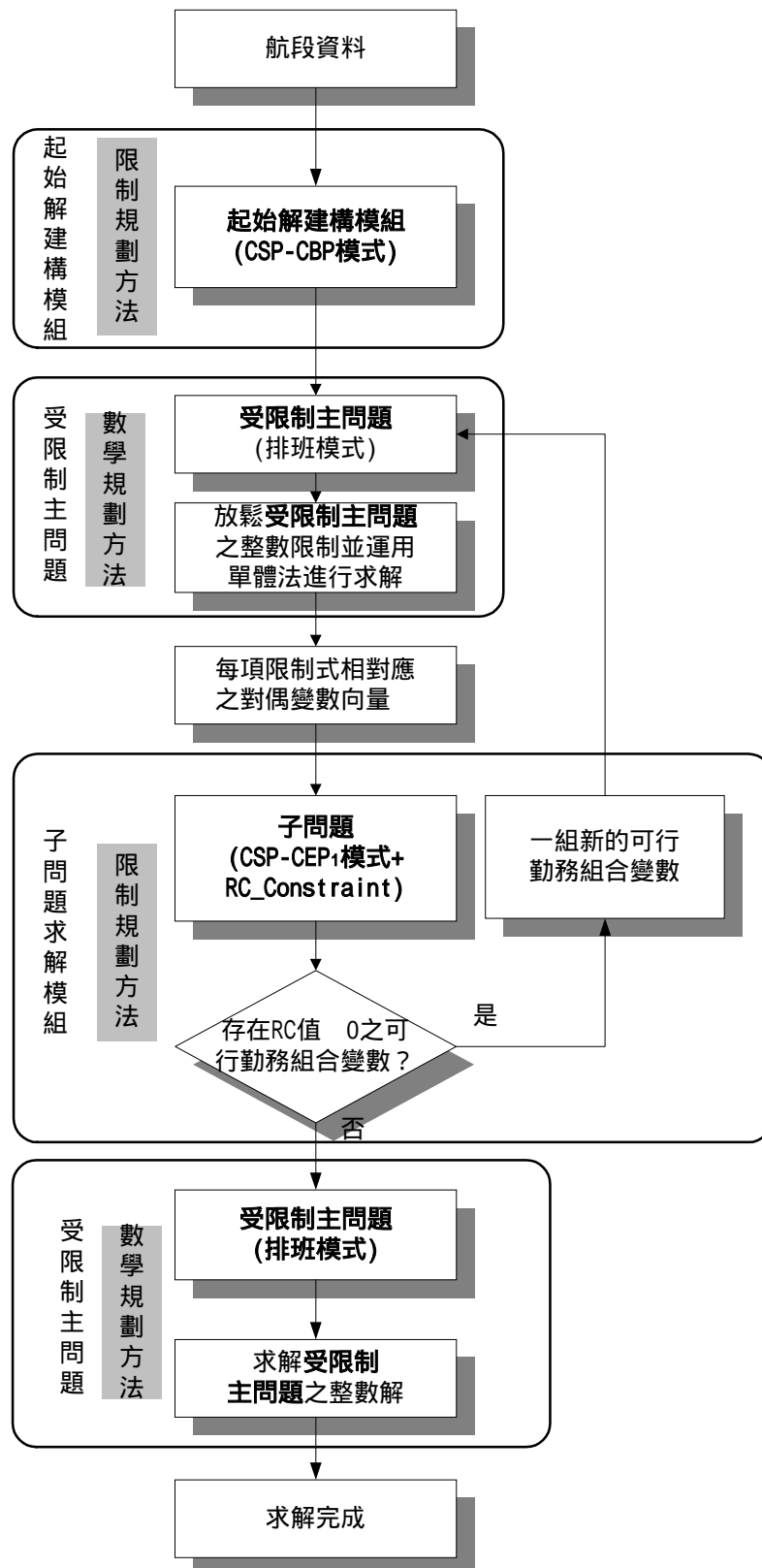


圖 5.1 以限制規劃為基礎之變數產生法求解架構

5.2.1.1 起始解建構模組

起始解建構模組之主要目的在於「快速產生一組能涵蓋所有航段且品質優良之勤務組合，以期提供受限制主問題一個較佳之起始解」，然而，有鑑於本研究所定義之 CBP 集合中，一定保證能涵蓋所有航段，所以本研究將以 CSP-CBP 模式作為起始解建構模組，以 CBP 集合作為起始勤務組合集合。

5.2.1.2 子問題求解模組

文獻上通常將子問題定式成一個最短路徑問題或是資源受限制之最短路徑問題，此種子問題定式之優點在於「每一次的子問題求解，能快速地找到一個對目前受限制主問題目標值最有貢獻之可行勤務組合變數」，缺點在於「每一次的子問題求解，只能提供受限制主問題一個可以改善目標值之新變數」，而此一缺點將使得傳統以網路模式為基礎之變數產生法收斂速度過於緩慢，有鑑於此，本研究將子問題定式成一個勤務組合產生問題限制規劃模式，其定式如【CSP₁】所示，其優點為「可以快速地產生一組改善目前受限制主問題目標值之可行勤務組合變數」。

【CSP₁】： CSP-CEP₁ + [Reduced_Cost 限制式]

根據 5.1 節之排班模式定式(如：【P₄】【P₅】【P₆】)與對偶理論，可將 Reduced_Cost 限制式歸納成【式 5.1】，其中之每個限制式分別表示：

5. 對於 k 基地、F 艙等組員而言，所服務之勤務組合 j 必須符合【式 5.1】中之(1)式，才能改善目前受限制主問題之目標值。
6. 對於 k 基地、C 艙等組員而言，所服務之勤務組合 j 必須符合【式 5.1】中之(2)式，才能改善目前受限制主問題之目標值。
7. 對於 k 基地、Y 艙等組員而言，所服務之勤務組合 j 必須符合【式 5.1】中之(3)式，才能改善目前受限制主問題之目標值。

$$c_{j_F}^{k'} = c_{j_F}^k - \sum_{i \in F(j_F)} a_{ij_F}^k \pi_{i_F} - \sum_{i \in F(j_F)} a_{ij_F}^k \pi_{i_C} - \sum_{i \in F(j_F)} a_{ij_F}^k \pi_{i_Y} \leq 0 \quad (1)$$

$$c_{j_C}^{k'} = c_{j_C}^k - \sum_{i \in F(j_C)} a_{ij_C}^k \pi_{i_C} - \sum_{i \in F(j_C)} a_{ij_C}^k \pi_{i_Y} \leq 0 \quad (2) \quad \text{【式 5.1】}$$

$$c_{j_Y}^{k'} = c_{j_Y}^k - \sum_{i \in F(j_Y)} a_{ij_Y}^k \pi_{i_Y} \leq 0 \quad (3)$$

符號說明：

$c_{j_F}^k$ 、 $c_{j_C}^k$ 、 $c_{j_Y}^k$ ：分別表示在目前受限制主問題中新增勤務組合 j_F 、 j_C 、 j_Y 變數，可讓目標值成本變更多少之懲罰成本(Reduced Cost)

π_{i_F} 、 π_{i_C} 、 π_{i_Y} ：航段 i_F 、 i_C 、 i_Y 所對應之對偶變數(dual variable)

$F(j_F)$ 、 $F(j_C)$ 、 $F(j_Y)$ ：分別表示構成勤務組合 j_F 、 j_C 、 j_Y 之航段集合

對於子問題求解模組所考慮勤務組合對象而言，由於起始解建構模組已經窮舉 CP 集合中之 CBP 集合，所以為了防止產生重複之可行勤務組合，因此子問題求解模組所產生之勤務組合對象必須要是與 CBP 集合互斥之 CP 子集合，然而因為實務運作上很少使用兩個(含)以上空載航段數之勤務組合，而且若是考慮包含兩個(含)以上空載航段數之勤務組合則將使得排班模式之規模迅速擴大，所以本研究乃以 CEP_1 集合作為子問題求解模組所考慮勤務組合對象。

子問題求解模組之求解概念為「以 CP 集合產生架構為基礎，以 CEP_1 集合為子問題求解模組所考慮之勤務組合對象，以 $CSP-CEP_1+Reduced_Cost$ 限制式作為求解模式，從 CEP_1 集合中產生對目前受限制主問題目標式有貢獻之勤務組合」，此概念以圖 5.5 簡單表示。

對於子問題求解模組中所包含之子問題個數而言，由於不同基地、不同艙等組員之「薪資結構」與「考慮之 $Reduced_Cost$ 限制式不同」，使得對於一個具有 m 個基地、 n 個艙等之排班模式而言，其子問題求解模組在每次之變數產生法運算回合中，共包含著 $m \times n$ 個子問題待求解。

5.2.2 限制列舉式之產生-優化法

產生-優化法之核心求解概念乃在於「其將整個組員排班問題分割成兩個獨立之問題，其中一個為負責考慮所有複雜排班法規限制之勤務組合產生問題，稱之為 Operational Sub Problem，另一個為負責考慮成本收益之最小成本勤務組合產生問題，稱之為 Economic Master Problem」，而本研究所構建的「限制列舉式之產生-優化法」(Constrained Enumeration Based Generate-and-Optimize)即是屬於產生-優化法(Generate-and-Optimize)之一種，其間之不同處乃在於「產生-優化法的勤務組合產生方式是使用最傳統之列舉法(Enumeration Method)，如：檢驗-回溯法法(Test-and-Backtrack)或網路模式方法，而限制列舉式之產生-優化法的勤務組合產生方式乃是使用運限制規劃方法來列舉一些品質較優良之可行勤務組合」。

限制列舉式之產生-優化法的運作方式為「先透過一般化勤務組合產生問題限制規劃模式 CSP 來列舉 CBP 與 CEP_1 集合中之所有優良勤務組合，然後再透過整數規劃方法來求解排班模式」，其詳細求解架構如圖 5.2 所示。

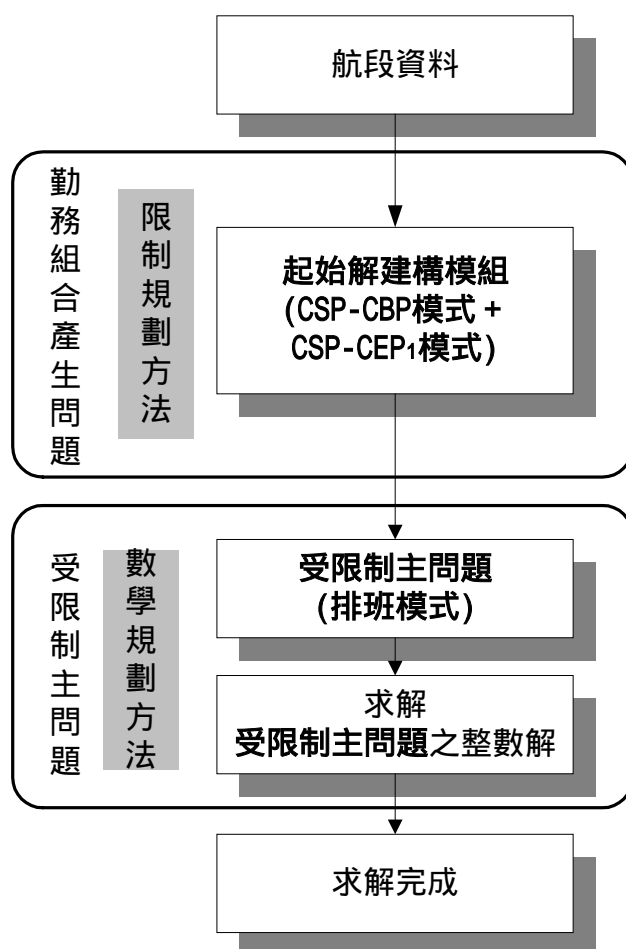


圖 5.2 限制列舉式之產生-優化法求解架構

與以限制規劃為基礎之變數產生法求解架構一樣使用起始解建構模組來「產生一組能涵蓋所有航段且品質優良之勤務組合」，然而不同的是，此架構之起始解建構模組所考慮之優良勤務組合規模必須要比起始解建構模組所考慮之規模還來的大，然而由於實務運作上很少使用兩個(含)以上空載航段數之勤務組合，而且若是考慮包含兩個(含)以上空載航段數之勤務組合則將使得排班模式之規模迅速擴大，所以本研究乃將限制列舉式之產生-優化法所考慮之勤務組合對象限定在 CBP 與 CEP_1 兩個 CP 子集合上。

第六章 實證分析

實證分析之主要目的有二，其一為「測試一般化勤務組合產生問題限制規劃模式在產生勤務組合時之執行績效」，另一為「測試以限制規劃為基礎之變數產生法與限制列舉式之產生-求解法兩種求解架構在實務問題求解上之執行績效」，基於此二個目的，本研究將分別針對最簡單之排班模式(單基地、單機型接續、單艙等)與最複雜之排班模式(多基地、多機型接續、多艙等)進實例行測試。

6.1 測試例題簡介

茲將本研究所使用之測試例題資料整理如表 6.1 所示。

表 6.1 測試例題資料

| 例題名稱 | 資料來源 | 資料說明 | 航段數 | 模式類型 |
|------|------------|---------------|-----|--|
| 例題 1 | 沈志展學長之碩士論文 | 1991 年之華航夏季班表 | 68 | 最簡單之排班模式 (單基地、單機型接續、單艙等) |
| 例題 2 | 湯敦台學長之碩士論文 | 1997 年之華航夏季班表 | 454 | 最複雜之排班模式 (多基地、多機型接續、多艙等) (4 個基地、7 種機型、3 種艙等) |

6.2 排班模式之測試-單基地、單機型接續、單艙等

6.2.1 排班資料

茲將測試例題 1 之詳細資料整理如表 6.2 所示，並分別將此測試例題所使用之勤務段產生規則、勤務組合產生規則、成本資料、勤務組合成本計算公式，依序整理如表 6.3、表 6.4、表 6.5、【式 6.1】所示。

表 6.2 例題資料-例題 1

| | |
|---------|------------------------------|
| 資料來源 | 1991 年之華航夏季班表/沈志展學長之碩士論文[40] |
| 個案公司 | 中華航空公司 |
| 航線種類 | 國際航線 |
| 航線資料 | 台北-香港、台北-東京、台北-洛杉磯三個航線 |
| 航線班次密集度 | 三個航線皆屬航線班次密集度高之航線 |
| 航段數 | 68，細節請參閱附錄一 |
| 艙等資料 | 1 各艙等 |
| 機型資料 | 只有 747-400 型一種機型 |
| 基地資料 | 只有 1 個組員基地-台北 |
| 班次週期 | 以週為一個循環週期 |
| 模式類型 | 單基地、單機型、單一艙等 |

| | |
|--------|--------------------------------|
| 排班對象 | 前艙組員(飛航組員) – 本研究將其視為單艙等之後艙組員排班 |
| 勤務組合型態 | Pairing |

表 6.3 勤務段產生規則-例題 1

| | |
|------|--|
| 規則 1 | 勤務段中，最多可包含四個航段(國際航線) |
| 規則 2 | 勤務段中，各航段所使用之機型必須相同 |
| 規則 3 | 勤務段中，前一航段的飛機到達時間與後一航段的飛機到達時間必須相距在 40 分鐘與 4 小時之間。(兩航段間之最短相距時間取 40 分鐘，乃由於兩航段間必須適當的時間供空勤組員進行下一航段之飛行準備，而目前案例航空公司將此一時間訂為 40 分；而兩航段間之最長相距時間取 4 小時，乃因為案例航空公司規定「若兩航段相距 4 小時以上，則必須將空勤組員送至旅館休息」)，意即，兩連續值勤航段間之最短休息時間必須大於 40 分鐘，最長休息時間不得超過 4 小時。 |
| 規則 4 | 勤務段中，前一航段之飛機降落地點與後一航段之飛機起飛地點不可為該網路的基地 |
| 規則 5 | 勤務段中，總工作時間不可超過 14 小時，總飛行時間不可超過 9 小時 10 分 |

表 6.4 勤務組合產生規則-例題 1

| | | |
|--------------|------|--|
| 飛行時間與工作時間之限制 | 規則 1 | 連續 24 時內，工作時間不得超過十四小時，飛行時間應不超過 9 小時。惟單趟落地航次可酌予延長，但需保留客位或床位不得少於 4 個 |
| | 規則 2 | 若工作時間超過 12 小時，則休息時間不得低於 24 小時 |
| | 規則 3 | 若工作時間超過 8 小時，不到十二小時，休息時間不得低於 12 小時 |
| | 規則 4 | 工作時間 8 小時以內，休息時間不得低於 8 小時 (所謂之工作時間，包括報到時間與報離時間) |
| 報到與報離時間之限制 | | 組員必須在飛機起飛前 1 小時報到，且於飛機落地後 1 小時報離 |

