

國立交通大學

交通運輸研究所

碩士論文

應用 Max-Plus 代數分析鐵路時刻表穩定性  
Applying Max-Plus Algebra to Analyze Stability of Railway  
Timetables



研究生: 邱戊吉

指導教授: 黃台生 老師

中華民國九十九年六月



應用 Max-Plus 代數分析鐵路時刻表穩定性

Applying Max-Plus Algebra to Analyze Stability of Railway Timetables

研 究 生: 邱 戊 吉

Student: Chiu, Wu-Chi

指 導 教 授: 黃 台 生

Advisor: Huang, Tai-Sheng

國立交通大學

交通運輸研究所

碩士論文



A Thesis

Submitted to Institute of Traffic and Transportation

College of Management Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Science

June 2010

Taipei, Taiwan, Republic of China

中華民國九十九年六月



本論文所引用之各商標及商品名稱分屬其合法註冊公司所有，絕無侵權之意，特此聲明。



Some rights reserved.

本著作使用 Creative Commons 「姓名標示-非商業性-相同方式分享」授權條款台灣3.0版，授權您可自由重製、散布、展示及演出本著作及創作衍生著作，惟需遵照下列條件：

- **姓名標示** 您必須按照作者或授權人所指定的方式，保留其姓名標示。
- **非商業性** 您不得為商業目的而使用本著作。
- **相同方式分享** 若您改變、轉變或改作本著作，僅在遵守與本著作相同的授權條款下，您始得散布由本著作而生的衍生著作。

使用或散布本著作時，您必須向他人清楚說明本著作所適用的授權條款。如果您取得著作權人之許可，這些條件中任一項都能被免除。您合理使用的權利及其他的權利，不因上述內容而受影響。更詳細的授權條款請見附件光碟或 <http://www.creativecommons.org.tw>。

附件光碟中之程式與程式碼以 GNU General Public License Version 2 授權，使用前請務必詳閱附件光碟隨附之授權條款以瞭解授權之範圍與限制。

本論文全文以  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}/\text{c}w\text{T}_{\text{E}}\text{X}$  排版製作。

# 應用 Max-Plus 代數分析鐵路時刻表穩定性

學生: 邱戊吉

指導教授: 黃台生

國立交通大學交通運輸研究所碩士班

## 摘要

在列車高度密集的鐵路系統中, 列車的延誤經常會影響到其他列車, 如何防止延誤不斷地擴散是鐵路經營管理的重要課題。時刻表穩定性指的是當延誤發生時系統恢復準點的能力, 分析穩定性有助於延誤的管理。Max-plus 代數可以用於分析鐵路時刻表穩定性, 但過去這方面的研究都假設時刻表為週期時刻表, 本研究則去除這個假設以應用於非週期時刻表。

本研究將鐵路系統模化為 max-plus 線性系統, 證明可運行的時刻表系統可以求解並能計算出餘裕矩陣與穩定商數, 餘裕矩陣和穩定商數可以用於衡量時刻表穩定性。本研究也比較了兩種求解 max-plus 系統的演算法, 分別給出其複雜度。本研究以臺鐵基隆-新竹路段之時刻表為實例進行穩定性分析, 最後指出穩定性分析可以應用於時刻表選擇及調整、運轉整理、設施改善方案評估以及列車服務可靠度分析。

# Applying Max-Plus Algebra to Analyze Stability of Railway Timetables

Student: Chiu, Wu-Chi

Advisor: Huang, Tai-Sheng

Institute of Traffic and Transportation  
National Chiao Tung University

## Abstract

Delay of a train would cause delay of another train in dense railway traffic. How to prevent the spread of delays is an important issue of railway management. In railway systems, stability of railway timetable means the ability of recovery from delays. Stability analysis can help management of delays. Max-plus Algebra can be used to analyze the stability of timetables, but previous researches assumed that the timetable is periodic. This thesis drops the periodicity to make the method can apply to non-periodic timetables.

Railway systems can be modeled as max-plus linear systems, and a system of operational timetable can be solved and obtain the slack matrix and stability quotients, which can be used to measure the stability. This thesis describes two algorithms to solving max-plus linear systems and gives the complexity of these algorithms. A case-study of the Taiwan Railway Administration on Keelung–Hsinchu section is implemented. Finally, this method can be used to timetable selection and adjustment, rescheduling, assessment of infrastructure improvement and analysis of reliability.

## 誌 謝

二年前在各種因緣巧合之下誤打誤撞進到交研所是我始料未及的。然而一路走到現在能夠順利完成學業，有太多人要感謝。

首先要感謝我的指導教授黃教授台生，不論是課堂上的授業還是論文寫作時的悉心指導，都令我受益良多。在論文寫作的過程中，與老師的討論讓我更能把握論文主題，澄清了許多概念。還有也要感謝老師包容我在排版上的堅持，我才能拋開 MS 的難用軟體。

感謝二位口試委員王局長國材與賴教授勇成的寶貴意見，特別要感謝賴老師提示我穩定性分析這個研究主題，以及幫忙取得許多參考資料。在交研所的這段期間，謝謝所上所有老師的教導與關心，也從各位老師的課堂上學到許多。還要感謝所辦洪姐、柳姐與何姐在行政程序上的協助和平日的照顧，以及班上所有同學的加油打氣。

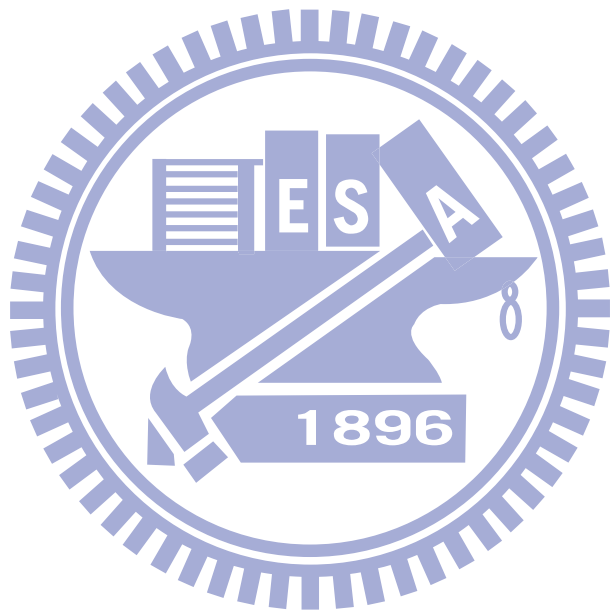
這本論文能夠編排完成要感謝 Donald Knuth, Leslie Lamport 和吳聰敏以及所有參與發展  $\text{TeX/LaTeX/cwTeX}$  的人員開發出如此完善的排版系統。我無法想像如果沒有這套系統，我要花費多少時間與心力排版大量的數學式與交互參照，還有版面會多麼慘不忍睹。

研究過程中，難免遭遇許多疑難雜症，從數學問題到電腦當機不一而足，感謝所有費心幫忙解決困難的朋友們。也要謝謝一路陪伴在我身邊的家人與朋友，謝謝你們的支持和鼓勵。

謹以此論文獻給我最摯愛的朋友

邱戊吉

庚寅仲夏 謹誌於三峽寓所

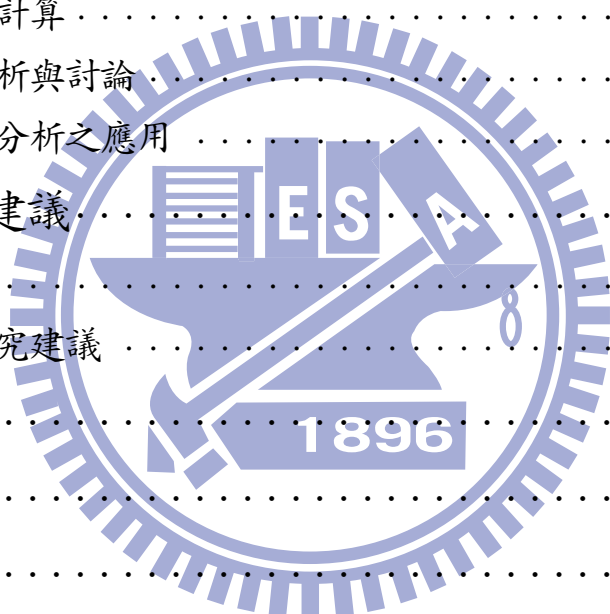




# 目 錄

摘要	i
英文摘要	ii
誌謝	iii
目錄	v
表目錄	vii
圖目錄	vii
符號表	viii
第一章 緒論	1
1.1 研究背景與動機	1
1.2 研究目的與課題	2
1.3 研究範圍與限制	2
1.4 研究架構	3
1.5 研究方法與流程	3
第二章 文獻回顧	5
2.1 鐵路系統穩定性、可靠度和穩健度的相關研究	5
2.2 Max-Plus 代數理論簡介	6
2.3 週期時刻表的穩定性分析	9
2.4 臺鐵之相關研究	16
第三章 鐵路系統的穩定性	19
3.1 鐵路系統的組成與運轉	19
3.2 列車時刻表的排班限制	20
3.3 延誤與餘裕	21
3.4 列車時刻表的穩定性	22

第四章	列車時刻表穩定性分析	25
4.1	以 Max-Plus 代數模化列車時刻表	25
4.2	$\mathbb{R}_{\max}^n$ 中的序關係	29
4.3	求解列車時刻表系統	32
4.4	餘裕矩陣與穩定性指標	35
4.5	有關演算法的討論	36
4.6	穩定性方析方法比較	38
第五章	實例研究與應用	39
5.1	研究範圍概況	39
5.2	模化與計算	39
5.3	結果分析與討論	42
5.4	穩定性分析之應用	46
第六章	結論與建議	47
6.1	結論	47
6.2	後續研究建議	47
參考文獻		49
索引		53
附件		54



# 表 目 錄

表 4.1	虛擬時刻表	28
表 5.1	臺鐵基隆-新竹段各車站軌道數資料	40
表 5.2	餘裕矩陣示意	43
表 5.3	事件說明、穩定商數與容許延誤示意表	43
表 5.4	各車站穩定商數平均值	45

# 圖 目 錄

圖 1.1	研究架構	3
圖 1.2	研究流程	4
圖 2.1	範例路網示意圖	11
圖 2.2	Timed event graph	12
圖 2.3	列車追越時產生的問題	17
圖 3.1	穩定平衡與不穩定平衡示意圖	22
圖 4.1	虛擬時刻表之 precedence graph	30
圖 5.1	帶有穩定性標記的運行圖	45

# 符號表

以下列出本論文所使用之符號及簡短的解釋，各符號定義之細節請參閱內文。另外在文獻回顧時，符號均依照原始論文的用法，可能與本列表有所不同。

$\oplus$	Max-plus 代數的運算，定義為 $a \oplus b = \max(a, b)$	$e$	Max-plus 代數之單位元
$\otimes$	Max-plus 代數的運算，定義為 $a \otimes b = a + b$	$E$	時刻表系統所有事件的集合
$a_{ij}$	Max-plus 代數矩陣的元	$E_i$	時刻表系統的事件
$A$	Max-plus 代數矩陣，也表示時刻表系統	$\varepsilon$	Max-plus 代數的零元
$A^+$	$\stackrel{\text{def}}{=} \lim_{k \rightarrow \infty} A \oplus A^2 \oplus \dots \oplus A^k$	$\mathcal{E}, \mathcal{E}_n$	Max-plus 代數的零矩陣
$A^*$	$A$ 的擬反矩陣	$G(A)$	$A$ 的 precedence graph
$A^k$	$A$ 的 $k$ 次方	$I, I_n$	Max-plus 代數的單位矩陣
$A^{(k)}$	$\stackrel{\text{def}}{=} I \oplus A \oplus A^2 \oplus \dots \oplus A^k$	$i, j, k$	區別不同事件之下標
$A^T$	矩陣 $A$ 的轉置	$\mathcal{M}_n(\mathbb{R}_{\max})$	Max-plus 矩陣集
$\mathbf{b}, \mathbf{y}$	$\mathbb{R}_{\max}^n$ 的向量	$\mathbb{N}$	自然數集
$d_i$	事件的表定時刻	$O(n)$ 等	Big O 記號, 見[14]
$\mathbf{d}$	時刻表向量	$r_{ij}$	餘裕矩陣的元
$\delta_i$	事件 $E_i$ 的延誤量	$R$	餘裕矩陣
$\Delta$	$\stackrel{\text{def}}{=} [\delta_1 \otimes d_1, \dots, \delta_n \otimes d_n]^T$	$\mathbb{R}$	實數集
		$S$	定義為 $s_{ij} = d_j - d_i + a_{ij}$
		$x_i$	事件的發生時刻
		$\mathbf{x}$	發生時刻向量

# 第一章 緒論

## 1.1 研究背景與動機

現代化的軌道運輸以其安全、準時、高運量等特性，在許多城市及地區擔負起主要大眾運具的角色。在臺灣，高鐵已經是西部地區南來北往的大動脈；臺鐵除了是東部地區最重要的交通工具之一，在西部都會區也正朝捷運化的方向轉型；而臺北及高雄的捷運系統讓都市內交通更加方便；機場捷運和臺中捷運也將在幾年後加入營運的行列。可以說，軌道運輸在臺灣運輸系統中之角色將會愈加重要。

時刻表編製是軌道營運中相當重要的一環，時刻表有以下的功能[22]:

1. 使設施設備達到最佳運用。
2. 讓列車運轉行為可以預測。
3. 提供旅客列車到發資訊。
4. 作為行車控制、車輛人員排班的依據。

當軌道運轉系統車次密集時，列車在路線上發生衝突的機會也就增加；旅客量多的時候列車也容易因等候旅客上下車而產生延誤。在列車密集的軌道運轉系統裡，可謂是牽一髮而動全身。基於安全的考量，列車的運轉控制有閉塞區間、最小時隔、速限等諸多方式，因此任何延誤都有可能造成許多其他列車也隨之延誤。一旦列車延誤相當嚴重時，這個時刻表就無法達成前述的功能，營運者無法預測列車的運轉，旅客也無從得知正確的發車資訊。要防止列車延誤不斷地擴大，解決之道可能是增加路線容量或改善時刻表的排點。但是前者要投入相當多的資源以建置基礎設施或增備車輛人員，相較之下改善時刻表就成了比較經濟迅速的作法。因此編製一個「穩定」的時刻表非常重要。

「穩定」依 Goverde 的定義是指：當軌道運輸系統中發生有限的延誤後，能夠在有限的時間內回復到準點狀態[20]。當然如果愈快回復到準點，就可以認為這個系統愈穩定。而準點是指列車的發車時間或到達時間與時刻表所定的時間相同，因此在不考慮其他使列車誤點的因素時，穩定的時刻表就是一個使軌道運輸系統列車運行穩定的時刻表。如何知道一個時刻表穩不穩定呢？直覺上來看，如果在排點時給每一列車預留比較多的餘裕時間，那麼這個時刻表就比較穩定。或者換個角度來看，路線容量的利用率低，則這個時刻表比較穩定。

臺灣在這方面的研究比較少，周學怡以模擬的方法來分析臺鐵列車運行可靠性[1]，不過這篇研究中是分析列車數少與路線容量低的南迴線，雖然該研究的計算中將列車數增加到容量上

限，但是作者並沒有進行演算法複雜度分析，如果用到西部幹線上則不知能否在短時間內模擬龐大的可能狀況。穩定性與容量的關係歐洲學者做了不少研究，Mattsson 對相關的成果做了一份整理[25]，大部分研究用的方法主要有解析模式和微觀模擬兩種。De Kort 研究荷蘭的一段高速鐵路，在給定可靠度要求(reliability requirement) 下，以隨機 max-plus 方法計算出路段的容量[15]。不過因為例子的規模比較小，列車也不像臺鐵一般有複雜的組成，要直接應用於臺灣的鐵路比較難。Vromans 則考慮異質的(heterogeneous) 列車組成，定義了兩種列車同質性(homogeneity) 的指標，並利用模擬得到同質性指標愈高則鐵路系統愈可靠的結論[31]。Salido 也定義了幾種指標用於比較時刻表的穩健度(robustness)，不過這些指標都是以直觀的想法定義，比較缺乏實證支持[30]。Delorme 則是用最短路徑的方法來分析一個站內列車延誤時造成的影響，是一種局部的穩定性分析[16, 17]。Goverde 則以 max-plus 代數來分析列車時刻表的穩定性[19, 20]，

上述幾篇文獻中最令人感興趣的是 Goverde 的作法。Goverde 把鐵路系統以 max-plus 代數模化成一個線性系統來求解，進而得到一系列與列車穩定性有關的指標，似乎還沒有見過這套方法應用在國內的軌道研究。不過 Goverde 的方法還是有些不足之處，主要的問題是該研究中假設時刻表是週期時刻表(periodic timetable, 或譯為定型化時刻表)，然而傳統鐵路常以一天以上的週期排班，延誤要影響到下一個週期的機會不大，因此在這個假設下的分析結果有其運用上的侷限。基於上述的問題，本研究希望在前人文獻的基礎上進一步發展，以 max-plus 代數建立一個能應用於臺灣鐵路系統的時刻表穩定性分析方法。

## 1.2 研究目的與課題

本研究之目的是在 Goverde 的研究成果之上，以 max-plus 代數建立一個能應用於臺灣鐵路系統的時刻表穩定性分析方法。具體的研究課題如下：

1. 了解鐵路運轉原理。
2. 回顧列車時刻表穩定性與 max-plus 代數的相關研究，了解以 max-plus 代數分析列車時刻表穩定性的基本原理。
3. 建立一套適用於臺灣列車時刻表穩定性分析之方法。
4. 以臺鐵系統為例，分析時刻表穩定性之狀況及相關特性，並提出相關的應用。

## 1.3 研究範圍與限制

本研究擬以傳統鐵路系統為研究對象，也將以臺鐵基隆—新竹路段為實例進行列車時刻表



的穩定性分析，因為臺鐵的系統較為完整，其軌道與月台配置、列車組成、行車控制方式均較捷運與高鐵多樣完整，研究結果代表性較高，也可以應用到捷運和高鐵。

從相關資料中可知鐵路系統大部分的延誤並不大<sup>1</sup>，而且運轉整理也不在本研究的範圍之內，因此本研究假定延誤量均不大所以不用進行運轉整理，也就是說不改變列車運行的順序。

## 1.4 研究架構

本研究的架構如圖 1.1 所示，其核心為列車運行模式之模化和時刻表穩定性分析。列車運行必須依據鐵路列車的運行限制和編製完成之時刻表來模化，建構列車運行模式和穩定性分析的方法以 max-plus 代數為基礎，利用 max-plus 線性系統來模化列車的運行，並藉由求解該線性系統來分析列車時刻表的穩定性，進一步可以分析列車延誤事件對鐵路系統的影響。

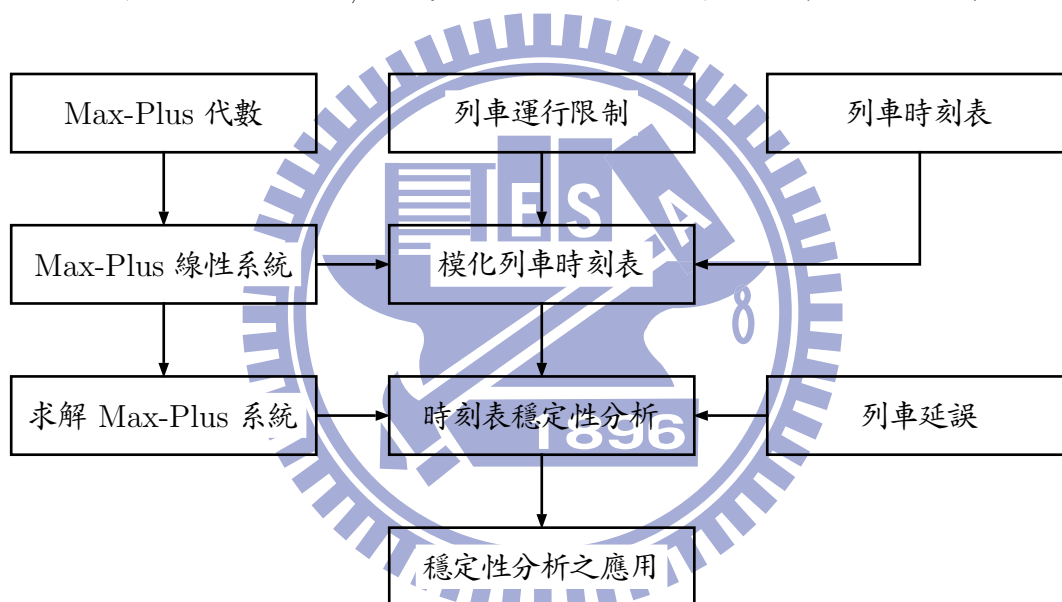


圖 1.1: 研究架構

## 1.5 研究方法與流程

本研究以 max-plus 代數理論建立鐵路列車時刻表的穩定性分析方法。研究的工作從蒐集的相關資料和文獻開始，了解鐵路系統的運轉規則，儘量使模化結果可以反應實際狀況。另一方面則要探討 max-plus 代數理論，並了解如何以 max-plus 理論模化列車時刻表。