表 6.5 成本資料-例題 1

| 成本項 | 成本計算方式 |
|----------------------------|------------|
| 報到時間之時間(preflight time)成本 | 1 單位成本/分 |
| 報離時間之時間(postflight time)成本 | 1 單位成本/分 |
| 實際飛行之服務時間(block time)成本 | 1 單位成本/分 |
| 等待下一班機之時間(layover time)成本 | 0.4 單位成本/分 |
| 組員在外站之住宿成本 | 100 單位成本/次 |

$$T_{\text{check_in}} * 1 + T_{\text{check_out}} * 1 + T_{\text{fly}} * 1 + T_{\text{fly_DH}} * (1/2) + N_{\text{resident_day}} * 100 + T_{\text{sit}} * 0.4$$

符號說明：

$T_{\text{check_in}}$ ：勤務組合之總報到時間(分)

$T_{\text{check_out}}$ ：勤務組合之總報離時間(分)

【式 6.1】

T_{fly} ：勤務組合中，非空載航段之總飛行時間(分)

$T_{\text{fly_DH}}$ ：勤務組合中，空載航段之總飛行時間(分)

$N_{\text{resident_day}}$ ：勤務組合在外站之外宿天數(天)

T_{sit} ：勤務組合之總休憩時間(分)

6.2.2 問題規模與最佳解

6.2.2.1 問題規模-TP 集合

雖然此測試例題只有 68 個航段，表面上看來此問題之規模並不大，但是由於此問題所考慮勤務組合型態為 Pair，所以透過接續情況之變多，進而將使得此問題之規模變得相當的龐大。

根據例題 1 之相關排班資料與 TP 集合產生架構，來窮舉測試例題 1 之所有可行勤務組合，可發現此問題之規模高達 1 百 70 幾萬個變數，顯示例題 1 之規模實為相當龐大。將窮舉出來的可行勤務組合，依據勤務組合之種類，整理如表 6.6 所示，其中 BP 集合與 EP 集合各佔 TP 集合之比率為 3.9%與 96.1%，從此一數據可推知，對於一般之排班問題而言，BP 集合只佔整個排班問題可行勤務組合總數之一小部分，而大部分之可行勤務組合是包含在 EP 集合當中。

表 6.6 問題規模-TP 集合-例題 1

| Pairing Type | BP 集合 | EP 集合 | | | | | Total |
|--------------|--------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------|
| | | EP ₁ | EP ₂ | EP ₃ | EP ₄ | EP ₅ | |
| Pair2 | 395 | 790 | N/A | N/A | N/A | N/A | 1,185 |
| Pair3 | 0 | 0 | 0 | N/A | N/A | N/A | 0 |
| Pair4 | 4,651 | 18,604 | 18,604 | 0 | N/A | N/A | 41,859 |
| Pair5 | 0 | 0 | 0 | 0 | 0 | N/A | 0 |
| Pair6 | 61,506 | 369,036 | 738,072 | 492,048 | 0 | 0 | 1,660,662 |
| Total | 66,552 | 388,430 | 756,676 | 492,048 | 0 | 0 | 1,703,706 |

6.2.2.2 問題規模-CP 集合

根據例題 1 之相關排班資料以及 CP 集合產生架構，來窮舉本研究所定義之 CP 集合，並將其結果整理如表 6.7 所示。相互比較 TP 集合與 CP 集合之規模，可發現 CP 集合只佔 TP 集合之 3.59%，可實證 CP 集合產生架構對於降低問題規模之重要性。

表 6.7 問題規模-CP 集合-例題 1

| Pairing Type | CBP 集合 | CEP 集合 | | | | | Total |
|--------------|--------|------------------|------------------|------------------|------------------|------------------|--------|
| | | CEP ₁ | CEP ₂ | CEP ₃ | CEP ₄ | CEP ₅ | |
| Pair2 | 119 | 238 | N/A | N/A | N/A | N/A | 357 |
| Pair3 | 0 | 0 | 0 | N/A | N/A | N/A | 0 |
| Pair4 | 477 | 1,908 | 1,908 | 0 | N/A | N/A | 4,293 |
| Pair5 | 0 | 0 | 0 | 0 | 0 | N/A | 0 |
| Pair6 | 2,094 | 12,564 | 25,128 | 16,752 | 0 | 0 | 56,538 |
| Total | 2,690 | 14,710 | 27,036 | 16,752 | 0 | 0 | 61,188 |

6.2.2.3 問題之最佳解

由於例題 1 之 TP 集合規模太大(1 百 70 幾萬個變數)，所以絕不可能在有限時間與記憶體限制之情況下求得例題之最佳解，然而，由於 CP 集合之規模不大(6 萬多個變數)，所以能在有限時間與記憶體限制之情況下求得最佳解，經過實際求解之後，本研究發現 CP 集合中之最佳解為 28,451，雖然本研究無法保證此解為例題 1 之最佳解，但本研究確信此解為一個相當接近最佳解之近似最佳解。

6.2.3 測試結果與分析

將測試例題 1 之航段資料、法規資料、成本資料輸入本研究所建立之一般化勤務組合產生問題限制規劃模式，並採用單基地、單機型接續、單艙等之排班模式(模式一)處理之，最後運用本研究所構建之兩種求解架構進行測試，並歸納測試結果於表 6.8。

表 6.8 測試結果-例題 1

| 分析項目 \ 求解架構 | | 以限制規劃 為基礎之變數產生法 | 限制列舉式之 產生-優化法 |
|-------------|--------------|--------------------|------------------|
| 起始解 | 受限制主問題之變數個數 | 2,690 | 17,400 |
| | 受限制主問題之限制式個數 | 68 | 68 |
| | 目標值 | 29,181(IP) | 28,451(IP) |
| | CP 求解時間(秒) | 5.849 | 10.395 |
| | LP 求解時間(秒) | 0.44 | 1.773 |
| 子問題 | 新增變數個數 | 519 | N/A |
| | 總運算回合數 | 2 | N/A |
| | LP 目標值 | 28,580 | N/A |
| | 目標值改善百分比 | 2.50% | N/A |
| | 運算時間(秒) | 15.813 | N/A |
| 整數 最佳解 | 受限制主問題之變數個數 | 3,209 | 17,400 |
| | IP 目標值 | 28,451 | 28,451 |
| | 整數解變數個數 | 23 | 23 |
| | 含空載之解 | 10 | 10 |
| | IP 求解時間(秒) | 1.792 | 3.575 |
| 總運算時間(秒) | | 23.894 | 15.743 |

分析一：求解架構間之求解結果與執行效率

彙整比較「以限制規劃為基礎之變數產生法」與「限制列舉式之產生-優化法」兩種求解架構之求解結果與執行效率，相關細節如表 6.9 所示。

表 6.9 兩種求解架構間之求解結果與執行績效分析-例題 1

| 分析項目 | (1) 以限制規劃為基礎之變數產生法 | (2) 限制列舉式之產生-優化法 | 比較分析 求解架構(1)與(2) |
|-------------|-----------------------|---------------------|--|
| 受限制主問題之變數個數 | 3,209 | 17,400 | 由於(2)窮舉了 CBP 與 CEP_1 兩個集合，而(1)只窮舉了 CBP 與部分之 CEP_1 集合，使得(1)所考慮之變數集合為(2)所考慮變數之一個子集，所以(2)所使用之變數會較(1)多 |
| LP 目標值 | 28,580 | 28,451 | 由於(1)所考慮之變數集合為(2)的一個子集，所以(2)之 LP 最佳解一定會小於等於測試結構(1)之 LP 最佳解 |
| IP 目標值 | 28,451 | 28,451 | 如同上述，由於(1)所考慮之變數集合為(2)的一個子集，所以(2)之 IP 最佳解一定會小於等於測試結構(1)之 IP 最佳解 |
| 總運算時間(秒) | 23.894 | 15.743 | 由於(2)是一次窮舉 CBP 與 CEP_1 兩個集合，而(1)是一次窮舉 CBP 集合與二次(子問題運算回合)部分列舉 CEP_1 集合，使得(1)一共花了三次的限制式檢驗時間，而(2)只花一次而已，所以在 IP 求解時間差距不大之情況下，(2)之總運算時間將較(1)來的少 |

分析二：本研究最佳解與文獻最佳解之比較

由於文獻上之最佳解只考慮部分之無空載可行勤務組合，因此基於公平性之考量，本研究將以 CP 集合中之最佳解來與文獻進行比較，相關細節如表 6.10 所示。

表 6.10 本研究最佳解與文獻最佳解之比較分析-例題 1

| 分析項目 | 文獻最佳解 | 本研究 | | 比較分析 |
|------------------------|--------------|-------------|---------------------------------|---|
| 所使用之變數 | 256 | (0) | 2690 (10.5) | 由於文獻在設計其檢驗-回溯法之電腦邏輯程式時,並沒有完全考慮所有可能之航段接續情況,所以文獻只產生一部份無空載可行勤務組合而已,而本研究乃是運用一般化勤務組合產生問題限制規劃模式來產生行勤務組合,能考慮各種接續情況之可行勤務組合,所以本研究所產生之可行勤務組合會比文獻上所產生之可行勤務組合來的多。此一結果,亦可充分地證實本研究建構的一般化勤務組合產生問題限制規劃模式能求解各種接續情況之勤務組合。 |
| | | (1) | 3,209 (12.5) | |
| | | (2) | 17,400 (68) | |
| 最佳解 | 29,337 | (0) | 29,181 (0.53%) | 由於 CBP 集合中之最佳解已優於文獻最佳解,所以當 (2)與(1)皆多考慮 CEP ₁ 集合中之勤務組合時,將使得(2)與(1)之最佳解一定優於(0)之最佳解。此一結果,亦充分地顯示本研究所定義之 CBP 集合實能考慮各種接續情況之優良勤務組合集合。 |
| | | (1) | 28,451 (3.02%) | |
| | | (2) | 28,451 (3.02%) | |
| 勤務組合產生方式 | 檢驗-回溯法 | 限制規劃方法 | | N/A |
| 求解架構 | 產生-優化法 | (0)、(1)、(2) | | N/A |
| 勤務組合對象 | 部分無空載之可行勤務組合 | (0) | CBP 集合 | 文獻中所考慮之無空載勤務組合範圍受限於文獻之檢驗-回溯法設計 |
| | | (1) | CBP 集合與一部份之 CEP ₁ 集合 | |
| | | (2) | CBP 集合與 CEP ₁ 集合 | |
| 求解架構： | | | | |
| (0) - 求解架構(1)中之起始解建構模組 | | | | |
| (1) - 以限制規劃為基礎之變數產生法 | | | | |
| (2) - 限制列舉式之產生-優化法 | | | | |

6.3 排班模式之測試-多基地、多機型接續、多艙等

6.3.1 排班資料

茲將測試例題 2 之詳細資料整理如表 6.11 所示，並分別將此測試例題所使用之勤務段產生規則、勤務組合產生規則、成本資料、勤務組合計算公式，依序整理如表 6.12、表 6.13、表 6.14、【式 6.2】所示，。

表 6.11 例題資料-例題 2

| | |
|---------|------------------------------|
| 資料來源 | 1997 年之華航夏季班表/湯敦台學長之碩士論文[42] |
| 個案公司 | 中華航空公司 |
| 航線種類 | 國際航線 |
| 航線資料 | 共有 19 條航線，細節如表 6.15 所示 |
| 航線班次密集度 | 包含航線班次密集度高、中、低之航線 |
| 航段數 | 454，細節請參閱附錄二 |
| 艙等資料 | 頭等艙(F)、商務艙(C)、經濟艙(Y) |
| 機型資料 | 共有 7 種機型，細節如表 6.16 所示 |
| 組員需求資料 | 各機型需各艙等組員之人數資料如表 6.17 所示 |
| 基地資料 | 共有 4 個組員基地-台北、東京、曼谷、新加坡 |
| 班次週期 | 以週為一個循環週期 |
| 模式類型 | 多基地、多機型、多艙等 |
| 排班對象 | 後艙組員(空服員) |
| 勤務組合型態 | Pairing |

表 6.12 勤務段產生規則-例題 2

| | |
|------|--|
| 規則 1 | 勤務段中，最多可包含四個航段(國際航線) |
| 規則 2 | 勤務段中，各航段所使用之機型必須相同 |
| 規則 3 | 勤務段中，前一航段的飛機到達時間與後一航段的飛機到達時間必須相距在 40 分鐘與 4 小時之間。(兩航段間之最短相距時間取 40 分鐘，乃由於兩航段間必須適當的時間供空勤組員進行下一航段之飛行準備，而目前案例航空公司將此一時間訂為 40 分；而兩航段間之最長相距時間取 4 小時，乃因為案例航空公司規定「若兩航段相距 4 小時以上，則必須將空勤組員送至旅館休息」)，意即，兩連續值勤航段間之最短休息時間必須大於 40 分鐘，最長休息時間不得超過 4 小時。 |
| 規則 4 | 勤務段中，前一航段之飛機降落地點與後一航段之飛機起飛地點不可為該網路的基地 |
| 規則 5 | 勤務段中，總工作時間不可超過 14 小時，總飛行時間不可超過 9 小時 |

表 6.13 勤務組合產生規則-例題 2

| | | |
|----------------------|------|--|
| 飛行時間 與工作時間之 限制 | 規則 1 | 連續 24 時內，工作時間不得超過十四小時，飛行時間應不超過 9 小時。惟單趟落地航次可酌予延長，但需保留客位或床位不得少於 4 個 |
| | 規則 2 | 若工作時間超過 12 小時，則休息時間不得低於 24 小時 |
| | 規則 3 | 若工作時間超過 8 小時，不到十二小時，休息時間不得低於 12 小時 |
| | 規則 4 | 工作時間 8 小時以內，休息時間不得低於 8 小時 (所謂之工作時間，包括報到時間與報離時間) |
| 報到與報離時間之限制 | | 組員必須在飛機起飛前 2 小時 20 分報到，且於飛機落地後 1 小時報離 |

表 6.14 成本資料-例題 2

| 成本項 | 成本計算方式 |
|-------------------------|--------------|
| 外站零用金之時間成本 | 60 元/時 |
| 實際飛行之服務時間(block time)成本 | 細節如表 6.17 所示 |
| 組員在外站之住宿成本 | 2100 元/天 |

表 6.15 航線資料-例題 2

| 洲航線別 | 區域航線別 | 單航線別 | 航線名稱 |
|------|-------|---------|---------------------------|
| 亞洲航線 | 日本航線 | 東京航線 | TPE-TYO、TPE-TYO-HNL 航線 |
| | | 名古屋航線 | TPE-NGO 航線 |
| | | 福岡航線 | TPE-FUK 航線 |
| | | 琉球航線 | TPE-OKA 航線 |
| | 大陸航線 | 香港航線 | TPE-HKG 航線 |
| | 東南亞航線 | 新加坡航線 | TPE-SIN 航線 |
| | | 曼谷航線 | TPE-BKK 航線、TPE-BKK-AMS 航線 |
| | | 巴里島航線 | TPE-DPS 航線 |
| | | 普吉島航線 | TPE-HKT 航線 |
| | | 雅加達航線 | TPE-JKT 航線 |
| | | 胡志明航線 | TPE-SGN 航線 |
| | | 馬尼拉航線 | TPE-MNL 航線 |
| | | 吉隆坡航線 | TPE-KUL 航線、TPE-KUL-FRA 航線 |
| 美洲航線 | 美國航線 | 檀香山航線 | TPE-TYO-HNL 航線、TPE-HNL 航線 |
| | | 舊金山航線 | TPE-SFO 航線 |
| | | 紐約航線 | TPE-ANC-NYC 航線 |
| | | 洛杉磯航線 | TPE-LAX 航線 |
| 歐洲航線 | 荷蘭航線 | 阿姆斯特丹航線 | TPE-BKK-AMS 航線 |
| | 法國航線 | 法蘭克福航線 | TPE-KUL-FRA 航線 |

表 6.16 機型資料-例題 2

| 代號 | 機型 | 代碼 |
|----|----------------|-----|
| 1 | Boeing 737-400 | 734 |
| 2 | Boeing 747-SP | 74L |
| 3 | Airbus A300 | AB3 |
| 4 | Airbus A600 | AB6 |
| 5 | MD-11 | M11 |
| 6 | Boeing 747-400 | 744 |
| 7 | Boeing 747-200 | 742 |