在資料與理論的基礎上，將以 max-plus 代數理論發展一套適用於傳統鐵路系統的穩定性分析方法。並從臺鐵的資料中決定實作時要考慮的項目及限制，然後撰寫程式實作此方法。利用

<sup>1</sup>例如參見[5]第20頁以下的表格，不過從中也可以看出臺鐵延誤的發生頻率有多麼頻繁。

該程式以現有的資料進行臺鐵時刻表之穩定性分析，並提出穩定性分析的應用方向。最後是本研究的結論和後續研究之建議。

本研究的流程可以用圖 1.2 表示。

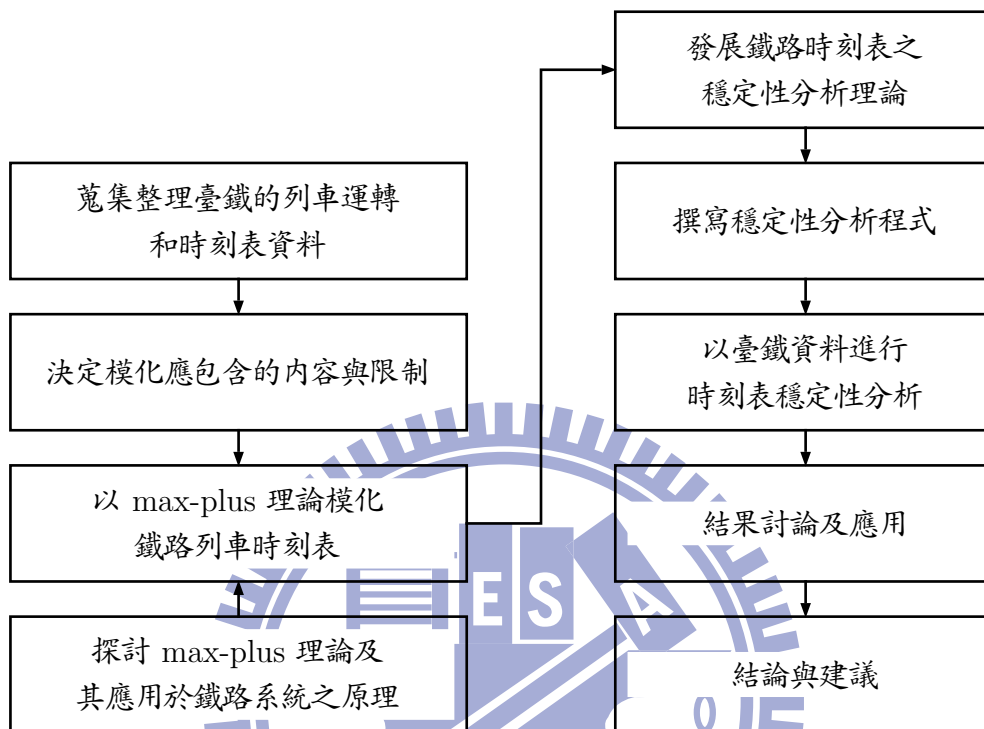


圖 1.2: 研究流程



## 第二章 文獻回顧

### 2.1 鐵路系統穩定性、可靠度和穩健度的相關研究

在這一節中將回顧鐵路系統穩定性、可靠度和穩健度的相關研究，三個概念雖然有所不同，但是也有許多類似的地方，所以在此一併討論。許多成果都是歐洲學者的研究，Mattsson 整理了一些相關成果，指出列車穩定性的研究方法主要有解析模式和微觀模擬兩種[25]。在相關研究中，每個作者用語並不一致，常見的用語有 stability、robustness、reliability 等，但是即使用同樣的字彙，但是其中的概念卻不一定相同。例如 reliability 也常用於路網的可靠度，指的是當路網在路線中斷時保持運能的程度。需要指出的是本文中的穩定性為鐵路系統在發生延誤時回復準點之能力或可能性，而回顧文獻時則保持原作者的用語。

用模擬來研究鐵路系統已經發展相當長的一段時間，也有不少鐵路系統模擬的商業軟體。例如 Simone 就是一個可以用來分析列車時刻表穩定性的軟體，在輸入路網和時刻表及初始延誤等參數後，它能以模擬的方法計算延誤傳播的狀況[27]。用不同的初始延誤重覆模擬程序，統計結果後就能得到鐵路系統的準點率，準點率可以作為穩定性的一個指標。

國內周學怡之研究也是用模擬來探討列車運行的可靠性，這篇文章中作者模擬了列車的跟車、錯車、會車和低級列車等候高級列車等規則。在假設列車延誤的分佈下，作者進行了100次模擬，也是統計準點率來分析時刻表的可靠性[1]。這篇研究中的例子是分析列車數少的南迴線，即使作者考慮到南迴線列車數極少，以各車種同比例增加的方式將路線利用率調整成100%，雙向合計仍只有78列車。且作者並沒有對其演算法進行複雜度分析，如果用到西部幹線上不知道能否在合理時間內模擬龐大的運行系統。

Vromans 則考慮異質的 (heterogeneous) 列車組成，以模擬的方式分析減少不同車種間速度差的效果，定義了兩種列車同質性 (homogeneity) 的指標，分別是每個站列車的「最小出發間隔時間倒數和」以「及最小抵達間隔時間倒數和」，這兩個值愈大就表示列車的班距愈小。最後作者利用模擬列車運行得到同質性指標愈高則鐵路系統愈可靠的結論[31]。

除了模擬之外，另外一類是以解析方法來分析鐵路系統的穩定性，方法有圖論(graph theory)、等候理論 (queuing theory)、max-plus 代數、統計分析與隨機最佳化 (stochastic optimization) 等幾類。

Delorme 用最短路徑的方法來分析一個站內列車延誤時造成的影響，是一種局部的穩定性分析[16, 17]。文章標題中的“at station level”意思是指穩定度分析在一個局部的範圍來進行

的, 而不是考慮整體的路網。所謂的局部是指在一個車站裡, 從站內配線、號誌轉轍器連鎖、列車行駛路線等可以知道列車間是不是有可能發生衝突(conflict), 然後求出這個衝突會在前車延誤多少時間後才會發生。於是可以在一個站內構造一個圖來表示列車間的關係, 這個圖的頂點是列車, 兩個頂點間存在一有向邊則表示它們可能會有衝突, 邊上的權重代表前列車可以容許的最大延誤, 當延誤超過這個值就會發生衝突。於是用這個圖就可以來計算當某個列車發生了延誤時, 會如何影響到其他的列車。計算的方法當然是在圖中找最短路徑。

Huisman 考慮在一段軌道上有不同速度之列車要競爭軌道的使用權, 列車的抵達是一個隨機過程, 在軌道上的運行時間也是一個機率分布。給定相關的機率分布後計算列車延誤期望值[24]。作者在不同的假設下給出運行時間的解, 可以用來分析鐵路系統在不同的列車數、異質性(即速度組成)、初始延誤、列車順序和緩衝時間對列車延誤的影響。

De Kort 研究荷蘭的一段約100公里高速鐵路[15], 這一段鐵路中有三個長隧道, 最長的一個長達7公里。雖然隧道都是雙線的, 但是為了安全的考量, 不允許隧道中同時有兩列車。在給定可靠度要求(reliability requirement)下, 作者建立了一個隨機 max-plus 方法來計算出該路段的容量。而文中的可靠度要求大概可以解釋成準點率。不過因為例子的規模比較小, 作者不考慮場站的影響, 列車也不像臺鐵一般有複雜的組成, 要應用於臺灣的鐵路比較難。

Goverde 以 max-plus 代數方法研究列車的穩定性[19, 20], 也是本研究主要參考的對象, 內容會在 2.3 節詳述。

## 2.2 Max-Plus 代數理論簡介

Max-plus 代數是 dioid 的一種特例[18], 它的發展是來自於1960年代之圖論研究, 當時是為了解決圖中的路徑尋找問題。而 Baccelli *et al.* 在1993年的經典著作中則闡述了離散動態系統和 max-plus 代數系統的關係, 是 max-plus 代數的一個重要應用方向[13]。另一個應用方向是求解 Hamilton-Jacobi-Bellman 偏微分方程(HJB PDEs), 這類型的 PDE 可用於描述一些非線性控制系統[26], 這與本研究之主題較不相干, 因此就不詳述了。本節將簡介 max-plus 代數理論的基礎, 內容主要整理自[18]和[19]。

**定義 2.1** (Max-Plus 代數). 定義 max-plus 代數  $\mathbb{R}_{\max} = (\mathbb{R} \cup \{-\infty\}, \oplus, \otimes)$  是在集合  $\mathbb{R} \cup \{-\infty\}$  上定義兩個運算:  $a \oplus b = \max(a, b)$ ,  $a \otimes b = a + b$  所構成的代數結構。<sup>1</sup> ||

**定義 2.2** ( $\mathbb{R}_{\max}$  的單位元和零元). 定義  $\mathbb{R}_{\max}$  的零元(zero element)  $\varepsilon = -\infty$  及單位元(identity)  $e = 0$ 。 ||

<sup>1</sup>這個名詞其實會引起一些誤會, 因為這個結構並不是代數理論中的「代數(algebra)」! 而是半體(semifield)。只是文獻中已經普遍如此稱呼, 本論文也只好跟著用了, 有關的細節可以見[18]。

**定義 2.3** ( $\mathbb{R}_{\max}$  上的指數). 若  $a \in \mathbb{R}_{\max}$ ,  $k \in \mathbb{N}$ , 定義  $a^k = \underbrace{a \otimes \cdots \otimes a}_{k \text{ 次}}$ . ||

Max-plus 代數具有許多與實數運算相同的性質, 但也有些性質和實數系大不相同, 這些性質都可以簡單地由定義導出, 下面不加證明地列舉一些有用的性質. 首先是一些和實數系相同的性質:

**性質 2.1.**  $a, b, c \in \mathbb{R}_{\max}$ , 則

1.  $a \oplus (b \oplus c) = (a \oplus b) \oplus c$ ,  $a \otimes (b \otimes c) = (a \otimes b) \otimes c$  (結合律)
2.  $a \oplus b = b \oplus a$ ,  $a \otimes b = b \otimes a$  (交換律)
3.  $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$  (分配律)
4.  $\varepsilon \oplus a = a \oplus \varepsilon = a$
5.  $e \otimes a = a \otimes e = a$

以下這些則是實數系沒有的:

**性質 2.2.**  $a, b \in \mathbb{R}_{\max}$ ,  $k \in \mathbb{N}$ , 則

1.  $a \oplus a = a$  (Idempotency)<sup>2</sup>
2.  $a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon$
3.  $a \oplus b \geq a$ ,  $a \oplus b \geq b$
4.  $(a \oplus b)^k = a^k \oplus b^k$

我們也可以在  $\mathbb{R}_{\max}$  上進行矩陣的運算。

**定義 2.4** (Max-Plus 代數上的矩陣運算). 以下的矩陣每個元均為  $\mathbb{R}_{\max}$  中的元素:

1. 若  $A = (a_{ij}), B = (b_{ij})$  均為  $m \times n$  矩陣, 定義  $A \oplus B$  為  $m \times n$  矩陣, 其元素為

$$[A \oplus B]_{ij} = a_{ij} \oplus b_{ij} = \max(a_{ij}, b_{ij})$$

2. 若  $A = (a_{ij})$  為  $m \times l$  矩陣,  $B = (b_{ij})$  為  $l \times n$  矩陣, 定義  $A \otimes B$  為  $m \times n$  矩陣, 其元素為

$$[A \otimes B]_{ij} = \bigoplus_{k=1}^l a_{ik} \otimes b_{kj} = \max_{k=1, \dots, l} \{a_{ik} + b_{kj}\}$$

3. 記  $\mathcal{M}_n(\mathbb{R}_{\max})$  為所有  $n \times n$  矩陣構成的集合。

<sup>2</sup>這個性質在理論上有舉足輕重的地位, 也是 dioid 與其他代數結構最不一樣之處, 進一步可參閱[13]、[18]。

4. 定義  $\mathcal{E}_n \in \mathcal{M}_n(\mathbb{R}_{\max})$  是  $\oplus$  的單位元, 即  $\forall A \in \mathcal{M}_n(\mathbb{R}_{\max}), A \oplus \mathcal{E}_n = \mathcal{E}_n \oplus A = A$ , 以及  $I_n \in \mathcal{M}_n(\mathbb{R}_{\max})$  是  $\otimes$  的單位元, 即  $\forall A \in \mathcal{M}_n(\mathbb{R}_{\max}), A \otimes I_n = I_n \otimes A = A$ 。用矩陣寫下來就是

$$\mathcal{E}_n = \begin{bmatrix} \varepsilon & \cdots & \varepsilon \\ \vdots & \ddots & \vdots \\ \varepsilon & \cdots & \varepsilon \end{bmatrix}, \quad I_n = \begin{bmatrix} e & \varepsilon & \cdots & \varepsilon \\ \varepsilon & e & \cdots & \varepsilon \\ \vdots & \vdots & \ddots & \vdots \\ \varepsilon & \varepsilon & \cdots & e \end{bmatrix}$$

當矩陣的維度不重要或清楚不致誤解時, 下標  $n$  則省略。

5. 若  $A \in \mathcal{M}_n(\mathbb{R}_{\max}), k \in \mathbb{N}$ , 定義  $A^k = \underbrace{A \otimes \cdots \otimes A}_{k \text{ 次}}$  和  $A^{(k)} = I \oplus A \oplus A^2 \oplus \cdots \oplus A^k$ , 另外定義  $A^0 = A^{(0)} = I_n$

6. 對於  $A \in \mathcal{M}_n(\mathbb{R}_{\max})$  如果  $\bigoplus_{l=1}^{\infty} A^l = A \oplus A^2 \oplus A^3 \oplus \cdots$  收斂, 定義

$$A^+ = \bigoplus_{l=1}^{\infty} A^l = A \oplus A^2 \oplus A^3 \oplus \cdots$$

如果上式的右邊不收斂, 則  $A^+$  沒有意義。 ||

以下討論 max-plus 代數結構在圖論中的意義, 下面兩個定義描述圖和矩陣要如何對應。

**定義 2.5** (相鄰矩陣).  $G$  為一個有  $n$  個頂點的簡單有向圖 (simple directed graph), 每個頂點對應一個從 1 到  $n$  的編號, 且每個邊都給定一個權重。那麼  $G$  的相鄰矩陣 (adjacency matrix)  $A \in \mathcal{M}_n(\mathbb{R}_{\max})$  定義為  $[A]_{ij} = a_{ij} =$  從頂點  $i$  到頂點  $j$  的邊之權重, 如果沒有邊從  $i$  到  $j$  則  $a_{ij} = \varepsilon$ 。另外定義  $a_{ii} = e$ 。 ||

**定義 2.6** ( $A$  的 Precedence graph<sup>3</sup>).  $A = (a_{ij}) \in \mathcal{M}_n(\mathbb{R}_{\max})$ , 定義  $A$  的 Precedence graph  $G(A)$  為有  $n$  個頂點的簡單有向圖, 每個頂點對應於一個從 1 到  $n$  的編號。若  $a_{ij} \neq \varepsilon$ , 則從編號  $j$  的頂點到編號  $i$  的頂點有一個邊權重為  $a_{ij}$ 。 ||

以下的定理說明了 max-plus 代數與最長路徑問題的關係。

**定理 2.1.**  $G$  是有  $n$  個頂點的簡單有向圖,  $A$  是其相鄰矩陣, 則

1. 對任意的正整數  $k$ ,  $[A^k]_{ij}$  是所有從頂點  $j$  到頂點  $i$  恰包含  $k$  個邊的路徑中, 每邊權重總和最大的路徑之權重和。如果  $j$  到  $i$  沒有滿足條件的路徑, 則  $[A^k]_{ij} = \varepsilon$ 。
2. 若  $A^+$  存在, 則  $[A^+]_{ij}$  是從頂點  $j$  到頂點  $i$  的所有路徑中, 每邊權重總和最大的路徑之權重和。如果  $j$  到  $i$  沒有路徑, 則  $[A^+]_{ij} = \varepsilon$ 。

<sup>3</sup>不同文獻有不同用語, 例如 underlying graph, associated graph of  $A$ 。



**證明.** 第一個部分敘述以數學歸納法證之。當  $k = 1$  時，從  $A$  的定義  $a_{ij}$  就是連接  $j$  到  $i$  的邊之權重，因為  $G$  是簡單圖， $j$  到  $i$  最多只有一條邊，所以敘述自然成立。

設  $k = k' > 1$  且  $k' < n$  時敘述成立。當  $k = k' + 1$ ，我們知道

$$[A^{k'+1}]_{ij} = [A^{k'} \otimes A]_{ij} = \bigoplus_{l=1}^n [A^{k'}]_{il} \otimes a_{lj} = \max_{l=1 \dots n} \{[A^{k'}]_{il} + a_{lj}\}$$

若  $[A^{k'}]_{il} \neq \varepsilon$ ，從歸納假設知  $[A^{k'}]_{il}$  就是從  $l$  到  $i$  恰含  $k'$  條邊路徑中權重和最大值，所有從  $j$  到  $i$  恰含  $k' + 1$  條邊的路徑可以分成前  $k'$  條邊和最後一條邊兩部分，因此  $\max_{l=1 \dots n} \{[A^{k'}]_{il} + a_{lj}\}$  就是從頂點  $j$  到頂點  $i$  恰包含  $k' + 1$  個邊的路徑中，每邊權重總和最大的路徑之權重和。在此如果  $a_{lj} = \varepsilon$ ，則  $\max_{l=1 \dots n} \{[A^{k'}]_{il} + a_{lj}\} = \varepsilon$ ，也就是說如果沒有從  $j$  到  $l$  的邊，就沒有從  $j$  到  $l$  再經過  $k'$  條邊連接到  $i$  的路徑。若  $[A^{k'}]_{il} = \varepsilon$ ，則同樣有  $\max_{l=1 \dots n} \{[A^{k'}]_{il} + a_{lj}\} = \varepsilon$ 。綜合以上討論知道對  $k = k' + 1$  敘述也成立，於是由數學歸納法原理可以知道敘述對所有  $k \in \mathbb{N}$  成立。

第二個部分是第一部分的直接推論，由  $A^+$  的定義知  $[A^+]_{ij} = a_{ij} \oplus [A^2]_{ij} \oplus [A^3]_{ij} \oplus \dots$ ，這就是從  $j$  到  $i$  不論包含多少邊的路徑中權重和最大之值。□

在前面的定理中，若把權重解釋為長度，那麼  $A^+$  的元就代表了最長路徑的長度。為了簡化敘述起見，在後文中的權重就直接以長度稱之。

## 2.3 週期時刻表的穩定性分析 1896

本節簡短地討論 Goverde 如何對週期列車時刻表進行穩定性分析，內容是整理自[20]，符號的用法均按原始論文，並不加以更動，而在[19]中有更詳細的討論。除了說明理論的內容之外，也討論其方法的限制，以及不能直接應用在臺鐵系統上的理由。

### 2.3.1 以 Max-Plus 線性系統模化列車時刻表

考慮一個週期時刻表(或稱定型化時刻表)，其週期為  $T$ ，通常是 60 分鐘。時刻表中的事件  $i = (E_i, L_i, S_i)$  是一個三元有序對。 $E_i$  是事件的類別，分成出發、抵達和通過三種，而抵達和通過的時刻可以直接從上一個出發時間加上列車運行時間得到，所以只要考慮出發事件就可以了。而  $L_i$  是事件  $i$  對應的列車路線， $S_i$  是其發生的車站。

令  $x_i(k)$  是事件  $i$  第  $k$  週期的實際出發時刻， $d_i(k)$  是事件  $i$  時刻表中預定的出發時刻。而  $x_i(0)$  和  $d_i(0)$  分別是事件  $i$  的第一個週期實際出發時刻和第一個週期表定出發時刻。

事件必須在滿足若干的限制條件後才能發生。第一個限制條件是列車不可以比表定的時刻早出發，即  $x_i(k) \geq d_i(k)$ 。因為是週期時刻表，所以可以得到  $d_i(k) = d_i(0) + k \cdot T$ ，或是以

max-plus 代數的形式表示為

$$d_i(k) = d_i(0) \otimes T^k$$

其次是列車出發的時刻應大於上一站出發加上站間運轉時間及站內停等時間，若事件  $i$  和  $j$  是同一路線 (即  $L_i = L_j$ ) 中，相鄰二站的出發事件， $j$  在  $i$  的前一站，令  $a_{ij}$  是事件  $j$  於前一站的出發時刻加上站內停等時間，所以這個限制就是  $x_i(k) \geq a_{ij} + x_j(k - \mu_{ij})$ ，或寫成

$$x_i(k) \geq a_{ij} \otimes x_j(k - \mu_{ij})$$

其中的  $\mu_{ij}$  指週期落差 (period delay)，用以表示若上一個出發時刻和這一個出發時刻不在同一週期發生時，兩個時刻相隔多少週期，<sup>4</sup>其求法為

$$\mu_{ij} = \left\lceil \frac{a_{ij} + d_j(0) - d_i(0)}{T} \right\rceil$$

兩出發時刻在同一週期時則  $\mu_{ij} = 0$ 。其他例如列車交會、待避、轉乘接駁、車輛連結和分解、行車最小時隔等都可以仿照上面的方式把限制式寫下來。如果假設列車在滿足上述所有條件後立即開車，把所有的限制寫在一起就是

$$x_i(k) = \max_j (\max_j (a_{ij} + x_j(k - \mu_{ij}))) = \bigoplus_{j=1}^n (a_{ij} \otimes x_j(k - \mu_{ij})) \oplus d_i(k)$$

其中如果事件  $j$  不限制  $i$ ，則  $a_{ij} = \varepsilon$ 。把所有事件的發生時刻寫成一個向量  $x(k)$ ，我們可以把上面的式子用矩陣形式表達：

$$x(k) = \bigoplus_{l=0}^p (A_l x(k-l)) \oplus d(k) \quad (2.1)$$

$p$  是最大的週期落差，又稱為這個系統的階 (order)。定義回溯算子 (backward-shift operator)  $\gamma$  使得  $\gamma x(k) = x(k-1)$ ，並且  $\gamma^l x(k) = x(k-l)$ 。所以式 (2.1) 可以表示成

$$x(k) = \bigoplus_{l=0}^p (A_l x(k-l)) \oplus d(k) = \bigoplus_{l=0}^p A_l \gamma^l x(k) \oplus d(k) = A(\gamma)x(k) \oplus d(k) \quad (2.2)$$

其中  $A(\gamma) = \bigoplus_{l=0}^p A_l \gamma^l$  稱為  $\gamma$  的多項式矩陣 (polynomial matrix)。如果不考慮表定出發時刻的限制，則我們得到以下的齊次線性系統 (homogeneous linear system)：

$$x(k) = \bigoplus_{l=0}^p A_l x(k-l) = \bigoplus_{l=0}^p A_l \gamma^l x(k) = A(\gamma)x(k) \quad (2.3)$$

<sup>4</sup>若時刻表不是週期時刻表，當然就沒有週期落差可言。

### 2.3.2 Timed Event Graph

Timed event graph 是一種帶有標記 (mark) 的有向圖 (digraph), 可用於表示及分析動態系統。文中作者將列車時刻表系統以 time event graph 表示, 再以圖論演算法進行穩定性分析。

Timed event graph  $\mathcal{G} = (\mathcal{T}, \mathcal{P}, \mu, w)$  包含了四個部分,  $\mathcal{T}$  是 transition 的集合, 即圖的頂點。 $\mathcal{P}$  是 place 的集合, 是圖的邊。 $\mu$  是初始的標記, 即每個邊於初始狀態時標記的數目。最後的  $w$  是每個邊的保持時間 (holding time), 也就是邊的權重, 通常代表該過程所需的時間。它是以標記的變動來表示系統的動態行為, 變動規則是以下兩階段的 firing rule:

1. 若每個指向 transition  $i$  的邊都含有至少一個保持時間已到的標記, 則此 transition 是 enabled。
2. 一個 enabled 的 transition firing 時會移除一個指向它的邊的標記並加入一個標記到由它出發的邊。

前述的列車時刻表系統也可以看成是一種 timed event graph。從時刻表系統構造相對應之 timed event graph 的方式如下: 每個出發事件是一個 transition, 邊則由系統的多項式矩陣來決定, 若  $[A_l]_{ij} = w_{ij} \neq \varepsilon$ , 則 transition  $j$  到  $i$  間連上一個保持時間為  $w_{ij}$  且有  $l$  個標記的有向邊。這個相應的 timed event graph 記為  $\mathcal{G}(A(\gamma))$ 。

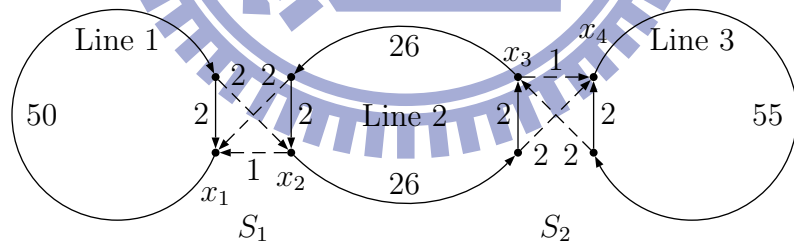


圖 2.1: 範例路網示意圖

**例 2.1.** 假設有一個如圖 2.1 的鐵路系統, 包含了三條路線、兩個車站和四個出發事件 ( $x_1$  到  $x_4$ )。三條路線的運行時間分別為 50、26、55 分鐘, 而停靠時間均為 2 分鐘, 在站內還要預留 2 分鐘的轉乘時間, 最後是最小出發時隔為 1 分鐘, 四個事件的表定時刻依序為 31、30、0、1, 即  $d(0) = [31, 30, 0, 1]^T$ 。

首先考慮  $x_1$  的限制。Line 1 給出了  $x_1(k) \geq x_1(k - \mu_{11}) \otimes 50 \otimes 2$ , 其中的  $\mu_{11} = [(52 + 31 - 31)/60] = 1$ 。類似地  $x_1$  和  $x_2$  的最小出發時隔則為  $x_1(k) \geq x_2(k) \otimes 1$ 。還有

$x_1(k) \geq x_3(k) \otimes 28$  和表定出發時刻的限制是  $x_1(k) \geq d_1(k)$ 。全部寫在一起並把不等號換成等號就是

$$x_1(k) = (52 \otimes x_1(k-1)) \oplus (1 \otimes x_2(k)) \oplus (28 \otimes x_3(k)) \oplus d_1(k)$$

以同樣的方式寫下其他出發事件之限制式為:

$$x_2(k) = (52 \otimes x_1(k-1)) \oplus (28 \otimes x_3(k)) \oplus d_2(k)$$

$$x_3(k) = (28 \otimes x_2(k-1)) \oplus (57 \otimes x_4(k-1)) \oplus d_3(k)$$

$$x_4(k) = (28 \otimes x_2(k-1)) \oplus (1 \otimes x_3(k)) \oplus (57 \otimes x_4(k-1)) \oplus d_4(k)$$

上面的式子表示成矩陣的型式如下:

$$x(k) = \begin{bmatrix} \varepsilon & 1 & 28 & \varepsilon \\ \varepsilon & \varepsilon & 28 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 1 & \varepsilon \end{bmatrix} x(k-1) \oplus \begin{bmatrix} 52 & \varepsilon & \varepsilon & \varepsilon \\ 52 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 28 & \varepsilon & 57 \\ \varepsilon & 28 & \varepsilon & 57 \end{bmatrix} d(k)$$

其多項式矩陣則為:

$$A(\gamma) = A_0 \oplus A_1 \gamma = \begin{bmatrix} 52\gamma & 1 & 28 & \varepsilon \\ 52\gamma & \varepsilon & 28 & \varepsilon \\ \varepsilon & 28\gamma & \varepsilon & 57\gamma \\ \varepsilon & 28\gamma & 1 & 57\gamma \end{bmatrix}$$

從上面的多項式矩陣可以得到這個系統的 timed event graph 如圖 2.2。 ◇

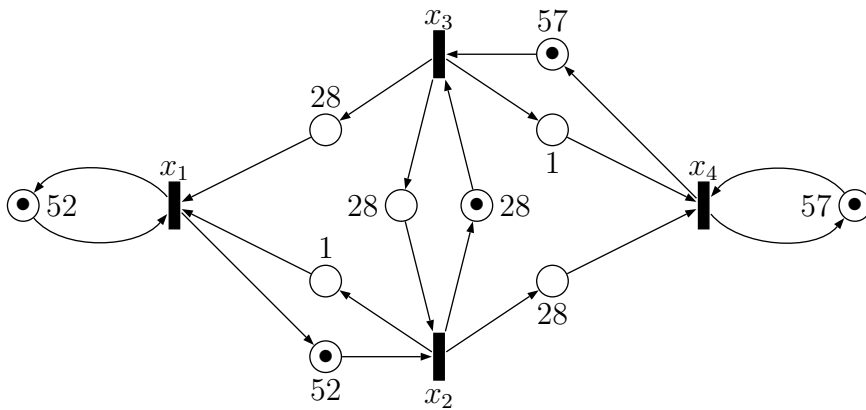


圖 2.2: Timed event graph



### 2.3.3 穩定性分析

文中作者對列車時刻表穩定的定義是：列車延誤可以在週期中藉由時刻表中的緩衝而減少，並能在有限時間內回到原來的表定時刻上運行。從上面的定義出發，從而有下述二個定理：

**Theorem 1.** <sup>5</sup>令  $A(\gamma) = \bigoplus_{l=0}^p A_l \gamma^l$  是不可約的多項式矩陣，<sup>6</sup>且  $G(A_0)$  不包含圈 (acyclic)。則  $A(\gamma)$  存在唯一的廣義特徵值 (generalized eigenvalue)  $\lambda > \varepsilon$  和有限個特徵向量  $v > \varepsilon$  使得  $A(\lambda^{-1}) \otimes v = v$ ，<sup>7</sup>而且  $\lambda$  等於  $\mathcal{G}(A(\gamma))$  的最大平均週期 (maximum cycle mean)

$$\eta = \max_{\xi \in C} \frac{w(\xi)}{\mu(\xi)}$$

其中的  $C$  是  $\mathcal{G}(A(\gamma))$  中所有基本圈<sup>8</sup>的集合， $w(\xi)$  是圈  $\xi$  的權重， $\mu(\xi)$  是圈  $\xi$  內標記總數。

**Theorem 2.** 令  $A(\gamma) = \bigoplus_{l=0}^p A_l \gamma^l$  是多項式矩陣且最大之廣義特徵值為  $\lambda_0$ 。則系統 (2.2) 是穩定的若且唯若  $\lambda_0 < T$ 。

第一個定理提示了  $G(A_0)$  的重要，用 timed event graph 的語言來說，就是圖中不可以有沒有標記的圈，否則有些事件就永遠不會發生。以列車時刻表的觀點言之就是某些列車無法發車，塞在路上了 (deadlock)。換句話說，如果時刻表不是週期時刻表時，從前面的敘述可知  $\mathcal{G}(A(\gamma)) = G(A_0)$  於是這個分析就失效了，因為這個 timed event graph 上沒有標記根本不會運作。

第二個定理則告訴我們， $A(\gamma)$  的最大廣義特徵值  $\lambda_0$  就是這個時刻表容許的最小週期時間，所以當  $\lambda_0$  小於時刻表週期  $T$  時表示此時刻表有  $T - \lambda_0$  的餘裕時間，反之則沒有餘裕時間所以是不穩定的。再從第一個定理可以知道，這個  $\lambda_0$  是  $\mathcal{G}(A(\gamma))$  中耗時最多的路徑長度，也就是完成一個週期的最小時間。這也告訴我們這個方法不適於直接用來分析像台鐵這樣的系統，就算我們用一天 (甚至是一個星期) 作為一個週期，要完成一個週期所需的最小時間也一定遠小於一天 (或一星期)，而且實務上也很少有列車延誤影響到下一天的運行。所以如果我們用這個方法來分析臺鐵的時刻表，一定會得到穩定的結論 (即  $\lambda_0 < T$ )，然而這個結論是沒有什麼用處。

作者還定義了三種穩定度的指標如下：

1.  $\rho = \lambda_0/T$
2.  $\Delta_1 = T - \lambda_0$  <sup>9</sup>
3.  $\Delta_2$  滿足  $(\Delta_2 \otimes A(T^{-1})) \otimes v = v$

<sup>5</sup>定理的標號是原始論文中的標號。

<sup>6</sup>也就是相應的 timed event graph 是強連通的 (strongly connected)!