表 6.17 各機型需各艙等組員之人數資料-例題 2

| 機型 \ 空服員 | F 資格 | C 資格 | Y 資格 | Total |
|----------|------|------|------|-------|
| 734 | 0 | 1 | 3 | 4 |
| 74L | 0 | 3 | 9 | 12 |
| AB3 | 0 | 3 | 8 | 11 |
| AB6 | 0 | 3 | 8 | 11 |
| M11 | 3 | 3 | 7 | 13 |
| 744 | 3 | 3 | 12 | 18 |
| 742 | 3 | 3 | 9 | 15 |

表 6.18 各基地、各艙等組員之平均飛時薪資成本資料-例題 2

| 平均飛時薪資(元) \ 基地 | 台北 | 東京 | 曼谷 | 新加坡 |
|----------------|-----|-----|-----|-----|
| F 資格空服員 | 988 | 920 | 663 | 685 |
| C 資格空服員 | 823 | 767 | 556 | 574 |
| Y 資格空服員 | 719 | 671 | 488 | 501 |

$$T_{\text{check_in}} * 60 + T_{\text{check_out}} * 60 + T_{\text{duty}} * C_j - T_{\text{fly_DH}} * C_j (1/2) + N_{\text{resident_day}} * 2100 + T_{\text{rest}} * 60$$

符號說明：

$T_{\text{check_in}}$ ：勤務組合之總報到時間(時)

$T_{\text{check_out}}$ ：勤務組合之總報離時間(時)

T_{duty} ：勤務組合中，所有勤務之總飛行時間(時)

$T_{\text{fly_DH}}$ ：勤務組合中，空載航段之總飛行時間(時)

C_j ：j 艙等組員之飛行時薪薪資(元/時)

$N_{\text{resident_day}}$ ：勤務組合在外站之外宿天數(天)

T_{rest} ：勤務組合之總休息時間(時)

【式 6.2】

6.3.2 問題規模與最佳解

6.3.2.1 問題規模-TP 集合

測試例題 2 共包含 454 個航段，與例題 1 不同之地方在於，此例題所考慮勤務組合型態為 Pair、並考慮多基地、多艙等，所以此例題之規模一定較例題 1 更為龐大、更為驚人。

根據例題 2 之相關排班資料與 TP 集合產生架構，來窮舉測試例題 2 之所有可行勤務組合集合(TP 集合)，可發現此問題之 TP 集合規模高達 6 仟萬個變數(若不考慮多艙等因素，問題之規模為 2 仟萬個變數)，顯示例題 2 之問題規模實為相當龐大。將窮舉出來的所有可行勤務組合(不分艙等)，依據勤務組合之種類，整理如表 6.19 所示，其中 BP 集合與 EP 集合各佔總可行勤務組合之比率為 1.7%與 98.3%，從此一數據可推知，一般之排班問題中，BP 集合只佔整個排班問題可行勤務組合總數之一小部分，而大部分之可行勤務組合乃是包含在 EP 集合當中。

表 6.19 問題規模-TP 集合(單艙等)-例題 2

| 基地別 | Pairing Type | BP 集合 | | | EP 集合 | | | | | 單艙等 Total |
|-----------|--------------|--|---------------------------------------|-------------------|-----------------|-----------------|-----------------|-----------------|-----------------|------------|
| | | {F _H 、 F _M F _L } | {F _M 、 F _L } | {F _L } | EP ₁ | EP ₂ | EP ₃ | EP ₄ | EP ₅ | |
| TPE | Pair2 | 4,243 | 117 | 9 | 8,738 | N/A | N/A | N/A | N/A | 8,738 |
| | Pair3 | 430 | 96 | 0 | 1,578 | 1,578 | N/A | N/A | N/A | 3,156 |
| | Pair4 | 14,293 | 1,305 | 0 | 62,392 | 93,588 | 62,392 | N/A | N/A | 218,372 |
| | Pair5 | 7,868 | 1,699 | 3 | 47,850 | 95,700 | 95,700 | 47,850 | N/A | 287,100 |
| | Pair6 | 281,621 | 25,001 | 0 | 1,839,732 | 4,599,330 | 6,132,440 | 4,599,330 | 1,839,732 | 19,010,564 |
| | Total | 308,455 | 28,218 | 12 | 1,960,290 | 4,790,196 | 6,290,532 | 4,647,180 | 1,839,732 | 19,864,615 |
| TYO | Pair2 | 418 | 42 | N/A | 920 | N/A | N/A | N/A | N/A | 1,380 |
| | Total | 418 | 42 | 0 | 920 | 0 | 0 | 0 | 0 | 1,380 |
| BKK | Pair2 | 366 | 61 | N/A | 854 | N/A | N/A | N/A | N/A | 1,281 |
| | Total | 366 | 61 | 0 | 854 | 0 | 0 | 0 | 0 | 1,281 |
| SIN | Pair2 | 154 | N/A | N/A | 308 | N/A | N/A | N/A | N/A | 462 |
| | Total | 154 | 0 | 0 | 308 | 0 | 0 | 0 | 0 | 462 |
| 單艙等 Total | | 309,393 | 28,321 | 12 | 1,962,372 | 4,790,196 | 6,290,532 | 4,647,180 | 1,839,732 | 19,867,738 |

6.3.2.2 問題規模-CP 集合

根據例題 2 之相關排班資料以及 CP 集合產生架構，來窮舉本研究所定義之 CP 集合，並將其結果整理如表 6.20 所示。相互比較 TP 集合與 CP 集合之規模，可發現 CP 集合只佔 TP 集合之 0.36%，可實證 CP 集合產生架構對於降低問題規模之重要性。

表 6.20 問題規模-CP 集合(單艙等)-例題 2

| 基地別 | Pairing Type | CBP 集合 | | | CEP 集合 | | | | | 單艙等 Total |
|-----------|--------------|---------------------|----------------|-----------|------------------|------------------|------------------|------------------|------------------|-----------|
| | | $\{F_H, F_M, F_L\}$ | $\{F_M, F_L\}$ | $\{F_L\}$ | CEP ₁ | CEP ₂ | CEP ₃ | CEP ₄ | CEP ₅ | |
| TPE | Pair2 | 1,157 | 72 | 9 | 2,476 | N/A | N/A | N/A | N/A | 3,714 |
| | Pair3 | 19 | 28 | 0 | 141 | 141 | N/A | N/A | N/A | 282 |
| | Pair4 | 85 | 161 | 0 | 984 | 1,476 | 984 | N/A | N/A | 3,444 |
| | Pair5 | 12 | 124 | 3 | 695 | 1,390 | 1,390 | 695 | N/A | 4,170 |
| | Pair6 | 50 | 884 | 0 | 5,604 | 14,010 | 18,680 | 14,010 | 5,604 | 57,908 |
| | Total | 1,323 | 1,269 | 12 | 9,900 | 17,017 | 21,054 | 14,705 | 5,604 | 70,884 |
| TYO | Pair2 | 122 | 24 | 0 | 292 | N/A | N/A | N/A | N/A | 438 |
| | Total | 122 | 24 | 0 | 292 | 0 | 0 | 0 | 0 | 438 |
| BKK | Pair2 | 107 | 30 | 0 | 274 | N/A | N/A | N/A | N/A | 411 |
| | Total | 107 | 30 | 0 | 274 | 0 | 0 | 0 | 0 | 411 |
| SIN | Pair2 | 42 | 0 | 0 | 84 | N/A | N/A | N/A | N/A | 126 |
| | Total | 42 | 0 | 0 | 84 | 0 | 0 | 0 | 0 | 126 |
| 單艙等 Total | | 1,594 | 1,323 | 12 | 10,550 | 17,017 | 21,054 | 14,705 | 5,604 | 71,859 |

6.3.2.3 問題之最佳解

由於例題 2 之勤務組合集合之規模太大(約 6 仟萬個變數)，所以不可能在有限時間與記憶體限制之情況下求得最佳解，然而，由於 CP 集合之規模不大(21 萬個變數)，所以可在有限時間與記憶體限制之情況下求得最佳解，經過實際求解發現，CP 集合中之其值為 29,542,956，雖然本研究無法保證此解為例題 2 之最佳解，但本研究確信此解為一個相當接近最佳解之近似最佳解。

6.3.3 測試結果與分析

將測試例題 2 之航段資料、法規資料、成本資料輸入本研究所建立之一般化勤務組合產生問題限制規劃模式，並採用多基地、多機型接續、多艙等之排班模式(模式八)處理之，最後運用本研究所構建之兩種求解架構進行測試，並歸納測試結果於表 6.21。

表 6.21 測試結果-例題 2

| 分析項目 \ 求解架構 | | 以限制規劃 為基礎之變數產生法 | 限制列舉式之 產生-優化法 |
|-------------|--------------|--------------------|-------------------|
| 起始解 | 受限制主問題之變數個數 | 8,787 | 40,437 |
| | 受限制主問題之限制式個數 | 1,362 | 1,362 |
| | 目標值 | 29,815,010(IP) | 29,541,477.5 |
| | CP 求解時間(秒) | 24.376 | 31.445 |
| | LP 求解時間(秒) | 2.473 | 7 |
| 子問題 | 新增變數個數 | 522 | N/A |
| | 總運算回合數 | 3 | N/A |
| | LP 目標值 | 29,541,477.5 | N/A |
| | 目標值改善百分比 | 0.92% | N/A |
| | 運算時間(秒) | 213.156 | N/A |
| 整數 最佳解 | 受限制主問題之變數個數 | 9,309 | 40,437 |
| | IP 目標值 | 29,542,956 | 29,542,956 |
| | 整數解變數個數 | 567 | 567 |
| | 含空載之解 | 20 | 20 |
| | IP 求解時間(秒) | 6.63 | 58.474 |
| 總運算時間(秒) | | 203.161 | 96.919 |

分析一：求解架構間之求解結果與執行效率

彙整比較「以限制規劃為基礎之變數產生法」與「限制列舉式之產生-優化法」兩種求解架構之求解結果與執行效率，相關細節如表 6.22 所示。

表 6.22 兩種求解架構間之求解結果與執行績效分析-例題 2

| 分析項目 | (1) 以限制規劃為基礎之變數產生法 | (2) 限制列舉式之產生-優化法 | 比較分析 求解架構(1)與(2) |
|-------------|-----------------------|---------------------|--|
| 受限制主問題之變數個數 | 9,309 | 40,437 | 由於(2)窮舉了 CBP 與 CEP_1 兩個集合，而(1)只窮舉了 CBP 與部分之 CEP_1 集合，使得(1)所考慮之變數集合為(2)所考慮變數之一個子集，所以(2)所使用之變數會較(1)多 |
| LP 目標值 | 29,541,477.5 | 29,541,477.5 | 由於(1)所考慮之變數集合為(2)的一個子集，所以(2)之 LP 最佳解一定會小於等於測試結構(1)之 LP 最佳解 |
| IP 目標值 | 29,542,956 | 29,542,956 | 如同上述，由於(1)所考慮之變數集合為(2)的一個子集，所以(2)之 IP 最佳解一定會小於等於測試結構(1)之 IP 最佳解 |
| 總運算時間(秒) | 203.161 | 96.919 | 由於(2)是一次窮舉 CBP 與 CEP_1 兩個集合，而(1)是一次窮舉 CBP 集合與二次(子問題運算回合)部分列舉 CEP_1 集合，使得(1)一共花了三次的限制式檢驗時間，而(2)只花一次而已，所以在 IP 求解時間差距不大之情況下，(2)之總運算時間將較(1)來的少 |

分析二：本研究最佳解與文獻最佳解之比較

茲將比較項目與結果整理如表 6.23 所示。

表 6.23 本研究最佳解與文獻最佳解之比較分析-例題 2

| 分析項目 | 文獻 (1) | 本研究 (2) | 本研究 (3) | 比較分析 |
|---|-----------------------------|-------------------------------------|--------------------------------|---|
| 使用之變數 | 2,007 | 9,309 (4.64) | 40,437 (20.15) | 由於(2)在起始解建構模組中,即窮舉所有 CBP 集合,而(3)則是窮舉所有 CBP 與 CEP ₁ 集合,所以本研究所使用之變數個數一定會較文獻來的多。 |
| 子問題之運算 回合數 | 312 | 3 | N/A | 由於(2)之起始解就已經比文獻之最佳解好,所以可證實本研究之所定義之 CP 集合實能考慮各種接續狀況之優良勤務組合。然而,基於擁有一個非常好的解來作為 RMP 之起始解,再加上本研究在子問題之求解機制為一次產生多組對目標值有貢獻之可行勤務組,使得本研究之變數產生法求解架構其收斂速度較比文獻來的快很多。 |
| 最佳解 | 30,349,381 | 29,542,956 (2.657%) | 29,542,956 (2.657%) | 由於本研究所考慮之勤務組合較文獻多,而且文獻所使用之變數數皆在本研究所使用之變數集合當中,所以本研究之最佳解一定會較文獻來的好。 |
| 勤務組合 產生方式 | 運用最短路徑法 來求解排班網路 | 限制列舉法 | 限制列舉法 | N/A |
| 求解架構 | 以網路模式為基礎之變數產生法 | 以限制規劃為基礎之變數產生法 | 限制列舉式之產生-優化法 | N/A |
| 勤務組合對象 | 同時考慮無空載 與有空載之可能 之勤務組合 | CBP 集合與一部份 之 CEP ₁ 集合 | CBP 集合與 CEP ₁ 集合 | 文獻中所考慮之勤務組合範圍受排班網路之設計 |
| (1)-文獻：以網路模式為基礎之變數產生法 (2)-本研究：以限制規劃為基礎之變數產生法 (3)-本研究：限制列舉式之產生-優化法 | | | | |

文獻所考慮之勤務組合 v.s. 本研究所考慮之勤務組合

比較文獻所考慮之勤務組合和本研究之相異處，並將其歸納成以下兩點來描述：

1. 所考慮之接續情況不同：

深入探究文獻所建構之排班網路，可發現文獻中所建構之排班網路，在有些情況下，少考慮了一些勤務接續情況，以 TPE-ANC-NYC 航線來說明，文獻就少考慮了圖中部分(2)之接續情況，而本研究確是完整考慮各種接續狀況。除此之外，文獻所提之排班網路建構方式亦少考慮了某些可行之接續節線，因而使得某些不錯之可行勤務組合完全不會被考慮到。