<sup>7</sup>注意這裡的  $\lambda^{-1}$  其實就是  $-\lambda$ ，後面的  $T^{-1}$  亦同，因為在 max-plus 代數中的指數是通常代數中的乘法。

<sup>8</sup>基本圈 (elementary circuits) 是指在圈中每個頂點只經過一次的圈。

<sup>9</sup>原文中的  $\lambda_0$  誤為  $\lambda$ ，據[19]改正。

### 2.3.4 可實現的列車時刻表

列車時刻表並不一定都是可以實現的，有可能某一個出發事件兩次發生所需的最小時間比一個週期要大，於是這個時刻表就不可能準點。作者給出了一個可實現的充分必要條件：

**Theorem 4.** 一個週期時刻表在系統 (2.2) 中是可實現的(realizable) 若且唯若其初始時刻  $d_0$  滿足  $d_0 \geq A(T^{-1}) \otimes d_0$ 。

作者也證明了可實現的時刻表其最大特徵值  $\lambda_0$  至多就是  $T$ 。也就是說可實現的時刻表除了  $\lambda_0 = T$  的情況外是穩定的。

### 2.3.5 時刻表的穩健度(robustness) 和恢復矩陣 (recovery matrix)

在這篇文章中，作者認為穩健度決定於事件彼此影響的程度 (accessibility) 以及餘裕時間 (slack time) 的多寡。因此為了討論時刻表的穩健度，作者引入了恢復矩陣  $R = (r_{ij})$ ,  $r_{ij}$  是事件  $i$  到事件  $j$  的恢復時間，即所有路徑上餘裕時間之最小值，於是以下的定理：

**Theorem 6.** 若系統 (2.2) 是穩定的，則

$$r_{ij} = d_i^0 - d_j^0 - [A_T^+]_{ij}$$

其中  $A_T^+ = A^+(T^{-1}) = [\bigoplus_{l=0}^p A_l T^{-1}]^+$ 。

恢復矩陣的行、列和對角線元素還可以有以下的解釋：

1. 延誤衝擊向量 (delay impact vectors):  $R$  中的第  $j$  行是事件  $j$  到其他事件的恢復時間，所以可以表示事件  $j$  延誤造成對其他事件衝擊的程度。
2. 延誤敏感向量 (delay sensitivity vectors):  $R$  中的第  $i$  列是其他事件到事件  $i$  的恢復時間，所以是表示事件  $i$  對其他事件之延誤的敏感程度。
3. 循環恢復時間 (circulation recovery times):  $R$  中之對角線元素是每個事件循環一次的恢復時間。

其中的循環恢復時間可以視為是一種穩健度的指標，和之前所用的穩定度指標來比較，循環恢復時間是局部的指標，對每一個事件會有不同的值，而  $\lambda_0$ 、 $\rho$  等是時刻表整體的指標。

### 2.3.6 延誤傳遞

作者提出了一個計算列車延誤傳遞的方法。首先把一個時刻表系統寫成以下的型式：

$$\begin{cases} x(k) = A(\gamma)x(k) \oplus d(k) & k \in \mathbb{N} \\ d(k) = d_0 \otimes T^k & k \in \mathbb{N} \\ z(k) = D^{-1}(k) \otimes x(k) & k \in \mathbb{N} \\ x(1^-) = x_1, x(l) = x_l & 1-p \leq l \leq 0 \end{cases} \quad (2.4)$$

其中  $D^{-1}(k) = \text{diag}((d_1(k))^{-1}, \dots, (d_n(k))^{-1})$ ，或者用矩陣表示為

$$D^{-1}(k) = \begin{bmatrix} (d_1(k))^{-1} & & \mathcal{E} \\ & \ddots & \\ \mathcal{E} & & (d_n(k))^{-1} \end{bmatrix}$$

所以  $z_i(k) = (d_i(k))^{-1} \otimes x_i(k) = x_i(k) - d_i(k)$ ，即  $z(k)$  是延誤向量 (delay vector)，表示了系統的延誤狀況。

在此需要解釋的是第四行的初始條件。在這裡假設當下的時間為  $t_0 \in [0, T)$ ，而前  $p$  個週期的狀態  $x_0, x_{-1}, \dots, x_{1-p}$  是已知的 ( $p$  是系統的階數)。但是因為  $t_0 \in [0, T)$ ，所以在第一個週期中，我們只知道在  $t_0$  之前的狀態，於是  $x_1$  是有一部分是確定的，另一部分則是要計算的對象，所以在式子中的  $x(1^-)$  表示  $x_1$  在  $t_0$  之前的狀態。

接著只要解式 (2.4) 就可以知道延誤傳遞的情況。對矩陣  $A \in \mathbb{R}_{\max}^{n \times n}$  定義  $A^* = \bigoplus_{l=0}^{n-1} A^l$ 。作者得到以下的結果：

**Theorem 7.** 考慮可實現的時刻表系統 (2.4)，且  $G(A_0)$  不包含圈，則  $\{x(k)\}_{k \in \mathbb{N}}$  可以用下面的式子遞迴地決定：

$$\begin{aligned} x(1) &= A_0^* \otimes \left( x_1 \oplus \bigoplus_{l=1}^p A_l x_{1-l} \right) \oplus d(1) \\ x(k) &= \bigoplus_{l=1}^p A_0^* A_l x(k-l) \oplus d(k), \forall k \geq 2 \end{aligned}$$

定義 settling period  $k_s = \min\{k \in \mathbb{N} | x(k+l) = d(k+l), 1 \leq l \leq p\}$ ，即在經過  $k_s$  個週期後，系統回復到準點的狀態。而有關延誤的許多資訊都可以由此計算，例如個別列車或車站的延誤狀況等。

### 2.3.7 實例研究

作者用 2000/2001 年度荷蘭鐵路網和時刻表作為計算的實例。整個系統有 729 個站 (包含各種列車的衝突點，例如軌道交會點、可動式橋樑等)，分析的時段是上午尖峰。模化為 timed event graph 後，包含 3536 個 transition、25472 個 place 以及 12429 個 token，系統階數是 2。

因為每種列車在每個路段的最小運轉時間並不能從時刻表中知道，作者採用了二種方式來計算，第一種是以排定的時刻表時間作為最小運行時間；另外一種方式是把表定運行時間依經驗折減 7% 作為最小運行時間。在上述兩種方式下，分別算出了平均週期最大的四條列車路徑，也就是餘裕時間最小的列車路徑。

為了知道系統中影響穩定性的最關鍵因素，作者考慮不同的限制式組合，來分析每一種限制式對時刻表穩定性的影響。結論是取消等候旅客轉乘限制和使用額外的車輛（也就是放鬆車輛使用的限制）能有效地增進穩定性；而減少列車的整備時間和轉乘時間則沒有明顯的效果。

最後作者計算了一個延誤傳遞的例子。同樣的，作者也考慮放鬆其中的一些限制式來觀察其效果。最後作者表示這個方法可以應用到即時的列車延誤分析上。

### 2.3.8 Goverde 方法的長處與侷限

Goverde 方法最大的優點是其模化能力相當強，許多實務上的限制都可以被納入這個架構。像是列車與人員的運用、車輛的連結與分解、等候旅客轉乘等考慮都可以寫成 max-plus 代數的式子。因此應用時可以根據系統的特性、資料的詳略及應用目的等考量選擇所需的限制式。另外其演算法相當有效率，可以在短時間內計算龐大的路網，也能作為運轉整理時的輔助。

Goverde 的方法不適用於非週期時刻表，其理由已於 2.3.3 節中詳述，所以不能直接用於臺灣鐵路系統。另外一個值得討論的是 Goverde 的模化只考慮列車的出發時刻，因為他假定抵達和通過時刻就是列車的上一個出發時刻加上固定的站間運轉時間。但是實務上經常因為各種因素使得列車在站間運轉時發生延誤，進而影響到其他列車的站間運轉時間，所以這樣的簡化在分析延誤傳遞時會有問題。我們以圖 2.3 來說明，圖中粗實線表示列車表定的運轉曲線，虛線表示延誤發生時的運轉曲線。圖 2.3 顯示了在車站 2 發生列車追越的情形，如果先行的慢車發生延誤，如圖中的虛線所示，於是後方的快車被迫延後進站，其站間的運轉時間因延誤而增加，所以抵達和通過的時刻不再是上一個出發時刻加上站間運轉時間。因為不考慮抵達時刻，所以進站時隔的要求無法被反映出來，於是前方的慢車延誤不會影響快車通過，這是不合理的現象。即使把慢車運轉時的延誤視為上一站的出發延誤，問題仍然存在。如圖中所示，如果快車的後方還有一慢車，在這個情境下延誤不會影響到後方的慢車，但如果將運轉延誤視為上一站的出發延誤，那麼就會使後方的慢車也受到影響。從上面這個例子可以知道，還是應該把列車的抵達和通過時刻都考慮進來，才比較符合鐵路系統的實際狀況。