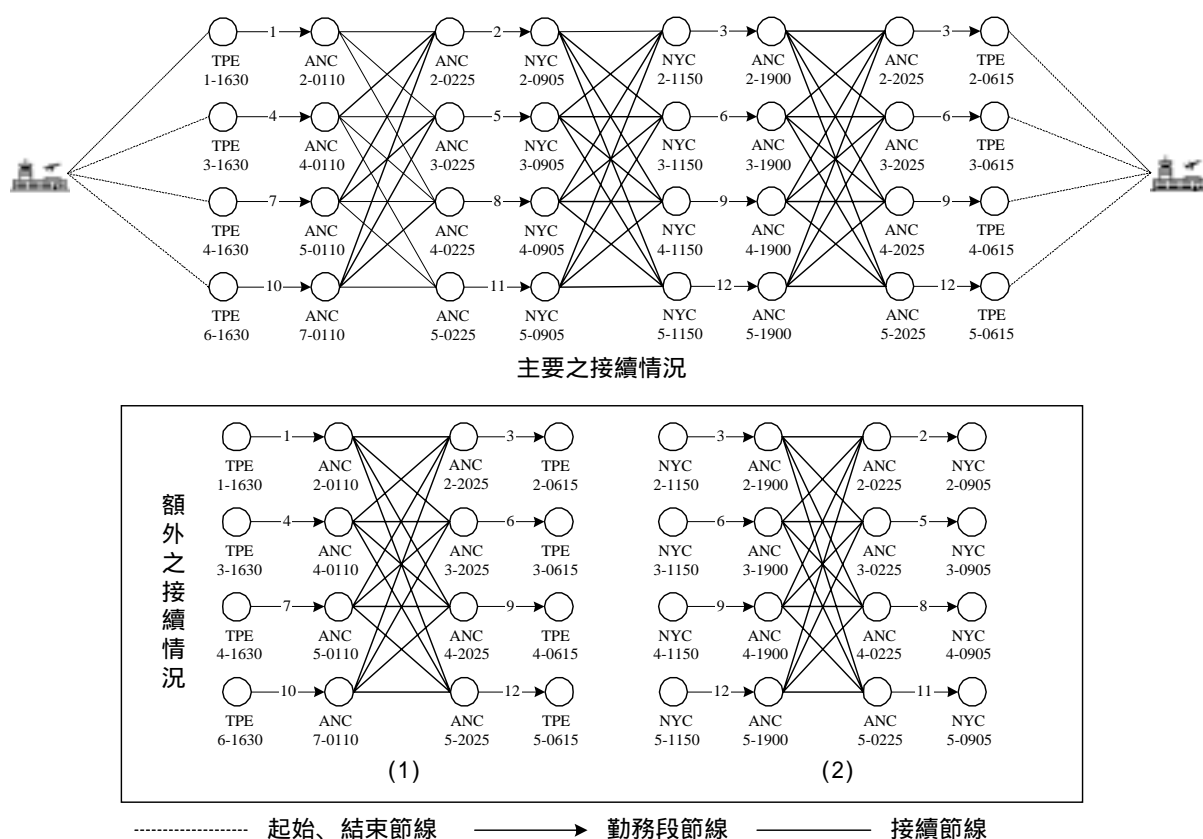


圖 6.1 TPE-ANC-NYC 航線之排班網路

2. 所考慮之勤務類型不同：

分析例題 2 之排班資料，可發現此問題之勤務類型只包含 Duty1 與 Duty2 二種。根據此兩種勤務類型，比較文獻與本研究所考慮之勤務種類與個數之不同，並將其整理如表 6.24 所示，其中可發現，對於 Duty1 勤務而言，文獻與本研究所考慮之個數皆為 454 個，對於同機型接續之 Duty2 勤務而言，文獻比本研究少考慮一個勤務，對於不同機型接續之 Duty2 勤務而言，文獻完全沒有考慮此種類型之勤務，而本研究則是考慮 35 個勤務。因此，整體而言，本研究所考慮之勤務個數為 172 個，而文獻所考慮之勤務個數為 136 個。

表 6.24 文獻與本研究所考慮之勤務類型-例題 2

| Duty 類型 | 湯敦台學長 | 本研究 |
|--------------|-------|-----|
| Duty1 | 454 | 454 |
| Duty2-同機型接續 | 136 | 137 |
| Duty2-不同機型接續 | 0 | 35 |
| 總計 | 136 | 172 |

第七章 結論與建議

綜觀國外之組員排班相關文獻，就可行勤務組合之產生方式而言，雖然已經有部分學者開始運用限制規劃來產生空勤組員之可行勤務組合，但其發表之文獻當中卻沒詳細地指出其所建構出來之勤務組合產生問題限制規劃模式；就勤務組合最佳化之求解方式而言，雖然目前亦已有部分學者開始結合限制規劃與數學規劃於求解相關之組員排班問題，但其發表之文獻之中亦沒有詳細地指出要如何才能最有效地結合此兩方法論於求解組員排班問題。基於上述二項研究動機，本研究嘗試利用限制規劃之運作特性與優點，提出一個一般化之勤務組合產生問題限制規劃模式與一個結合限制規劃與數學規劃之變數產生法求解架構，以供國內外未來相關研究之參考

本研究構建之一般化勤務組合產生問題限制規劃模式能同時考慮各種排班限制(如：工時、飛時、休時)與排班因素(如：組員基地、機型接續、艙等、空載)，非常具有彈性，可確實有效地輔助排班人員產生符合其需求之可行勤務組合，尤其隨著飛航網路之日益擴大與複雜化時，此模式亦能有效地產生各種型態之可行勤務組合。另外，本研究提出的兩種求解架構(以限制規劃為基礎之變數產生法與限制列舉式之產生-優化法)，皆充分地運用限制規劃與數學規劃之特性與優點，實為二種非常有效率之求解架構，可確實提供國內外學術界與實務界之參考。根據上一章之實證分析，本研究歸納出數點結論與建議，並分述如下：

7.1 結論

經本研究之實證分析後，獲得以下幾點結論：

結論一：針對國際航線之後艙空勤組員排班問題而言，本研究所定義之 *CP 集合產生架構* 所產生之優良勤務組合，實能大規模地降低問題規模且不失解的品質。

結論二：一般化之勤務組合產生問題限制規劃模式之實用價值相當高

1. 就 模式彈性 而言：相當具有彈性，除可輔助排班人員產生任何其所需要之可行勤務組合類型(如：航段間接續時間較短之勤務組合、完全不包含空載之勤務組合、只包含 1 個空載之勤務組合、某特定航線之所有可行勤務組合等)，亦可輔助各種排班模式所需要之可型勤務組合類型(如：某基地、某機型接續、某艙等組員資格所能服務之勤務組合)。
2. 就 限制式管理之透明度 而言：完全將限制式透明化以方便進行限制式管理。
3. 就 限制式管理之方便性 而言：可隨時地、動態地對限制式進行管理(新增、刪除、修改等)。
4. 就 產生可行勤務組合之效率 而言：較一般常用的窮舉法、網路模式方法來的有效率。

結論三：起始解建構模組之求解績效相當好

1. 一般而言，文獻上所提之起始解建構模組，乃為「運用人工或啟發式方法，來產生一組不管品質優劣但涵蓋所有航段之起始勤務組合」，而本研究基於限制規劃之運作特性，建立出一個能快速產生所有品質優良且涵蓋所有航段之勤務組合，且經實證結果顯示，在無須考慮任何空載之情況下，運用本研究所提出之起始解建構模組所產生出來之起始解品質皆已比文獻上之最佳解佳，充分地顯示本研究之起始解建構模組之求解績效。
2. 本研究所提出之起始解建構模組除了可作為變數產生法架構中之起始解產生模組之外，還能作為限制列舉式之產生-優化法的勤務組合產生產生模組，模組之功能彈性相當大。

結論四：以限制規劃為基礎之變數產生法求解架構之求解績效相當好

1. 變數產生法之收斂速度：基於「起始解建構模組所產生之起始解相當好」與「子問題求解模組能一次可產生所有 ReducedCost 值小於 0 之可行勤務組合」兩個原因，使得此變數產生法之收斂速度相當快，大多不超過 3 回合就已收斂，而且求解結果也相當不錯。
2. 子問題對目標值之改善比率：一般文獻上所提之變數產生法，皆是先產生一個粗劣之起始解，然後再依賴變數產生法之理論來改善解之品質，所以其子問題對目標值之改善比率相當高；然而，本研究所提之變數產生法乃是先找到一個很好的起始解，再透過變數產生法來改善解的品質，所以本研究之子問題對目標值改善比率會較低。
3. 子問題之運算時間與新增之變數個數：因為以網路模式為基礎之變數產生法乃是運用非常有效率的短路徑演算法或資源受限制的最短路徑演算法來求得一個對目標值有貢獻之新變數，而本研究之子問題乃是運用智慧型之空間搜尋演算法來求得一組對目標值有貢獻之新變數，所以，以網路模式為基礎之變數產生法之子問題運算時間較以限制規劃為基礎之變數產生法來的短，而且每次子問題所能新增之變數個數亦較少。

結論五：限制列舉式之產生-優化法的求解績效相當好

1. 在本研究之測試結果中，不管是哪個測試例題，限制列舉式之產生-優化法的求解績效(解品質、運算時間)皆較以限制規劃為基礎之變數產生法來的佳。
2. 不同於傳統之產生-優化法之求解架構，限制列舉式之產生-優化法乃以 *CP* 集合產生架構為基礎來產生優良勤務組合，使得此種求解架構之求解績效有機會能比變數產生法之求解績效還得好。

結論六：限制規劃相當適合用來求解限制程度高之限制滿足問題

1. 限制規劃方法非常適用於求解限制程度高之限制滿足問題 (Highly-Constrained Satisfaction Problem)(如：勤務組合產生問題、排課問題等)，而且只要能將問題充分根據限制規劃之運作特性，細心地將問題構建成一個其相對應之限制規劃模式，一定會有相當好的求解效率。

7.2 建議

經本研究之實證分析後，提供下列幾點建議，以供為來後續研究之參考：

1. 礙於 ILOG OPL Studio 之限制，本研究無法運用最短路徑演算法，自行設計一些加快搜尋速度之 Constraint Propagation 機制，未來之相關研究可考慮使用 ILOG Solver 與 ILOG CPLEX 函式庫來撰寫程式並加入自己設計之空間搜尋演算法，以期發現更好、更有效率的求解架構。
2. 本研究沒有針對國內航線進行實證分析，未來可嘗試適當地修正本研究之求解架構於國內航線之組員排班問題上，然而由於國內航線之航段距離較短與國際航線不同，所以在求解國內線之排班問題時，應當多考慮至 2 個空載，或甚至是 3 個空載之情況，而且由於國內航線是以天為班次循環週期，所以在求解國內航線時，必須適度地修改一般化優良勤務組合產生架構，以期達到同樣的求解效果。
3. 本研究只針對如何有效地結合限制規劃與數學規劃於求解後艙空勤組員排班問題，對於如何有效地結合限制規劃與數學規劃於求解空勤組員派遣、公車排班、鐵路排班等問題並沒有任何討論，未來之相關研究亦可針對此一主題進行探討。
4. 國內在限制規劃領域之研究尚不興盛，希望藉由本研究之努力與成果，吸引更多學者加入此一研究領域，以加速限制規劃在排班、排課、車輛派遣、生產排程等相關領域之應用。
5. 若考慮本研究實用性，未來可再進一步將此研究成果發展成一個決策支援系統 (DSS)，以期提供排班人員一個良好之決策輔助工具與減輕其負擔。

參考文獻

1. Arabeyre, H. L., F. Fearrnley, C. Steiger, and W. Teather, "The Airline Crew Scheduling Problem : Survey," *Transp. Sci.* 3, 140-163, 1969.
2. Anbil, R., E. Gelman, B. Patty, and R. Tanga, "Recent Advances in Crew-Pairing Optimization at American Airlines," *Interfaces*, vol. 21 , pp. 62-74, 1991.
3. Andersson E., E. Housos, N. Kohl, and D. Wedlin, "Crew Pairing Optimization," 1997.
4. Appelgren, L. H., "A Column Generation Algorithm for a Ship Scheduling Problem," *Transportation Science*, vol 3, pp. 53-68 ,1969.
5. Amal de silva, "Bus Driver Duty Optimization by Combining Constraint Programming and Linear Programming", ILOG(S) Pte Ltd. Singapore, 2000.
6. Bodin, L., B. Golden, A. Assad, and M. Ball, "Routing and Scheduling of Vehcles and Crews --- The State of the Art," *Computers & Operations Research*, Vol. 10, No. 2, Ch.3.3.3, pp. 125-127, 1983.
7. Beasley J. E and B. Cao, "A Tree Search Algorithm for the Crew Scheduling Problem," *European Journal of Operational Research*, vol. 94, no.3, pp. 517-526, 1996.
8. Barnhart, C., and R. Shenoi, "An Approximate Model and Solution Approach for the Long-Haul Crew Pairing Problem," *Transportation Science*, vol. 32, no. 3, pp. 221-231, August 1998.
9. Brailsford, S. C., C. N. Potts, B. M. Smith, "Constraint Satisfaction Problems : Algorithms and Applications," *European Journal of Operational Research* 119, 557-581, 1999.
10. Barták Roman, *On-Line Guide to Constraint Programming*,
<http://ktiml.mff.cuni.cz/~bartak/constraints/> 1998.
11. Crainic, T. G. and J. Rousseau, "The Column Generation Principle and the Airline Crew Scheduling Problem," *INFOR*, vol. 25, no. 2, pp. 136-151 ,1987.
12. Dantzig, G. B. and P. Wolfe, "The Decomposition Algorithm for Linear Programming", *Operations Research*, vol 8, pp. 101-111, 1960.
13. Desrochers, M. and F. Soumis, "A Column Generation Approach to the Urban Transit Crew Scheduling Problem," *Transportation Science*, vol. 23, no. 1, pp. 1-13, 1989.
14. Desrochers, M., J. Gilbert, M. Sauve, F. Soumis, "Crew-Opt : Subproblem Modeling in a Column Generation Approach to Urban Crew Scheduling," IN: DADUNA e WREN (Eds.) *Computer-Aided Transit Scheduling*, pp. 395-406. Lecture Notes in Economics and Mathematical System 386. Springer-Verlag, Berlin, 1992.
15. Desrosiers J., Y. Dumas, M. Desrochers, F. SOUMIS, B. Sanso and P. Trudeau, "A Breakthrough in Airline Crew Scheduling," *Proceedings of the 26th Annual Meeting of the Canadian Transportation Research*, Québec, pp. 464-478, 1991.
16. Desrosiers J., M. Gamache, F. Soumis and G. Marquis, "A Column Generation Approach for Large-Scale Aircrew Rostering Problems," *Operations Research*, vol. 47, no. 2, pp.

- 247-263, 1999.
17. Graves G.W., R.D. McBride and I. Gershkoff, "Flight Crew Scheduling," *Management Science*, vol. 39, no. 6, pp. 736-745, 1993.
 18. Gilmore, P. C. and R. E. Gomory, "A Linear Programming Approach to the Cutting-Stock Problem", *Operations Research*, vol. 9, pp. 849-859 ,1961.
 19. Hillier, F. S. and G. J. Lieberman, *Introduction to Operations Research*, McGraw-Hill, 1995.
 20. Hoffman Karla L. and Padberg M. , "Solving Airline Crew Scheduling Problem by Branch-and-Cut," *Management Science*, vol. 39, no. 6, pp. 657-682, 1993.
 21. Halatsis C., P. Stamatopoulos, I. Karali, T. Bitsikas, G. Fessakis, A. Schizas, S. Sfakianakis, C. Fouskakis, T. Koukoumpetsos and D. Papageorgiou, "Crew Scheduling Based on Constraint Programming: The PARACHUTE Experience". *Proceedings of the 3rd Hellenic-European Conference on Mathematics and Informatics HERMIS '96*, pp. 424-431, Athens, 1996.
 22. Haralick, R. and G. Elliott, "Increasing Tree Search Efficiency for Constraint Satisfaction Problems," *Artificial Intelligence*, vol 14, pp. 263-313, 1980.
 23. Lavoie, S., M. Minoux, and E. Odier, "A New Approach for Crew Pairing Problems by Column Generation with an Application to Air Transportation," *European Journal of Operational Research*, vol. 35, pp. 45-58, 1988.
 24. Levine, D. and Argonne, "Application of a Hybrid Genetic Algorithm to Airline Crew Scheduling", *Computers and Operations Research*, vol. 23, pp. 547-558, 1996.
 25. Lustig, Irvin J., J. F. Puget, "Program Does Not Equal Program: Constraint Programming and Its Relationship to Mathematical Programming," *Interfaces*, vol. 31, Issue 6, pp. 29-53, 2001.
 26. Minoux, M. "Column Generation Techniques in Combinatorial Optimization : A New Application to Crew Pairing Problems," *Proceedings XXIVth AGIFORMS Symposium*, Strabosurg, France, 1984.
 27. Marriott K. and P. J. Stuckey , *Programming with Constraints*, 1998.
 28. Marsten, R. E. and F. Shepardson, "Exact Solution of Crew Scheduling Problems Using the Set Partitioning Model: Successful Applications," *Network*, vol 11, pp. 165-177, 1981.
 29. Nemhauser G.L. and L.A. Wolsey, "*Integer and Combinatorial Optimization*," John Wiley & Sons, New York, 1993.
 30. Niklas. K., "Application of OR and CP Techniques in a Real World Crew Scheduling System", *proceedings of CP-AI-OR'00*, 2000.
 31. Pulleyblank W. R, "The Airline Crew Pairing Optimization Problem," *ISTCS*, 1996.
 32. Pavlopoulou C., A. Gionis, P. Stamatopoulos and C. Halatsis, "Crew Pairing Optimization Based on CLP". *Proceedings of the 2nd International Conference on the Practical Applications of Constraint Technology PACT '96*, pp. 191-210, London, 1996.
 33. Tallys H. Yunes, Arnaldo V. Moura and Cid C. de Souza, "A Hybrid Approach for Solving

- Large Scale Crew Scheduling Problems”, August 1999.
34. Vance, P., E.L. Johnson, and G.L. Nemhauser, “Airline Crew Scheduling : A New Formulation and Decomposition Algorithm,” *Operations Research*, vol. 45, no. 2, pp. 188-200, March-April 1997.
 35. Vance, P.H., E.L., Johnson, G.L., Nemhauser, M.W.F., Savelsbergh, and Barnhart, C., “Branch-and-Price: Column Generation for Solving Huge Integer Programs,” *Operations Research*, vol. 46, No. 3, pp. 316-329, May-June 1998.
 36. Van Hentenryck P. , *The OPL Optimization Programming Language*, 1999.
 37. Wren A. and D. O. Wren , ”A Genetic Algorithm for Public Transport Driver Scheduling,“ *Computers & Operations Research*, vol 22, pp.101-110, 1995.
 38. Young R. D., ”A Simplified Primal (All-Integer) Integer Programming Algorithm,” *Operations Research* vol 16, pp. 750-782, 1968.
 39. Yan S., T.T. Tung, and Y.P. Tu, “Optimal Construction of Airline Individual Crew Pairings,” *Computers & Operations Research*, vol. 29, pp. 341-363, 2002.
 40. 沈志展 , (指導教授：韓復華) , “民航空運排程分析模式之應用” , 國立交通大學交通運輸研究所碩士論文 , 1991。
 41. 林錦翌 , (指導教授：顏上堯) , “空服員排班組合最佳化之研究” , 國立中央大學土木工程研究所 , 1995。
 42. 湯敦台 , (指導教授：顏上堯) , “多基地空服員排班組合最佳化” , *中華民國運輸學會第十一屆論文研討會論文集* , pp. 73-84 , 1996。
 43. 杜宇平 , (指導教授：顏上堯) , “空服員排班網路模式之研究” , 國立中央大學土木工程學系博士論文 , 2000。
 44. 盧宗成 , (指導教授：王晉元) , “捷運公司機員排班問題之研究-以台北捷運公司為例” , 國立交通大學運輸工程與管理學系碩士論文 , 1999。
 45. 蔡文昉 , (指導教授：王晉元) , “大眾運輸排班系統之研究” , 國立交通大學運輸工程與管理學系碩士論文 , 2001。
 46. 中華航空公司網站 , <http://www.china-airlines.com/>。

附錄一

測試例題 1 之航段資料-[沈志展學長碩士論文]

| 航段編號 | 起站 | 星期* | 起飛時間* | 迄站 | 星期* | 抵達時間* | 飛航時間 |
|------|-----|-----|-------|-----|-----|-------|-------|
| 1 | 洛杉磯 | 1 | 14:55 | 台北 | 2 | 19:50 | 12:55 |
| 2 | 洛杉磯 | 2 | 14:55 | 台北 | 3 | 19:50 | 12:55 |
| 3 | 洛杉磯 | 3 | 14:55 | 台北 | 4 | 19:50 | 12:55 |
| 4 | 洛杉磯 | 4 | 14:55 | 台北 | 5 | 19:50 | 12:55 |
| 5 | 洛杉磯 | 5 | 14:55 | 台北 | 6 | 19:50 | 12:55 |
| 6 | 洛杉磯 | 6 | 14:55 | 台北 | 7 | 19:50 | 12:55 |
| 7 | 洛杉磯 | 7 | 14:55 | 台北 | 1 | 19:50 | 12:55 |
| 8 | 香港 | 1 | 11:15 | 台北 | 1 | 12:40 | 01:25 |
| 9 | 香港 | 2 | 11:15 | 台北 | 2 | 12:40 | 01:25 |
| 10 | 香港 | 3 | 11:15 | 台北 | 3 | 12:40 | 01:25 |
| 11 | 香港 | 4 | 11:15 | 台北 | 4 | 12:40 | 01:25 |
| 12 | 香港 | 5 | 11:15 | 台北 | 5 | 12:40 | 01:25 |
| 13 | 香港 | 6 | 11:15 | 台北 | 6 | 12:40 | 01:25 |
| 14 | 香港 | 7 | 11:15 | 台北 | 7 | 12:40 | 01:25 |
| 15 | 台北 | 1 | 08:30 | 香港 | 1 | 10:05 | 01:35 |
| 16 | 台北 | 1 | 12:15 | 東京 | 1 | 16:10 | 02:55 |
| 17 | 台北 | 1 | 12:40 | 香港 | 1 | 14:15 | 01:35 |
| 18 | 台北 | 1 | 14:30 | 東京 | 1 | 18:25 | 02:55 |
| 19 | 台北 | 1 | 16:10 | 洛杉磯 | 1 | 12:50 | 12:40 |
| 20 | 台北 | 1 | 17:20 | 東京 | 1 | 21:15 | 02:55 |
| 21 | 台北 | 2 | 08:30 | 香港 | 2 | 10:05 | 01:35 |
| 22 | 台北 | 2 | 12:15 | 東京 | 2 | 16:10 | 02:55 |
| 23 | 台北 | 2 | 14:30 | 東京 | 2 | 18:25 | 02:55 |
| 24 | 台北 | 2 | 16:10 | 洛杉磯 | 2 | 12:50 | 12:40 |
| 25 | 台北 | 2 | 17:20 | 東京 | 2 | 21:15 | 02:55 |
| 26 | 台北 | 3 | 08:00 | 香港 | 3 | 09:35 | 01:35 |
| 27 | 台北 | 3 | 08:30 | 香港 | 3 | 10:05 | 01:35 |
| 28 | 台北 | 3 | 12:15 | 東京 | 3 | 16:10 | 02:55 |
| 29 | 台北 | 3 | 14:30 | 東京 | 3 | 18:25 | 02:55 |
| 30 | 台北 | 3 | 16:10 | 洛杉磯 | 3 | 12:50 | 12:40 |

| | | | | | | | |
|----|----|---|-------|-----|---|-------|-------|
| 31 | 台北 | 3 | 17:20 | 東京 | 3 | 21:15 | 02:55 |
| 32 | 台北 | 4 | 08:30 | 香港 | 4 | 10:05 | 01:35 |
| 33 | 台北 | 4 | 12:15 | 東京 | 4 | 16:10 | 02:55 |
| 34 | 台北 | 4 | 12:40 | 香港 | 4 | 14:15 | 01:35 |
| 35 | 台北 | 4 | 14:30 | 東京 | 4 | 18:25 | 02:55 |
| 36 | 台北 | 4 | 16:10 | 洛杉磯 | 4 | 12:50 | 12:40 |
| 37 | 台北 | 4 | 17:20 | 東京 | 4 | 21:15 | 02:55 |
| 38 | 台北 | 5 | 08:30 | 香港 | 5 | 10:05 | 01:35 |
| 39 | 台北 | 5 | 12:15 | 東京 | 5 | 16:10 | 02:55 |
| 40 | 台北 | 5 | 14:30 | 東京 | 5 | 18:25 | 02:55 |
| 41 | 台北 | 5 | 16:10 | 洛杉磯 | 5 | 12:50 | 12:40 |
| 42 | 台北 | 5 | 17:20 | 東京 | 5 | 21:15 | 02:55 |
| 43 | 台北 | 6 | 08:30 | 香港 | 6 | 10:05 | 01:35 |
| 44 | 台北 | 6 | 12:15 | 東京 | 6 | 16:10 | 02:55 |
| 45 | 台北 | 6 | 12:40 | 香港 | 6 | 14:15 | 01:35 |
| 46 | 台北 | 6 | 14:30 | 東京 | 6 | 18:25 | 02:55 |
| 47 | 台北 | 6 | 16:10 | 洛杉磯 | 6 | 12:50 | 12:40 |
| 48 | 台北 | 6 | 17:20 | 東京 | 6 | 21:15 | 02:55 |
| 49 | 台北 | 7 | 08:30 | 香港 | 7 | 10:05 | 01:35 |
| 50 | 台北 | 7 | 12:15 | 東京 | 7 | 16:10 | 02:55 |
| 51 | 台北 | 7 | 14:30 | 東京 | 7 | 18:25 | 02:55 |
| 52 | 台北 | 7 | 16:10 | 洛杉磯 | 7 | 12:50 | 12:40 |
| 53 | 台北 | 7 | 17:20 | 東京 | 7 | 21:15 | 02:55 |
| 54 | 東京 | 1 | 08:50 | 台北 | 1 | 10:55 | 03:05 |
| 55 | 東京 | 1 | 17:20 | 台北 | 1 | 19:25 | 03:05 |
| 56 | 東京 | 2 | 08:50 | 台北 | 2 | 10:55 | 03:05 |
| 57 | 東京 | 2 | 17:20 | 台北 | 2 | 19:25 | 03:05 |
| 58 | 東京 | 3 | 08:50 | 台北 | 3 | 10:55 | 03:05 |
| 59 | 東京 | 3 | 17:20 | 台北 | 3 | 19:25 | 03:05 |
| 60 | 東京 | 4 | 08:50 | 台北 | 4 | 10:55 | 03:05 |
| 61 | 東京 | 4 | 17:20 | 台北 | 4 | 19:25 | 03:05 |
| 62 | 東京 | 5 | 08:50 | 台北 | 5 | 10:55 | 03:05 |
| 63 | 東京 | 5 | 17:20 | 台北 | 5 | 19:25 | 03:05 |
| 64 | 東京 | 6 | 08:50 | 台北 | 6 | 10:55 | 03:05 |
| 65 | 東京 | 6 | 17:20 | 台北 | 6 | 19:25 | 03:05 |
| 66 | 東京 | 7 | 08:50 | 台北 | 7 | 10:55 | 03:05 |

| | | | | | | | |
|----|----|---|-------|----|---|-------|-------|
| 67 | 東京 | 7 | 17:20 | 台北 | 7 | 19:25 | 03:05 |
| 68 | 東京 | 7 | 19:35 | 台北 | 7 | 21:40 | 03:05 |

*：乃指當地時間

附錄二

測試例題 2 之航段資料-[湯敦台學長碩士論文]

| 航段 編號 | 航段 代碼 | 起站 | 星期* | 起飛 時間* | 迄站 | 星期* | 抵達 時間* | 飛航時間 (分) | 機型 |
|----------|----------|-----|-----|-----------|-----|-----|-----------|-------------|-----|
| 1 | 018_4 | TPE | 4 | 14:25 | TYO | 4 | 17:25 | 180 | 74L |
| 2 | 018_7 | TPE | 7 | 14:25 | TYO | 7 | 17:25 | 180 | 742 |
| 3 | 100_1 | TPE | 1 | 12:00 | TYO | 1 | 14:55 | 175 | 744 |
| 4 | 100_2 | TPE | 2 | 12:00 | TYO | 2 | 14:55 | 175 | 744 |
| 5 | 100_3 | TPE | 3 | 12:00 | TYO | 3 | 14:55 | 175 | 744 |
| 6 | 100_4 | TPE | 4 | 12:00 | TYO | 4 | 14:55 | 175 | 742 |
| 7 | 100_5 | TPE | 5 | 12:00 | TYO | 5 | 14:55 | 175 | 744 |
| 8 | 100_6 | TPE | 6 | 12:00 | TYO | 6 | 14:55 | 175 | 744 |
| 9 | 100_7 | TPE | 7 | 12:00 | TYO | 7 | 14:55 | 175 | 744 |
| 10 | 106_1 | TPE | 1 | 17:20 | TYO | 1 | 20:20 | 180 | 742 |
| 11 | 106_2 | TPE | 2 | 17:20 | TYO | 2 | 20:20 | 180 | 742 |
| 12 | 106_3 | TPE | 3 | 17:20 | TYO | 3 | 20:20 | 180 | 742 |
| 13 | 106_4 | TPE | 4 | 17:20 | TYO | 4 | 20:20 | 180 | 742 |
| 14 | 106_5 | TPE | 5 | 17:20 | TYO | 5 | 20:20 | 180 | 742 |
| 15 | 106_6 | TPE | 6 | 17:20 | TYO | 6 | 20:20 | 180 | 742 |
| 16 | 106_7 | TPE | 7 | 17:20 | TYO | 7 | 20:20 | 180 | 742 |
| 17 | 110_1 | TPE | 1 | 16:20 | FUK | 1 | 18:30 | 130 | AB6 |
| 18 | 110_2 | TPE | 2 | 16:20 | FUK | 2 | 18:30 | 130 | AB6 |
| 19 | 110_3 | TPE | 3 | 16:20 | FUK | 3 | 18:30 | 130 | AB6 |
| 20 | 110_4 | TPE | 4 | 16:20 | FUK | 4 | 18:30 | 130 | AB6 |
| 21 | 110_5 | TPE | 5 | 16:20 | FUK | 5 | 18:30 | 130 | AB6 |
| 22 | 110_6 | TPE | 6 | 16:20 | FUK | 6 | 18:30 | 130 | AB6 |
| 23 | 110_7 | TPE | 7 | 16:20 | FUK | 7 | 18:30 | 130 | AB6 |
| 24 | 120_2 | TPE | 2 | 08:10 | OKA | 2 | 09:30 | 80 | 734 |
| 25 | 120_3 | TPE | 3 | 08:10 | OKA | 3 | 09:30 | 80 | 734 |
| 26 | 120_4 | TPE | 4 | 08:10 | OKA | 4 | 09:30 | 80 | 734 |
| 27 | 120_6 | TPE | 6 | 08:10 | OKA | 6 | 09:30 | 80 | 734 |
| 28 | 120_7 | TPE | 7 | 08:10 | OKA | 7 | 09:30 | 80 | 734 |
| 29 | 122_2 | TPE | 2 | 16:10 | OKA | 2 | 17:30 | 80 | 734 |
| 30 | 122_3 | TPE | 3 | 16:10 | OKA | 3 | 17:30 | 80 | 734 |
| 31 | 122_4 | TPE | 4 | 16:10 | OKA | 4 | 17:30 | 80 | 734 |
| 32 | 122_6 | TPE | 6 | 16:10 | OKA | 6 | 17:30 | 80 | 734 |

| | | | | | | | | | |
|----|-------|-----|---|-------|-----|---|-------|-----|-----|
| 33 | 122_7 | TPE | 7 | 16:10 | OKA | 7 | 17:30 | 80 | 734 |
| 34 | 126 | TPE | 1 | 08:10 | OKA | 1 | 09:30 | 80 | 734 |
| 35 | 128 | TPE | 5 | 16:10 | OKA | 5 | 17:30 | 80 | 734 |
| 36 | 150_1 | TPE | 1 | 16:40 | NGO | 1 | 19:25 | 165 | AB6 |
| 37 | 150_2 | TPE | 2 | 16:40 | NGO | 2 | 19:25 | 165 | AB6 |
| 38 | 150_4 | TPE | 4 | 16:40 | NGO | 4 | 19:25 | 165 | AB6 |
| 39 | 150_5 | TPE | 5 | 16:40 | NGO | 5 | 19:25 | 165 | AB6 |
| 40 | 150_6 | TPE | 6 | 16:40 | NGO | 6 | 19:25 | 165 | AB6 |
| 41 | 017_5 | TYO | 5 | 13:00 | TPE | 5 | 16:20 | 200 | 74L |
| 42 | 101_1 | TYO | 1 | 16:20 | TPE | 1 | 19:40 | 200 | 744 |
| 43 | 101_2 | TYO | 2 | 16:20 | TPE | 2 | 19:40 | 200 | 744 |
| 44 | 101_3 | TYO | 3 | 16:20 | TPE | 3 | 19:40 | 200 | 744 |
| 45 | 101_4 | TYO | 4 | 16:20 | TPE | 4 | 19:40 | 200 | 742 |
| 46 | 101_5 | TYO | 5 | 16:20 | TPE | 5 | 19:40 | 200 | 744 |
| 47 | 101_6 | TYO | 6 | 16:20 | TPE | 6 | 19:40 | 200 | 744 |
| 48 | 101_7 | TYO | 7 | 16:20 | TPE | 7 | 19:40 | 200 | 744 |
| 49 | 105 | TYO | 7 | 18:35 | TPE | 7 | 21:55 | 200 | 742 |
| 50 | 107_1 | TYO | 1 | 07:50 | TPE | 1 | 11:10 | 200 | 742 |
| 51 | 107_2 | TYO | 2 | 07:50 | TPE | 2 | 11:10 | 200 | 742 |
| 52 | 107_3 | TYO | 3 | 07:50 | TPE | 3 | 11:10 | 200 | 742 |
| 53 | 107_4 | TYO | 4 | 07:50 | TPE | 4 | 11:10 | 200 | 742 |
| 54 | 107_5 | TYO | 5 | 07:50 | TPE | 5 | 11:10 | 200 | 742 |
| 55 | 107_6 | TYO | 6 | 07:50 | TPE | 6 | 11:10 | 200 | 742 |
| 56 | 107_7 | TYO | 7 | 07:50 | TPE | 7 | 11:10 | 200 | 742 |
| 57 | 111_1 | FUK | 1 | 09:10 | TPE | 1 | 11:25 | 135 | AB6 |
| 58 | 111_2 | FUK | 2 | 09:10 | TPE | 2 | 11:25 | 135 | AB6 |
| 59 | 111_3 | FUK | 3 | 09:10 | TPE | 3 | 11:25 | 135 | AB6 |
| 60 | 111_4 | FUK | 4 | 09:10 | TPE | 4 | 11:25 | 135 | AB6 |
| 61 | 111_5 | FUK | 5 | 09:10 | TPE | 5 | 11:25 | 135 | AB6 |
| 62 | 111_6 | FUK | 6 | 09:10 | TPE | 6 | 11:25 | 135 | AB6 |
| 63 | 111_7 | FUK | 7 | 09:10 | TPE | 7 | 11:25 | 135 | AB6 |
| 64 | 121_2 | OKA | 2 | 10:20 | TPE | 2 | 11:45 | 85 | 734 |
| 65 | 121_3 | OKA | 3 | 10:20 | TPE | 3 | 11:45 | 85 | 734 |
| 66 | 121_4 | OKA | 4 | 10:20 | TPE | 4 | 11:45 | 85 | 734 |
| 67 | 121_6 | OKA | 6 | 10:20 | TPE | 6 | 11:45 | 85 | 734 |
| 68 | 121_7 | OKA | 7 | 10:20 | TPE | 7 | 11:45 | 85 | 734 |
| 69 | 123_2 | OKA | 2 | 18:20 | TPE | 2 | 19:45 | 85 | 734 |