## 2.4 臺鐵之相關研究

近年有關國內的軌道研究以鐘志成、李治綱等學者的一系列研究報告最為詳盡[7, 8, 9, 10,

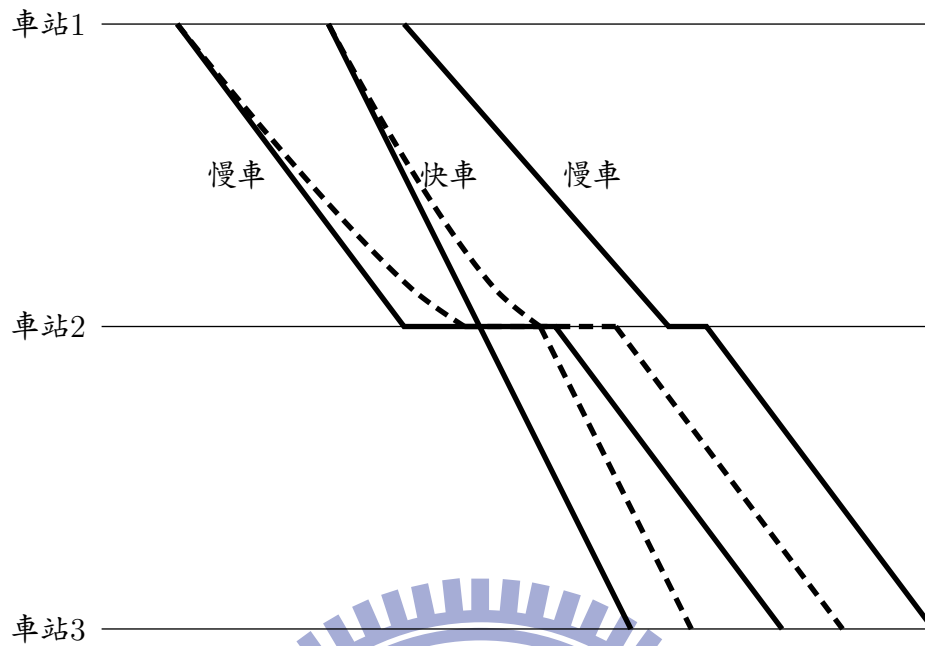


圖 2.3: 列車追越時產生的問題

11, 12]。除了[12]是排班的研究外, 其他都是容量分析。報告中詳細蒐集了臺鐵的路線、車輛、場站和運轉方式等資料並加以分析, 相當具有參考價值。

在[7]中, 作者建立了一個解析模式來分析臺鐵的軌道容量, 並實際計算臺鐵基隆到新竹路段的容量。其方法是把全部的路段先分成許多子路段, 對每個子路段計算最小運轉時隔, 計算不同的車種的運轉曲線及組成比例、站內的月台配置、站間運轉時間、號誌變換時間等, 然後求出每個路段單位時間內最多可以通過的列車數。最後取全路段的最小值就是全路段的容量。而[8]則把前述的研究成果轉換成以旅客數為單位的容量, 方法是調查每一種車種的乘載量分佈, 就可以估計出每列車的載客數, 進而換算成以旅客數為單位的容量。後續研究[9, 10, 11]則是前期研究的應用, 包括了模式改善與簡化、容量分析程式、容量分析手冊編製、列車服務可靠度分析等。另外[12]的主要成果是利用基因演算法建立一套臺鐵的列車排班模式。





## 第三章 鐵路系統的穩定性

### 3.1 鐵路系統的組成與運轉

鐵路系統包含了「路線」、「車輛」、「場站」、「號誌保安系統」等部分。

列車或車輛運轉所必需的通路稱為路線，包括軌道及道岔等。臺鐵依據軌道數的多寡及運轉方式，將路線區分為單線、複線、雙單線及多線等。單線是車站間只有一股軌道供列車雙向運轉。複線則是有兩股軌道分別供不同方向的列車行駛，而且每股軌道的列車運行方向是固定的，不能逆向行駛。雙單線和複線一樣有兩股軌道，不同的是兩股軌道都可以雙向運轉，如同兩單線一般，故稱為雙單線。路線上軌道數在三股以上的多線比較少見，通常是在非常繁忙的區間才有，例如日本 JR 東海道之東海道本線就是複複線，每方向有兩股軌道。臺鐵目前則已經沒有多線的區間。道岔通常設置於在路線的銜接點及車站內，是列車變換行駛軌道的設施，有了道岔列車才能進入正確的路線、停靠到正確的月台。

鐵路車輛行駛在路線上以運送旅客，在路線上的車輛是以編組的列車為運轉之單位。不同列車依鐵路單位的營運需求，會有不同的行駛速度和停站。例如臺鐵的列車中，就有自強、莒光、電車... 等不同等級的列車，其速度和停站都不同。但是如果路線上有許多不同速度的列車行駛會使路線容量減少，這是因為如果速度慢的列車先出發，下一列速度快的列車就要等候較長的時隔才可以出發，以確保不會追撞前車。因此在運量高的都市軌道系統中，同一路線上通常只使用一種速度的列車以增進運輸效率。而臺鐵的列車種類多，在排班時有九種速度別，也使臺鐵系統的排班和運行變得十分複雜[7]。

場站是列車停車進行上下客、交會待避其他列車的地點，而場站內月台和軌道路線數量是影響列車運行的關鍵因素之一。車站內的月台數會影響到列車進出站的時隔，也會影響到列車能否進行追越和待避作業。例如若月台或軌道數不足，在這個站快車就不能追越慢車，或是列車會和對向列車有衝突等。

號誌的功能是指示列車的行止，防止列車發生衝突，因此運轉中列車必須隨時遵守號誌的指示以保持列車間的距離。不同的號誌系統會使得列車間的容許間隔不同，影響列車運轉時隔與軌道容量甚鉅。早期的號誌多以人工手作的方式為之，依賴人員聯繫與確認，效率較差。現在則多以自動化的號誌設備取代之，雖然提高了效率與軌道容量，使得列車可以密集運轉，但是一旦號誌系統發生故障則容易對列車運轉造成嚴重影響。

運轉為列車於路線上行駛的過程，有時也稱為行車。為了保障列車在軌道上運轉之安全，所

採行間隔列車的方法稱為行車制度[4]。隨著技術的發展，行車制度也從早期依賴人員確認到現在全面自動化，除了較為安全外，也減少列車運轉時不必要的停等，增進運轉的效率。目前臺鐵大部分的區間都已經採行中央控制行車制度(CTC)，在此制度下列車行動完全依號誌指示，而所有號誌和道岔轉轍器均統一由行控中心控制，路線上所有列車都集中調度以提高效率。

大部分鐵路系統的行車制度都採行某種形式的閉塞制度，閉塞制度的基本概念是把路線分成一些區間，每區間只容許一列車進入以分隔列車，當列車進入每一個區間前，必須確認該區間內是否有其他列車佔用軌道，而辦理確認的手續稱為閉塞 (block)，管制其他列車進入的區間稱為閉塞區間。閉塞制度可分為固定區間閉塞制和移動區間閉塞制兩種，目前技術較成熟且廣被採用的是固定區間閉塞制，臺鐵也採用此種閉塞制度。這個方法是將路線分成固定的區間，每區間長度均要大於列車的煞車距離，每個時刻區間內只容許一列車進入，如果區間內有列車占用，區間外方的號誌會告知其他列車不得進入該區間，以達到列車分隔的目的。而移動式閉塞區間則是利用通訊和定位的技術，確認每列車的位置和速度，計算出適當的安全距離，再據以指示其他列車運行，如果兩列車距離過近就會指示後方列車減速。臺北捷運的中運量系統目前就採用移動式閉塞區間。應用移動式閉塞區間的系統並不多，而且列車間的交互影響也比固定式閉塞區間複雜，因此本研究只考慮固定式閉塞區間制。

### 3.2 列車時刻表的排班限制

列車時刻表是列車運轉的依據，通常包含了每列車於每一站的抵達和出發時間，列車必須按照時刻表上排定的時間運行。為了列車運轉的安全與順暢以及營運上的需求，在排班時會有許多的限制，這些限制又可以分成下述的幾類[12]：

1. 排班時間限制：基於營運的需求，會限制列車的出發或抵達時間要在某個範圍或時間點上。例如早上 8:00 左右在臺北站開出一列自強號的要求即屬於此類限制。列車排班時間限制可以非常嚴苛，也可以適度的放寬，例如可以要求該次列車必須於早上 8:00 準時從臺北站開出，或只要求 8:00 左右開出。但過於嚴苛的限制可能導致排班問題無解，因此列車排班時間限制通常是一個範圍，僅對於高等級列車才會有嚴格的限制。
2. 列車續進限制：即列車抵達下一站的時刻不小於其於前一站的離站時刻加上兩站間之最小運轉時間。不同的列車可以有不一樣的運轉時間，像是高級列車的運轉時間通常比低級列車短。另外也會視列車在前後兩站是否停車給予不同的站間運轉時間，這樣就有「停-停」、「停-通」、「通-停」和「通-通」四種組合。例如「停-通」就表示列車在出發站停車，而到達站通過的運轉時間，餘依此類推。這個限制也可以看成是對列車速度的限制。
3. 停站時間限制：列車在車站的停車時間必須足以讓所有旅客上、下車，因此有最短停車時



間的限制。若列車不停站，則最小停站時間為零。有時候列車可能會因交會或待避等原因而必須在車站停留較久的時間，但由於過長的停車時間容易導致旅客不耐，且從運轉的角度而言，亦顯得沒有效率，因此列車的停站時間一般會有最長時間限制。但是在折返或終點站可能會需要進行整備或變換方向及軌道行駛，所以會容許會較長的停站時間。

4. 運轉時隔限制: 為了使列車能夠安全地行駛於路線上，鐵路系統藉由閉塞制度來管制列車的運轉，以確保列車間保持安全的間距，因此會有運轉時隔的限制。這是列車排班及運轉時最重要的限制條件，在列車運轉過程中的任何時刻、任何地點均必須滿足。但是排班的實務上只要在列車容易發生衝突的地方加以限制就可以了，例如出發時隔、進站時隔、道岔時隔等。
5. 月台及軌道配置限制: 一段軌道及一個月台在一個時間點上只能有一列車占用，另外複線運轉的路線上列車不能使用對向軌道。
6. 其他限制條件: 例如車輛的運用、列車間旅客轉乘的銜接等。可以視營運上的要求增加限制條件，但是限制條件愈多，排班的問題也就愈複雜。

這些限制又可以分成「硬性」和「軟性」兩種。硬性限制主要源於安全上的要求，因此一定要滿足，例如運轉時隔、列車速度、月台及軌道配置限制等。軟性限制是因為營運上的需求，例如排班時間限制，這並沒有安全上的顧慮，所以通常比較寬鬆，必要時可以予以放寬。