| | | | | | | | | | |
|-----|---------|-----|---|-------|-----|---|-------|-----|-----|
| 70 | 123_3 | OKA | 3 | 18:20 | TPE | 3 | 19:45 | 85 | 734 |
| 71 | 123_4 | OKA | 4 | 18:20 | TPE | 4 | 19:45 | 85 | 734 |
| 72 | 123_6 | OKA | 6 | 18:20 | TPE | 6 | 19:45 | 85 | 734 |
| 73 | 123_7 | OKA | 7 | 18:20 | TPE | 7 | 19:45 | 85 | 734 |
| 74 | 127 | OKA | 1 | 10:20 | TPE | 1 | 11:45 | 85 | 734 |
| 75 | 129 | OKA | 5 | 18:20 | TPE | 5 | 19:45 | 85 | 734 |
| 76 | 151_2 | NGO | 2 | 08:10 | TPE | 2 | 11:10 | 180 | AB6 |
| 77 | 151_3 | NGO | 3 | 08:10 | TPE | 3 | 11:10 | 180 | AB6 |
| 78 | 151_5 | NGO | 5 | 08:10 | TPE | 5 | 11:10 | 180 | AB6 |
| 79 | 151_6 | NGO | 6 | 07:25 | TPE | 6 | 10:25 | 180 | AB6 |
| 80 | 151_7 | NGO | 7 | 08:10 | TPE | 7 | 11:10 | 180 | AB6 |
| 81 | 002_2 | TPE | 2 | 21:00 | HNL | 3 | 06:20 | 560 | 74L |
| 82 | 002_4 | TPE | 4 | 21:00 | HNL | 5 | 06:20 | 560 | 74L |
| 83 | 002_6 | TPE | 6 | 21:00 | HNL | 7 | 06:20 | 560 | 74L |
| 84 | 004_1 | TPE | 1 | 16:40 | SFO | 2 | 04:55 | 735 | M11 |
| 85 | 004_2 | TPE | 2 | 16:40 | SFO | 3 | 04:55 | 735 | M11 |
| 86 | 004_3 | TPE | 3 | 16:40 | SFO | 4 | 04:55 | 735 | M11 |
| 87 | 004_4 | TPE | 4 | 16:40 | SFO | 5 | 04:55 | 735 | M11 |
| 88 | 004_5 | TPE | 5 | 16:40 | SFO | 6 | 04:55 | 735 | M11 |
| 89 | 004_6 | TPE | 6 | 16:40 | SFO | 7 | 04:55 | 735 | M11 |
| 90 | 004_7 | TPE | 7 | 16:40 | SFO | 1 | 04:55 | 735 | M11 |
| 91 | 006_1 | TPE | 1 | 16:10 | LAX | 2 | 04:35 | 745 | 744 |
| 92 | 006_2 | TPE | 2 | 16:10 | LAX | 3 | 04:35 | 745 | 744 |
| 93 | 006_3 | TPE | 3 | 16:10 | LAX | 4 | 04:35 | 745 | 744 |
| 94 | 006_4 | TPE | 4 | 16:10 | LAX | 5 | 04:35 | 745 | 744 |
| 95 | 006_5 | TPE | 5 | 16:10 | LAX | 6 | 04:35 | 745 | 744 |
| 96 | 006_6 | TPE | 6 | 16:10 | LAX | 7 | 04:35 | 745 | 744 |
| 97 | 006_7 | TPE | 7 | 16:10 | LAX | 1 | 04:35 | 745 | 744 |
| 98 | 008_2 | TPE | 2 | 22:30 | LAX | 3 | 10:55 | 745 | 744 |
| 99 | 008_5 | TPE | 5 | 22:30 | LAX | 6 | 10:55 | 745 | 744 |
| 100 | 008_7 | TPE | 7 | 22:30 | LAX | 1 | 10:55 | 745 | 744 |
| 101 | 012_1_A | TPE | 1 | 16:30 | ANC | 2 | 01:10 | 520 | 744 |
| 102 | 012_1_B | ANC | 2 | 02:25 | NYC | 2 | 09:05 | 400 | 744 |
| 103 | 012_3_A | TPE | 3 | 16:30 | ANC | 4 | 01:10 | 520 | 744 |
| 104 | 012_3_B | ANC | 4 | 02:25 | NYC | 4 | 09:05 | 400 | 744 |
| 105 | 012_4_A | TPE | 4 | 16:30 | ANC | 5 | 01:10 | 520 | 744 |
| 106 | 012_4_B | ANC | 5 | 02:10 | NYC | 5 | 08:50 | 400 | 744 |

| | | | | | | | | | |
|-----|---------|-----|---|-------|-----|---|-------|-----|-----|
| 107 | 012_6_A | TPE | 6 | 16:30 | ANC | 7 | 01:10 | 520 | 744 |
| 108 | 012_6_B | ANC | 7 | 02:25 | NYC | 7 | 09:05 | 400 | 744 |
| 109 | 018_1_A | TPE | 1 | 14:25 | TYO | 1 | 17:25 | 180 | 742 |
| 110 | 018_1_B | TYO | 2 | 18:35 | HNL | 2 | 01:50 | 435 | 742 |
| 111 | 018_2_A | TPE | 2 | 14:25 | TYO | 2 | 17:25 | 180 | 742 |
| 112 | 018_2_B | TYO | 2 | 18:35 | HNL | 3 | 01:50 | 435 | 742 |
| 113 | 018_3_A | TPE | 3 | 14:25 | TYO | 3 | 17:25 | 180 | 742 |
| 114 | 018_3_B | TYO | 3 | 18:35 | HNL | 4 | 01:50 | 435 | 742 |
| 115 | 018_5_A | TPE | 5 | 14:25 | TYO | 5 | 17:25 | 180 | 742 |
| 116 | 018_5_B | TYO | 5 | 18:35 | HNL | 6 | 01:50 | 435 | 742 |
| 117 | 018_6_A | TPE | 6 | 14:25 | TYO | 6 | 17:25 | 180 | 742 |
| 118 | 018_6_B | TYO | 6 | 18:35 | HNL | 7 | 01:50 | 435 | 742 |
| 119 | 001_3 | HNL | 3 | 08:10 | TPE | 3 | 18:20 | 610 | 74L |
| 120 | 001_5 | HNL | 5 | 08:10 | TPE | 5 | 18:20 | 610 | 74L |
| 121 | 001_7 | HNL | 7 | 08:10 | TPE | 7 | 18:20 | 610 | 74L |
| 122 | 003_2 | SFO | 2 | 06:55 | TPE | 2 | 19:50 | 775 | M11 |
| 123 | 003_3 | SFO | 3 | 06:55 | TPE | 3 | 19:50 | 775 | M11 |
| 124 | 003_4 | SFO | 4 | 06:55 | TPE | 4 | 19:50 | 775 | M11 |
| 125 | 003_5 | SFO | 5 | 06:55 | TPE | 5 | 19:50 | 775 | M11 |
| 126 | 003_6 | SFO | 6 | 06:55 | TPE | 6 | 19:50 | 775 | M11 |
| 127 | 003_7 | SFO | 7 | 06:55 | TPE | 7 | 19:50 | 775 | M11 |
| 128 | 003_1 | SFO | 1 | 06:55 | TPE | 1 | 19:50 | 775 | M11 |
| 129 | 005_2 | LAX | 2 | 06:30 | TPE | 2 | 19:45 | 795 | 744 |
| 130 | 005_3 | LAX | 3 | 06:30 | TPE | 3 | 19:45 | 795 | 744 |
| 131 | 005_4 | LAX | 4 | 06:30 | TPE | 4 | 19:45 | 795 | 744 |
| 132 | 005_5 | LAX | 5 | 06:30 | TPE | 5 | 19:45 | 795 | 744 |
| 133 | 005_6 | LAX | 6 | 06:30 | TPE | 6 | 19:45 | 795 | 744 |
| 134 | 005_7 | LAX | 7 | 06:30 | TPE | 7 | 19:45 | 795 | 744 |
| 135 | 005_1 | LAX | 1 | 06:30 | TPE | 1 | 19:45 | 795 | 744 |
| 136 | 007_1 | LAX | 1 | 16:30 | TPE | 2 | 05:45 | 795 | 744 |
| 137 | 007_3 | LAX | 3 | 16:30 | TPE | 4 | 05:45 | 795 | 744 |
| 138 | 007_6 | LAX | 6 | 16:30 | TPE | 7 | 05:45 | 795 | 744 |
| 139 | 011_2_A | NYC | 2 | 11:50 | ANC | 2 | 19:00 | 430 | 744 |
| 140 | 011_2_B | ANC | 2 | 20:25 | TPE | 3 | 06:15 | 590 | 744 |
| 141 | 011_4_A | NYC | 4 | 11:50 | ANC | 4 | 19:00 | 430 | 744 |
| 142 | 011_4_B | ANC | 4 | 20:25 | TPE | 5 | 06:15 | 590 | 744 |
| 143 | 011_5_A | NYC | 5 | 11:50 | ANC | 5 | 19:00 | 430 | 744 |

| | | | | | | | | | |
|-----|---------|-----|---|-------|-----|---|-------|-----|-----|
| 144 | 011_5_B | ANC | 5 | 20:25 | TPE | 6 | 06:15 | 590 | 744 |
| 145 | 011_7_A | NYC | 7 | 11:50 | ANC | 7 | 19:00 | 430 | 744 |
| 146 | 011_7_B | ANC | 7 | 20:25 | TPE | 1 | 06:15 | 590 | 744 |
| 147 | 017_2_A | HNL | 2 | 03:40 | TYO | 2 | 11:50 | 490 | 742 |
| 148 | 017_2_B | TYO | 2 | 13:00 | TPE | 2 | 16:20 | 200 | 742 |
| 149 | 017_3_A | HNL | 3 | 03:40 | TYO | 3 | 11:50 | 490 | 742 |
| 150 | 017_3_B | TYO | 3 | 13:00 | TPE | 3 | 16:20 | 200 | 742 |
| 151 | 017_4_A | HNL | 4 | 03:40 | TYO | 4 | 11:50 | 490 | 742 |
| 152 | 017_4_B | TYO | 4 | 13:00 | TPE | 4 | 16:20 | 200 | 742 |
| 153 | 017_6_A | HNL | 6 | 03:40 | TYO | 6 | 11:50 | 490 | 742 |
| 154 | 017_6_B | TYO | 6 | 13:00 | TPE | 6 | 16:20 | 200 | 742 |
| 155 | 017_7_A | HNL | 7 | 03:40 | TYO | 7 | 11:50 | 490 | 742 |
| 156 | 017_7_B | TYO | 7 | 13:00 | TPE | 7 | 16:20 | 200 | 742 |
| 157 | 065_1_A | TPE | 1 | 20:30 | BKK | 2 | 00:10 | 220 | M11 |
| 158 | 065_1_B | BKK | 2 | 01:20 | AMS | 2 | 14:05 | 765 | M11 |
| 159 | 065_2_A | TPE | 2 | 20:30 | BKK | 3 | 00:10 | 220 | M11 |
| 160 | 065_2_B | BKK | 3 | 01:20 | AMS | 3 | 14:05 | 765 | M11 |
| 161 | 065_4_A | TPE | 4 | 20:30 | BKK | 5 | 00:10 | 220 | M11 |
| 162 | 065_4_B | BKK | 5 | 01:20 | AMS | 5 | 14:05 | 765 | M11 |
| 163 | 065_5_A | TPE | 5 | 20:30 | BKK | 6 | 00:10 | 220 | M11 |
| 164 | 065_5_B | BKK | 6 | 01:20 | AMS | 6 | 14:05 | 765 | M11 |
| 165 | 065_6_A | TPE | 6 | 20:30 | BKK | 7 | 00:10 | 220 | M11 |
| 166 | 065_6_B | BKK | 7 | 01:20 | AMS | 7 | 14:05 | 765 | M11 |
| 167 | 065_7_A | TPE | 7 | 20:30 | BKK | 1 | 00:10 | 220 | M11 |
| 168 | 065_7_B | BKK | 1 | 01:20 | AMS | 1 | 14:05 | 765 | M11 |
| 169 | 066_1_A | AMS | 1 | 20:25 | BKK | 2 | 08:05 | 700 | M11 |
| 170 | 066_1_B | BKK | 2 | 09:25 | TPE | 2 | 13:00 | 215 | M11 |
| 171 | 066_2_A | AMS | 2 | 20:25 | BKK | 3 | 08:05 | 700 | M11 |
| 172 | 066_2_B | BKK | 3 | 09:25 | TPE | 3 | 13:00 | 215 | M11 |
| 173 | 066_3_A | AMS | 3 | 20:25 | BKK | 4 | 08:05 | 700 | M11 |
| 174 | 066_3_B | BKK | 4 | 09:25 | TPE | 4 | 13:00 | 215 | M11 |
| 175 | 066_5_A | AMS | 5 | 20:25 | BKK | 6 | 08:05 | 700 | M11 |
| 176 | 066_5_B | BKK | 6 | 09:25 | TPE | 6 | 13:00 | 215 | M11 |
| 177 | 066_6_A | AMS | 6 | 20:25 | BKK | 7 | 08:05 | 700 | M11 |
| 178 | 066_6_B | BKK | 7 | 09:25 | TPE | 7 | 13:00 | 215 | M11 |
| 179 | 066_7_A | AMS | 7 | 20:25 | BKK | 1 | 08:05 | 700 | M11 |
| 180 | 066_7_B | BKK | 1 | 09:25 | TPE | 1 | 13:00 | 215 | M11 |

| | | | | | | | | | |
|-----|---------|------|---|-------|------|---|-------|-----|-----|
| 181 | 061_3_A | TPE | 3 | 17:50 | KUL | 3 | 22:20 | 270 | M11 |
| 182 | 061_3_B | KUL | 3 | 23:30 | FRA1 | 4 | 12:20 | 770 | M11 |
| 183 | 061_3_C | FRA1 | 4 | 13:20 | FRA | 4 | 19:25 | 365 | M11 |
| 184 | 061_7_A | TPE | 7 | 17:50 | KUL | 7 | 22:20 | 270 | M11 |
| 185 | 061_7_B | KUL | 7 | 23:30 | FRA1 | 1 | 12:20 | 770 | M11 |
| 186 | 061_7_C | FRA1 | 1 | 13:20 | FRA | 1 | 19:25 | 365 | M11 |
| 187 | 062_1_B | FRA | 1 | 20:40 | KUL | 2 | 08:50 | 730 | M11 |
| 188 | 062_1_C | KUL | 2 | 10:15 | TPE | 2 | 14:45 | 270 | M11 |
| 189 | 062_4_B | FRA | 4 | 20:40 | KUL | 5 | 08:50 | 730 | M11 |
| 190 | 062_4_C | KUL | 4 | 10:15 | TPE | 4 | 14:45 | 270 | M11 |
| 191 | 601_2 | TPE | 2 | 07:00 | HKG | 2 | 08:45 | 105 | AB3 |
| 192 | 601_5 | TPE | 5 | 07:00 | HKG | 5 | 08:45 | 105 | AB3 |
| 193 | 601_7 | TPE | 7 | 07:00 | HKG | 7 | 08:45 | 105 | 734 |
| 194 | 603_1 | TPE | 1 | 08:30 | HKG | 1 | 10:15 | 105 | 742 |
| 195 | 603_2 | TPE | 2 | 08:30 | HKG | 2 | 10:15 | 105 | 744 |
| 196 | 603_3 | TPE | 3 | 08:30 | HKG | 3 | 10:15 | 105 | M11 |
| 197 | 603_4 | TPE | 4 | 08:30 | HKG | 4 | 10:15 | 105 | 744 |
| 198 | 603_5 | TPE | 5 | 08:30 | HKG | 5 | 10:15 | 105 | 742 |
| 199 | 603_6 | TPE | 6 | 08:30 | HKG | 6 | 10:15 | 105 | 744 |
| 200 | 603_7 | TPE | 7 | 08:30 | HKG | 7 | 10:15 | 105 | AB6 |
| 201 | 605_1 | TPE | 1 | 10:00 | HKG | 1 | 11:45 | 105 | 744 |
| 202 | 605_2 | TPE | 2 | 10:00 | HKG | 2 | 11:45 | 105 | 744 |
| 203 | 605_3 | TPE | 3 | 10:00 | HKG | 3 | 11:45 | 105 | 744 |
| 204 | 605_4 | TPE | 4 | 10:00 | HKG | 4 | 11:45 | 105 | 744 |
| 205 | 605_5 | TPE | 5 | 10:00 | HKG | 5 | 11:45 | 105 | 744 |
| 206 | 605_6 | TPE | 6 | 10:00 | HKG | 6 | 11:45 | 105 | 744 |
| 207 | 605_7 | TPE | 7 | 10:00 | HKG | 7 | 11:45 | 105 | 744 |
| 208 | 607_1 | TPE | 1 | 10:45 | HKG | 1 | 12:30 | 105 | 744 |
| 209 | 607_2 | TPE | 2 | 10:45 | HKG | 2 | 12:30 | 105 | 744 |
| 210 | 607_3 | TPE | 3 | 10:45 | HKG | 3 | 12:30 | 105 | 744 |
| 211 | 607_4 | TPE | 4 | 10:45 | HKG | 4 | 12:30 | 105 | 744 |
| 212 | 607_5 | TPE | 5 | 10:45 | HKG | 5 | 12:30 | 105 | 744 |
| 213 | 607_6 | TPE | 6 | 10:45 | HKG | 6 | 12:30 | 105 | 744 |
| 214 | 607_7 | TPE | 7 | 10:45 | HKG | 7 | 12:30 | 105 | M11 |
| 215 | 609_1 | TPE | 1 | 12:30 | HKG | 1 | 14:15 | 105 | AB6 |
| 216 | 609_2 | TPE | 2 | 12:30 | HKG | 2 | 14:15 | 105 | AB6 |
| 217 | 609_3 | TPE | 3 | 12:30 | HKG | 3 | 14:15 | 105 | AB6 |

| | | | | | | | | | |
|-----|-------|-----|---|-------|-----|---|-------|-----|-----|
| 218 | 609_4 | TPE | 4 | 12:30 | HKG | 4 | 14:15 | 105 | AB6 |
| 219 | 609_5 | TPE | 5 | 12:30 | HKG | 5 | 14:15 | 105 | AB6 |
| 220 | 609_6 | TPE | 6 | 12:30 | HKG | 6 | 14:15 | 105 | AB6 |
| 221 | 609_7 | TPE | 7 | 12:30 | HKG | 7 | 14:15 | 105 | AB6 |
| 222 | 611_1 | TPE | 1 | 14:15 | HKG | 1 | 16:00 | 105 | M11 |
| 223 | 611_2 | TPE | 2 | 14:15 | HKG | 2 | 16:00 | 105 | AB6 |
| 224 | 611_3 | TPE | 3 | 14:15 | HKG | 3 | 16:00 | 105 | AB6 |
| 225 | 611_4 | TPE | 4 | 14:15 | HKG | 4 | 16:00 | 105 | AB6 |
| 226 | 611_5 | TPE | 5 | 14:15 | HKG | 5 | 16:00 | 105 | AB6 |
| 227 | 611_6 | TPE | 6 | 14:15 | HKG | 6 | 16:00 | 105 | M11 |
| 228 | 611_7 | TPE | 7 | 14:15 | HKG | 7 | 16:00 | 105 | M11 |
| 229 | 615_1 | TPE | 1 | 16:30 | HKG | 1 | 18:15 | 105 | AB6 |
| 230 | 615_2 | TPE | 2 | 16:30 | HKG | 2 | 18:15 | 105 | AB6 |
| 231 | 615_3 | TPE | 3 | 16:30 | HKG | 3 | 18:15 | 105 | AB6 |
| 232 | 615_4 | TPE | 4 | 16:30 | HKG | 4 | 18:15 | 105 | 744 |
| 233 | 615_5 | TPE | 5 | 16:30 | HKG | 5 | 18:15 | 105 | 744 |
| 234 | 615_6 | TPE | 6 | 16:30 | HKG | 6 | 18:15 | 105 | AB6 |
| 235 | 615_7 | TPE | 7 | 16:30 | HKG | 7 | 18:15 | 105 | AB6 |
| 236 | 617_1 | TPE | 1 | 20:30 | HKG | 1 | 22:15 | 105 | AB6 |
| 237 | 617_2 | TPE | 2 | 20:30 | HKG | 2 | 22:15 | 105 | AB6 |
| 238 | 617_3 | TPE | 3 | 20:30 | HKG | 3 | 22:15 | 105 | AB6 |
| 239 | 617_4 | TPE | 4 | 20:30 | HKG | 4 | 22:15 | 105 | AB6 |
| 240 | 617_5 | TPE | 5 | 20:30 | HKG | 5 | 22:15 | 105 | AB6 |
| 241 | 617_6 | TPE | 6 | 20:30 | HKG | 6 | 22:15 | 105 | AB6 |
| 242 | 617_7 | TPE | 7 | 20:30 | HKG | 7 | 22:15 | 105 | AB6 |
| 243 | 602_2 | HKG | 2 | 09:45 | TPE | 2 | 11:15 | 90 | AB3 |
| 244 | 602_5 | HKG | 5 | 09:45 | TPE | 5 | 11:15 | 90 | AB3 |
| 245 | 602_7 | HKG | 7 | 09:45 | TPE | 7 | 11:15 | 90 | 734 |
| 246 | 604_1 | HKG | 1 | 11:15 | TPE | 1 | 12:45 | 90 | 742 |
| 247 | 604_2 | HKG | 2 | 11:15 | TPE | 2 | 12:45 | 90 | 744 |
| 248 | 604_3 | HKG | 3 | 11:15 | TPE | 3 | 12:45 | 90 | M11 |
| 249 | 604_4 | HKG | 4 | 11:15 | TPE | 4 | 12:45 | 90 | 744 |
| 250 | 604_5 | HKG | 5 | 11:15 | TPE | 5 | 12:45 | 90 | 742 |
| 251 | 604_6 | HKG | 6 | 11:15 | TPE | 6 | 12:45 | 90 | 744 |
| 252 | 604_7 | HKG | 7 | 11:15 | TPE | 7 | 12:45 | 90 | AB6 |
| 253 | 606_1 | HKG | 1 | 12:50 | TPE | 1 | 14:20 | 90 | 744 |
| 254 | 606_2 | HKG | 2 | 12:50 | TPE | 2 | 14:20 | 90 | 744 |

| | | | | | | | | | |
|-----|-------|-----|---|-------|-----|---|-------|----|-----|
| 255 | 606_3 | HKG | 3 | 12:50 | TPE | 3 | 14:20 | 90 | 744 |
| 256 | 606_4 | HKG | 4 | 12:50 | TPE | 4 | 14:20 | 90 | 744 |
| 257 | 606_5 | HKG | 5 | 12:50 | TPE | 5 | 14:20 | 90 | 744 |
| 258 | 606_6 | HKG | 6 | 12:50 | TPE | 6 | 14:20 | 90 | 744 |
| 259 | 606_7 | HKG | 7 | 12:50 | TPE | 7 | 14:20 | 90 | 744 |
| 260 | 608_1 | HKG | 1 | 13:40 | TPE | 1 | 15:10 | 90 | 744 |
| 261 | 608_2 | HKG | 2 | 13:40 | TPE | 2 | 15:10 | 90 | 744 |
| 262 | 608_3 | HKG | 3 | 13:40 | TPE | 3 | 15:10 | 90 | 744 |
| 263 | 608_4 | HKG | 4 | 13:40 | TPE | 4 | 15:10 | 90 | 744 |
| 264 | 608_5 | HKG | 5 | 13:40 | TPE | 5 | 15:10 | 90 | 744 |
| 265 | 608_6 | HKG | 6 | 13:40 | TPE | 6 | 15:10 | 90 | 744 |
| 266 | 608_7 | HKG | 7 | 13:40 | TPE | 7 | 15:10 | 90 | M11 |
| 267 | 610_1 | HKG | 1 | 15:35 | TPE | 1 | 17:05 | 90 | AB6 |
| 268 | 610_2 | HKG | 2 | 15:35 | TPE | 2 | 17:05 | 90 | AB6 |
| 269 | 610_3 | HKG | 3 | 15:35 | TPE | 3 | 17:05 | 90 | AB6 |
| 270 | 610_4 | HKG | 4 | 15:35 | TPE | 4 | 17:05 | 90 | AB6 |
| 271 | 610_5 | HKG | 5 | 15:35 | TPE | 5 | 17:05 | 90 | AB6 |
| 272 | 610_6 | HKG | 6 | 15:35 | TPE | 6 | 17:05 | 90 | AB6 |
| 273 | 610_7 | HKG | 7 | 15:35 | TPE | 7 | 17:05 | 90 | AB6 |
| 274 | 612_1 | HKG | 1 | 17:05 | TPE | 1 | 18:35 | 90 | M11 |
| 275 | 612_2 | HKG | 2 | 17:05 | TPE | 2 | 18:35 | 90 | AB6 |
| 276 | 612_3 | HKG | 3 | 17:05 | TPE | 3 | 18:35 | 90 | AB6 |
| 277 | 612_4 | HKG | 4 | 17:05 | TPE | 4 | 18:35 | 90 | AB6 |
| 278 | 612_5 | HKG | 5 | 17:05 | TPE | 5 | 18:35 | 90 | AB6 |
| 279 | 612_6 | HKG | 6 | 17:05 | TPE | 6 | 18:35 | 90 | M11 |
| 280 | 612_7 | HKG | 7 | 17:05 | TPE | 7 | 18:35 | 90 | M11 |
| 281 | 616_1 | HKG | 1 | 19:20 | TPE | 1 | 20:50 | 90 | AB6 |
| 282 | 616_2 | HKG | 2 | 19:20 | TPE | 2 | 20:50 | 90 | AB6 |
| 283 | 616_3 | HKG | 3 | 19:20 | TPE | 3 | 20:50 | 90 | AB6 |
| 284 | 616_4 | HKG | 4 | 19:20 | TPE | 4 | 20:50 | 90 | 744 |
| 285 | 616_5 | HKG | 5 | 19:20 | TPE | 5 | 20:50 | 90 | 744 |
| 286 | 616_6 | HKG | 6 | 19:20 | TPE | 6 | 20:50 | 90 | AB6 |
| 287 | 616_7 | HKG | 7 | 19:20 | TPE | 7 | 20:50 | 90 | AB6 |
| 288 | 618_1 | HKG | 1 | 23:15 | TPE | 2 | 00:45 | 90 | AB6 |
| 289 | 618_2 | HKG | 2 | 23:15 | TPE | 3 | 00:45 | 90 | AB6 |
| 290 | 618_3 | HKG | 3 | 23:15 | TPE | 4 | 00:45 | 90 | AB6 |
| 291 | 618_4 | HKG | 4 | 23:15 | TPE | 5 | 00:45 | 90 | AB6 |

| | | | | | | | | | |
|-----|-------|-----|---|-------|-----|---|-------|-----|-----|
| 292 | 618_5 | HKG | 5 | 23:15 | TPE | 6 | 00:45 | 90 | AB6 |
| 293 | 618_6 | HKG | 6 | 23:15 | TPE | 7 | 00:45 | 90 | AB6 |
| 294 | 618_7 | HKG | 7 | 23:15 | TPE | 1 | 00:45 | 90 | AB6 |
| 295 | 631_1 | TPE | 1 | 08:10 | MNL | 1 | 10:10 | 120 | AB6 |
| 296 | 631_2 | TPE | 2 | 08:10 | MNL | 2 | 10:10 | 120 | AB6 |
| 297 | 631_3 | TPE | 3 | 08:10 | MNL | 3 | 10:10 | 120 | AB6 |
| 298 | 631_4 | TPE | 4 | 08:10 | MNL | 4 | 10:10 | 120 | AB6 |
| 299 | 631_5 | TPE | 5 | 08:10 | MNL | 5 | 10:10 | 120 | AB6 |
| 300 | 631_6 | TPE | 6 | 08:10 | MNL | 6 | 10:10 | 120 | AB6 |
| 301 | 631_7 | TPE | 7 | 08:10 | MNL | 7 | 10:10 | 120 | AB6 |
| 302 | 635_1 | TPE | 1 | 13:00 | MNL | 1 | 15:00 | 120 | 734 |
| 303 | 635_2 | TPE | 2 | 13:00 | MNL | 2 | 15:00 | 120 | 734 |
| 304 | 635_3 | TPE | 3 | 13:00 | MNL | 3 | 15:00 | 120 | 734 |
| 305 | 635_4 | TPE | 4 | 13:00 | MNL | 4 | 15:00 | 120 | 734 |
| 306 | 635_5 | TPE | 5 | 13:00 | MNL | 5 | 15:00 | 120 | 734 |
| 307 | 635_6 | TPE | 6 | 13:00 | MNL | 6 | 15:00 | 120 | 734 |
| 308 | 635_7 | TPE | 7 | 13:00 | MNL | 7 | 15:00 | 120 | 734 |
| 309 | 641_1 | TPE | 1 | 09:00 | BKK | 1 | 14:30 | 330 | 742 |
| 310 | 641_2 | TPE | 2 | 09:00 | BKK | 2 | 14:30 | 330 | 742 |
| 311 | 641_3 | TPE | 3 | 09:00 | BKK | 3 | 14:30 | 330 | 742 |
| 312 | 641_4 | TPE | 4 | 09:00 | BKK | 4 | 14:30 | 330 | 742 |
| 313 | 641_5 | TPE | 5 | 09:00 | BKK | 5 | 14:30 | 330 | 742 |
| 314 | 641_6 | TPE | 6 | 09:00 | BKK | 6 | 14:30 | 330 | 742 |
| 315 | 641_7 | TPE | 7 | 09:00 | BKK | 7 | 14:30 | 330 | 742 |
| 316 | 695_1 | TPE | 1 | 12:30 | BKK | 1 | 16:10 | 220 | AB6 |
| 317 | 695_2 | TPE | 2 | 12:30 | BKK | 2 | 16:10 | 220 | AB3 |
| 318 | 695_3 | TPE | 3 | 12:30 | BKK | 3 | 16:10 | 220 | AB6 |
| 319 | 695_4 | TPE | 4 | 12:30 | BKK | 4 | 16:10 | 220 | AB6 |
| 320 | 695_5 | TPE | 5 | 12:30 | BKK | 5 | 16:10 | 220 | AB3 |
| 321 | 695_6 | TPE | 6 | 12:30 | BKK | 6 | 16:10 | 220 | AB6 |
| 322 | 695_7 | TPE | 7 | 12:30 | BKK | 7 | 16:10 | 220 | AB6 |
| 323 | 645_2 | TPE | 2 | 09:30 | HKT | 2 | 14:00 | 270 | AB3 |
| 324 | 645_3 | TPE | 3 | 09:30 | HKT | 3 | 14:00 | 270 | AB3 |
| 325 | 645_4 | TPE | 4 | 09:30 | HKT | 4 | 14:00 | 270 | AB3 |
| 326 | 645_6 | TPE | 6 | 09:30 | HKT | 6 | 14:00 | 270 | AB3 |
| 327 | 645_7 | TPE | 7 | 09:30 | HKT | 7 | 14:00 | 270 | AB3 |
| 328 | 691_1 | TPE | 1 | 09:30 | HKT | 1 | 14:00 | 270 | AB3 |

| | | | | | | | | | |
|-----|-------|-----|---|-------|-----|---|-------|-----|-----|
| 329 | 691_5 | TPE | 5 | 09:30 | HKT | 5 | 14:00 | 270 | AB3 |
| 330 | 651_1 | TPE | 1 | 07:25 | KUL | 1 | 14:05 | 400 | AB3 |
| 331 | 651_2 | TPE | 2 | 07:25 | KUL | 2 | 14:05 | 400 | AB3 |
| 332 | 651_3 | TPE | 3 | 07:25 | KUL | 3 | 14:05 | 400 | AB3 |
| 333 | 651_4 | TPE | 4 | 07:25 | KUL | 4 | 14:05 | 400 | AB3 |
| 334 | 651_5 | TPE | 5 | 07:25 | KUL | 5 | 14:05 | 400 | AB3 |
| 335 | 651_6 | TPE | 6 | 07:25 | KUL | 6 | 14:05 | 400 | AB3 |
| 336 | 651_7 | TPE | 7 | 07:25 | KUL | 7 | 14:05 | 400 | AB3 |
| 337 | 655_1 | TPE | 1 | 09:30 | KUL | 1 | 14:00 | 270 | AB3 |
| 338 | 655_3 | TPE | 3 | 09:30 | KUL | 3 | 14:00 | 270 | AB3 |
| 339 | 655_4 | TPE | 4 | 09:30 | KUL | 4 | 14:00 | 270 | AB3 |
| 340 | 655_6 | TPE | 6 | 09:30 | KUL | 6 | 14:00 | 270 | AB3 |
| 341 | 655_7 | TPE | 7 | 09:30 | KUL | 7 | 14:00 | 270 | AB3 |
| 342 | 665_1 | TPE | 1 | 08:00 | SIN | 1 | 14:20 | 380 | AB3 |
| 343 | 665_2 | TPE | 2 | 08:00 | SIN | 2 | 14:20 | 380 | AB3 |
| 344 | 665_3 | TPE | 3 | 08:00 | SIN | 3 | 14:20 | 380 | AB3 |
| 345 | 665_4 | TPE | 4 | 08:00 | SIN | 4 | 14:20 | 380 | AB3 |
| 346 | 665_5 | TPE | 5 | 08:00 | SIN | 5 | 14:20 | 380 | AB3 |
| 347 | 665_6 | TPE | 6 | 08:00 | SIN | 6 | 14:20 | 380 | AB3 |
| 348 | 665_7 | TPE | 7 | 08:00 | SIN | 7 | 14:20 | 380 | AB3 |
| 349 | 661_1 | TPE | 1 | 09:15 | SIN | 1 | 13:45 | 270 | AB3 |
| 350 | 661_2 | TPE | 2 | 09:15 | SIN | 2 | 13:45 | 270 | AB3 |
| 351 | 661_3 | TPE | 3 | 09:15 | SIN | 3 | 13:45 | 270 | AB3 |
| 352 | 661_4 | TPE | 4 | 09:15 | SIN | 4 | 13:45 | 270 | AB3 |
| 353 | 661_5 | TPE | 5 | 09:15 | SIN | 5 | 13:45 | 270 | AB3 |
| 354 | 661_6 | TPE | 6 | 09:15 | SIN | 6 | 13:45 | 270 | AB3 |
| 355 | 661_7 | TPE | 7 | 09:15 | SIN | 7 | 13:45 | 270 | AB3 |
| 356 | 675_1 | TPE | 1 | 09:00 | DPS | 1 | 14:15 | 315 | AB6 |
| 357 | 675_2 | TPE | 2 | 09:00 | DPS | 2 | 14:15 | 315 | AB6 |
| 358 | 675_5 | TPE | 5 | 09:00 | DPS | 5 | 14:15 | 315 | AB6 |
| 359 | 675_6 | TPE | 6 | 09:00 | DPS | 6 | 14:15 | 315 | AB6 |
| 360 | 677_2 | TPE | 2 | 08:45 | JKT | 2 | 13:55 | 310 | AB6 |
| 361 | 677_3 | TPE | 3 | 08:45 | JKT | 3 | 13:55 | 310 | AB6 |
| 362 | 677_5 | TPE | 5 | 08:45 | JKT | 5 | 13:55 | 310 | AB6 |
| 363 | 677_7 | TPE | 7 | 08:45 | JKT | 7 | 13:55 | 310 | AB6 |
| 364 | 679_1 | TPE | 1 | 07:00 | JKT | 1 | 14:10 | 430 | AB6 |
| 365 | 679_3 | TPE | 3 | 07:00 | JKT | 3 | 14:10 | 430 | AB6 |

| | | | | | | | | | |
|-----|-------|-----|---|-------|-----|---|-------|-----|-----|
| 366 | 679_4 | TPE | 4 | 07:00 | JKT | 4 | 14:10 | 430 | AB6 |
| 367 | 679_6 | TPE | 6 | 07:00 | JKT | 6 | 14:10 | 430 | AB6 |
| 368 | 681_1 | TPE | 1 | 07:20 | SGN | 1 | 10:50 | 210 | AB6 |
| 369 | 681_2 | TPE | 2 | 07:20 | SGN | 2 | 10:50 | 210 | AB6 |
| 370 | 681_3 | TPE | 3 | 07:20 | SGN | 3 | 10:50 | 210 | AB6 |
| 371 | 681_4 | TPE | 4 | 07:20 | SGN | 4 | 10:50 | 210 | AB6 |
| 372 | 681_5 | TPE | 5 | 07:20 | SGN | 5 | 10:50 | 210 | AB6 |
| 373 | 681_6 | TPE | 6 | 07:20 | SGN | 6 | 10:50 | 210 | AB6 |
| 374 | 681_7 | TPE | 7 | 07:20 | SGN | 7 | 10:50 | 210 | AB6 |
| 375 | 632_1 | MNL | 1 | 11:10 | TPE | 1 | 13:10 | 120 | AB6 |
| 376 | 632_2 | MNL | 2 | 11:10 | TPE | 2 | 13:10 | 120 | AB6 |
| 377 | 632_3 | MNL | 3 | 11:10 | TPE | 3 | 13:10 | 120 | AB6 |
| 378 | 632_4 | MNL | 4 | 11:10 | TPE | 4 | 13:10 | 120 | AB6 |
| 379 | 632_5 | MNL | 5 | 11:10 | TPE | 5 | 13:10 | 120 | AB6 |
| 380 | 632_6 | MNL | 6 | 11:10 | TPE | 6 | 13:10 | 120 | AB6 |
| 381 | 632_7 | MNL | 7 | 11:10 | TPE | 7 | 13:10 | 120 | AB6 |
| 382 | 636_1 | MNL | 1 | 16:00 | TPE | 1 | 18:00 | 120 | 734 |
| 383 | 636_2 | MNL | 2 | 16:00 | TPE | 2 | 18:00 | 120 | 734 |
| 384 | 636_3 | MNL | 3 | 16:00 | TPE | 3 | 18:00 | 120 | 734 |
| 385 | 636_4 | MNL | 4 | 16:00 | TPE | 4 | 18:00 | 120 | 734 |
| 386 | 636_5 | MNL | 5 | 16:00 | TPE | 5 | 18:00 | 120 | 734 |
| 387 | 636_6 | MNL | 6 | 16:00 | TPE | 6 | 18:00 | 120 | 734 |
| 388 | 636_7 | MNL | 7 | 16:00 | TPE | 7 | 18:00 | 120 | 734 |
| 389 | 642_1 | BKK | 1 | 15:30 | TPE | 1 | 20:55 | 325 | 742 |
| 390 | 642_2 | BKK | 2 | 15:30 | TPE | 2 | 20:55 | 325 | 742 |
| 391 | 642_3 | BKK | 3 | 15:30 | TPE | 3 | 20:55 | 325 | 742 |
| 392 | 642_4 | BKK | 4 | 15:30 | TPE | 4 | 20:55 | 325 | 742 |
| 393 | 642_5 | BKK | 5 | 15:30 | TPE | 5 | 20:55 | 325 | 742 |
| 394 | 642_6 | BKK | 6 | 15:30 | TPE | 6 | 20:55 | 325 | 742 |
| 395 | 642_7 | BKK | 7 | 15:30 | TPE | 7 | 20:55 | 325 | 742 |
| 396 | 696_1 | BKK | 1 | 17:10 | TPE | 1 | 20:40 | 210 | AB6 |
| 397 | 696_2 | BKK | 2 | 17:10 | TPE | 2 | 20:40 | 210 | AB3 |
| 398 | 696_3 | BKK | 3 | 17:10 | TPE | 3 | 20:40 | 210 | AB6 |
| 399 | 696_4 | BKK | 4 | 17:10 | TPE | 4 | 20:40 | 210 | AB6 |
| 400 | 696_5 | BKK | 5 | 17:10 | TPE | 5 | 20:40 | 210 | AB3 |
| 401 | 696_6 | BKK | 6 | 17:10 | TPE | 6 | 20:40 | 210 | AB6 |
| 402 | 696_7 | BKK | 7 | 17:10 | TPE | 7 | 20:40 | 210 | AB6 |

| | | | | | | | | | |
|-----|-------|-----|---|-------|-----|---|-------|-----|-----|
| 403 | 646_2 | HKT | 2 | 15:00 | TPE | 2 | 19:25 | 265 | AB3 |
| 404 | 646_3 | HKT | 3 | 15:00 | TPE | 3 | 19:25 | 265 | AB3 |
| 405 | 646_4 | HKT | 4 | 15:00 | TPE | 4 | 19:25 | 265 | AB3 |
| 406 | 646_6 | HKT | 6 | 15:00 | TPE | 6 | 19:25 | 265 | AB3 |
| 407 | 646_7 | HKT | 7 | 15:00 | TPE | 7 | 19:25 | 265 | AB3 |
| 408 | 692_1 | HKT | 1 | 15:00 | TPE | 1 | 19:25 | 265 | AB3 |
| 409 | 692_5 | HKT | 5 | 15:00 | TPE | 5 | 19:25 | 265 | AB3 |
| 410 | 656_1 | KUL | 1 | 15:00 | TPE | 1 | 19:30 | 270 | AB3 |
| 411 | 656_3 | KUL | 3 | 15:00 | TPE | 3 | 19:30 | 270 | AB3 |
| 412 | 656_4 | KUL | 4 | 15:00 | TPE | 4 | 19:30 | 270 | AB3 |
| 413 | 656_6 | KUL | 6 | 15:00 | TPE | 6 | 19:30 | 270 | AB3 |
| 414 | 656_7 | KUL | 7 | 15:00 | TPE | 7 | 19:30 | 270 | AB3 |
| 415 | 652_1 | KUL | 1 | 15:25 | TPE | 1 | 21:50 | 385 | AB3 |
| 416 | 652_2 | KUL | 2 | 15:25 | TPE | 2 | 21:50 | 385 | AB3 |
| 417 | 652_3 | KUL | 3 | 15:25 | TPE | 3 | 21:50 | 385 | AB3 |
| 418 | 652_4 | KUL | 4 | 15:25 | TPE | 4 | 21:50 | 385 | AB3 |
| 419 | 652_5 | KUL | 5 | 15:25 | TPE | 5 | 21:50 | 385 | AB3 |
| 420 | 652_6 | KUL | 6 | 15:25 | TPE | 6 | 21:50 | 385 | AB3 |
| 421 | 652_7 | KUL | 7 | 15:25 | TPE | 7 | 21:50 | 385 | AB3 |
| 422 | 662_1 | SIN | 1 | 14:55 | TPE | 1 | 19:15 | 260 | AB3 |
| 423 | 662_2 | SIN | 2 | 14:55 | TPE | 2 | 19:15 | 260 | AB3 |
| 424 | 662_3 | SIN | 3 | 14:55 | TPE | 3 | 19:15 | 260 | AB3 |
| 425 | 662_4 | SIN | 4 | 14:55 | TPE | 4 | 19:15 | 260 | AB3 |
| 426 | 662_5 | SIN | 5 | 14:55 | TPE | 5 | 19:15 | 260 | AB3 |
| 427 | 662_6 | SIN | 6 | 14:55 | TPE | 6 | 19:15 | 260 | AB3 |
| 428 | 662_7 | SIN | 7 | 14:55 | TPE | 7 | 19:15 | 260 | AB3 |
| 429 | 666_1 | SIN | 1 | 15:25 | TPE | 1 | 21:55 | 390 | AB3 |
| 430 | 666_2 | SIN | 2 | 15:25 | TPE | 2 | 21:55 | 390 | AB3 |
| 431 | 666_3 | SIN | 3 | 15:25 | TPE | 3 | 21:55 | 390 | AB3 |
| 432 | 666_4 | SIN | 4 | 15:25 | TPE | 4 | 21:55 | 390 | AB3 |
| 433 | 666_5 | SIN | 5 | 15:25 | TPE | 5 | 21:55 | 390 | AB3 |
| 434 | 666_6 | SIN | 6 | 15:25 | TPE | 6 | 21:55 | 390 | AB3 |
| 435 | 666_7 | SIN | 7 | 15:25 | TPE | 7 | 21:55 | 390 | AB3 |
| 436 | 676_1 | DPS | 1 | 15:20 | TPE | 1 | 20:30 | 310 | AB6 |
| 437 | 676_2 | DPS | 2 | 15:20 | TPE | 2 | 20:30 | 310 | AB6 |
| 438 | 676_5 | DPS | 5 | 15:20 | TPE | 5 | 20:30 | 310 | AB6 |
| 439 | 676_6 | DPS | 6 | 15:20 | TPE | 6 | 20:30 | 310 | AB6 |

| | | | | | | | | | |
|-----|-------|-----|---|-------|-----|---|-------|-----|-----|
| 440 | 678_2 | JKT | 2 | 14:55 | TPE | 2 | 20:05 | 310 | AB6 |
| 441 | 678_3 | JKT | 3 | 14:55 | TPE | 3 | 20:05 | 310 | AB6 |
| 442 | 678_5 | JKT | 5 | 14:55 | TPE | 5 | 20:05 | 310 | AB6 |
| 443 | 678_7 | JKT | 7 | 14:55 | TPE | 7 | 20:05 | 310 | AB6 |
| 444 | 680_1 | JKT | 1 | 15:20 | TPE | 1 | 22:30 | 430 | AB6 |
| 445 | 680_3 | JKT | 3 | 15:20 | TPE | 3 | 22:30 | 430 | AB6 |
| 446 | 680_4 | JKT | 4 | 15:20 | TPE | 4 | 22:30 | 430 | AB6 |
| 447 | 680_6 | JKT | 6 | 15:20 | TPE | 6 | 22:30 | 430 | AB6 |
| 448 | 682_1 | SGN | 1 | 12:00 | TPE | 1 | 15:20 | 200 | AB6 |
| 449 | 682_2 | SGN | 2 | 12:00 | TPE | 2 | 15:20 | 200 | AB6 |
| 450 | 682_3 | SGN | 3 | 12:00 | TPE | 3 | 15:20 | 200 | AB6 |
| 451 | 682_4 | SGN | 4 | 12:00 | TPE | 4 | 15:20 | 200 | AB6 |
| 452 | 682_5 | SGN | 5 | 12:00 | TPE | 5 | 15:20 | 200 | AB6 |
| 453 | 682_6 | SGN | 6 | 12:00 | TPE | 6 | 15:20 | 200 | AB6 |
| 454 | 682_7 | SGN | 7 | 12:00 | TPE | 7 | 15:20 | 200 | AB6 |

*：乃指當地時間

簡 歷



中文姓名：王國琛

英文姓名：Kuo-Chen Wang

籍 貫：台灣省台中縣

出生日期：民國 67 年 8 月 22 日

聯絡地址：台中縣沙鹿鎮犁份里七賢路 372 號

聯絡電話：04-26624821

E-mail：pumma@mail2000.com.tw

簡 歷：

民國 91 年 6 月 國立交通大學運輸科技與管理學系碩士班畢業

民國 89 年 6 月 國立交通大學運輸工程與管理學系畢業

民國 85 年 6 月 台灣省立清水高級中學

民國 82 年 6 月 台中縣立鹿寮國民中學畢業

民國 79 年 6 月 台中縣立竹林國民小學畢業