### 3.3 延誤與餘裕

鐵路系統中的延誤可以分成兩種，原始延誤(primary delay) 和次生延誤 (secondary delay)。原始延誤是指列車在運轉或停等的過程發生的延誤，其原因可以是來自系統內部或外部。而次生延誤則是指因為延誤的列車由於運轉上的限制而造成其他列車也發生延誤，例如延誤的列車阻擋到其他列車，或是使有接續關係的列車也發生延誤 (如同一車輛的回程)。

原始延誤通常是源於人為操作不當、機械故障、旅客上下車的延誤、施工、重大事故等，欲減少原始延誤須由加強營運管理下手，不在本研究的探討範圍內。而藉由在時刻表中配置適當的餘裕，次生延誤在原始延誤不大的情況下是可以控制在合理範圍內。當然，如果發生重大的事故使路線中斷，再多餘裕都無濟於事，但是大部分的原始延誤都不大，因此在排班時加入適當的餘裕很重要。

一個與餘裕不同的概念是表定等候時間 (scheduled waiting time)，表定等候時間是指列車因為號誌、路線或車輛因素而等候的時間，例如在追越及單線區間交會時，列車要等候另一列車時增加的停等時間。從表面上看時刻表中不論是加入餘裕或是表定等候時間，都使列車運轉時間增加，但是餘裕有吸收延誤的功能，表定等候時間則否，其性質並不相同，有必要加以區別。

### 3.4 列車時刻表的穩定性

穩定性的概念源自於物理中關於平衡狀態的研究，舉一個簡單的例子，見圖 3.1，圖中的 A 和 B 兩質點分別靜止於山頂與山谷，均處於力平衡的狀態。但是質點 A 若受到一點水平的外力干擾，就會脫離平衡；反之質點 B 即使受到一點外力，仍可以保持在平衡狀態附近。所以質點 A 的平衡是不穩定平衡，而質點 B 是穩定平衡。粗略的說，一個系統為穩定的意思是：若系統的初始狀態離平衡狀態不遠，則系統的狀態軌跡 (state trajectory)<sup>1</sup>在任何時間都能保持在平衡狀態附近[2]。更進一步的說，當系統的狀態會收斂到平衡狀態時，則稱系統為漸近穩定的(asymptotic stable)。

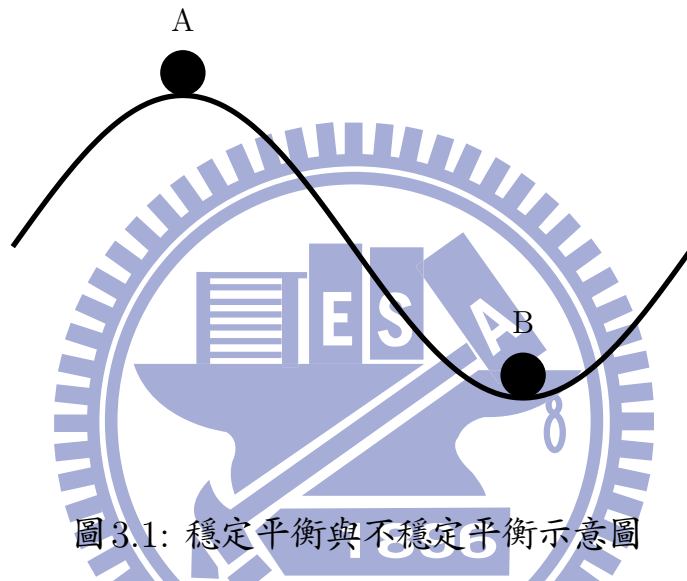


圖 3.1: 穩定平衡與不穩定平衡示意圖

將這個概念應用到鐵路系統上，列車的出發時刻可以視為系統的狀態，當給定了一個可運行的時刻表時，準點就是一個平衡狀態。而列車時刻表的穩定性就是當列車發生延誤時，系統是否可以自動地回復到準點，可以回復到準點為穩定，反之為不穩定。

一個時刻表穩定與否的關鍵因素在於餘裕時間的多寡，餘裕時間多則系統趨向穩定，反之則趨向不穩定。在 Goverde 的研究中[19, 20]，考慮的是週期時刻表，每個週期列車出發時刻是系統的狀態，藉由計算每個週期內的餘裕時間以判定時刻表系統是否穩定。這也是一種漸近穩定的概念，若週期內有餘裕時間，那麼有限的延誤就可以在有限個週期內恢復過來。

但是如果不是週期時刻表，漸近穩定的概念就無用武之地了，因為系統有起始和結束，當所有列車都抵達終點後，就沒有準點與否的問題。因此對於非週期列車時刻表，探討穩定性應從局部著手，對不同的列車、車站和時段分別檢視餘裕時間是否充足。具體地說就是對於一列車而言：

1. 於某處發生原始延誤時需要多久系統才能恢復準點？
2. 延誤會影響多少其他列車？即次生延誤有多少？

<sup>1</sup>簡單的說就是指系統狀態隨時間的變化情形。

### 3. 可以容許多少延誤而不影響其他列車？

穩定度是指穩定的程度，若原始延誤發生後能迅速回到準點狀態，次生延誤較少，以及容許的延誤量較多，則穩定度好，反之則穩定度差。如果列車本身即使沒有發生延誤仍會造成其他列車延誤，那麼就是不穩定的，這也表示時刻表一定不會準點，站在旅客的立場，不穩定的時刻表基本上不能被接受。

另外在文獻中也常見到列車服務的可靠度，簡單地說可以視為是列車的準點率或可用率。表面上看起來可靠度和穩定度很有關係，但是嚴格說來其實兩者之間不必然相關。即使是一個不甚穩定的時刻表，如果系統營運十分良好，原始延誤很少，那麼可靠度可以很高；反之如果時常生事故，即使時刻表的穩定度很好，可靠度還是相當差，所以兩者是不一樣的概念。對鐵路營運單位而言，提高時刻表穩定度和時刻表可靠度都很重要。





## 第四章 列車時刻表穩定性分析

### 4.1 以 Max-Plus 代數模化列車時刻表

鐵路系統能模化成以 max-plus 代數描述之離散事件系統(discrete event system), 系統由事件集合  $E$  與事件之發生時刻限制式所構成。系統中每個事件  $E_i = (T_i, L_i, S_i), i = 1 \cdots |E|$  是一個三元有序對,  $T_i$  是事件的類型, 可分為出發、抵達二種, 而通過事件則視為一個出發事件和一個抵達事件的組合, 其出發時刻等於抵達時刻。這是因為如果發生延誤時即使是通過的列車也可能被迫停留在站內, 所以還是把通過分為兩事件為宜, 而且在敘述時也比較簡明。 $L_i$  是事件對應的列車路線,  $S_i$  是事件發生的車站。對每一個事件  $E_i \in E$  定義  $x_i$  是其發生時刻,  $d_i$  是時刻表表定的抵達、出發或通過時刻。事件與事件的關係則可以用限制式表示。在 max-plus 代數的架構下, 鐵路運轉的限制可以表示成線性方程組。

不同的鐵路系統有不同的運轉限制, 即使在同一個鐵路系統中, 運轉的限制依不同路段、運轉的條件和方式也會不同, 例如在臺鐵系統中有些路段是單線運轉, 有些是雙線運轉, 列車間交互影響的方式也就不一樣。以下詳細說明各種限制如何以 max-plus 代數表示, 實際應用時應依具體情況加以取捨, 甚至略為修正以符合系統特性。

假設  $E_i, E_j, E_k \in E$  是系統中的事件:

1. 列車續進限制: 若  $E_i, E_j$  是同一列車在相鄰兩站的出發和抵達事件, 假設該列車的站間運轉時間是  $a_{ji}$ , 則  $E_i$  與  $E_j$  滿足:  $x_j \geq a_{ji} \otimes x_i = a_{ji} + x_i$ 。也就是說  $E_j$  的發生時刻不早於  $E_i$  的發生時刻加上站間運轉時間。
2. 列車停站限制: 若  $E_i, E_j$  是同一列車在同一站的抵達和出發事件, 假設該列車於該站的停站時間是  $a_{ji}$ , 則  $E_i$  與  $E_j$  滿足:  $x_j \geq a_{ji} \otimes x_i$ 。
3. 出發時隔限制: 若  $E_i, E_j$  是同一站的出發事件, 且  $E_j$  是  $E_i$  的下一個出發事件, 假設該站的最小出發時隔是  $a_{ji}$ , 則  $E_i$  與  $E_j$  滿足:  $x_j \geq a_{ji} \otimes x_i$ 。
4. 進站時隔限制: 若  $E_i, E_j$  是同一站的抵達事件, 且  $E_j$  是  $E_i$  的下一個抵達事件, 假設該站的最小進站時隔是  $a_{ji}$ , 則  $E_i$  與  $E_j$  滿足:  $x_j \geq a_{ji} \otimes x_i$ 。
5. 站內軌道及月台數限制: 如果站內的軌道都有列車使用時, 則下一列車就必須等候有列車出發後才可以進站。考慮列車抵達和通過二種情況, 列車抵達時必須有臨月台的軌道可用方能進站, 而通過的列車只要有軌道可用就可以進站, 因此早抵達的列車的出站時刻會限制後來列車的到站時刻。舉例來說, 如果站內只有一軌道, 則列車的進站時刻就不能早於



前一列車的出發時刻加上同股道到開時隔；若站內有二軌道，則列車的進站時刻就不能早於前方第二列車的出發時刻加上同股道到開時隔。如果前方列車有追越的情況，那麼該列車出發後站內還有列車，可以看成是軌道數減少。上述限制能以下面的方法表達：設站內有  $u$  個臨月台軌道和  $v$  個不臨月台軌道，該站的同股道到開時隔為  $a$ 。若  $E_i$  是出發事件， $E_{i'}$  是  $E_i$  對應的列車在  $S_i$  的抵達事件，並且在  $S_i$  追越了  $c$  列列車，<sup>1</sup>則

(a) 令  $E_j$  為  $E_{i'}$  發生後該站的第  $u - c$  個需停靠月台的抵達事件，則  $x_j \geq a \otimes x_i$ 。

(b) 令  $E_k$  為  $E_{i'}$  發生後該站的第  $u + v - c$  個抵達事件，則  $x_k \geq a \otimes x_i$ 。

因為時刻表是給定的，所以每一站列車的抵達和出發順序也是確定的，若兩列車預定同時抵達，則任選一個次序即可，不過選擇較早離開之列車先抵達會比較好，如此可以避免追越的現象，於是  $E_j, E_k$  可以唯一確定。不論是單線或複線運轉的情況，月台和軌道可能只供其中一個方向使用，也可能是兩個方向共用。如果沒有兩方向共用的軌道，只要把兩個方向分開處理即可。若存在共用軌道，則先將兩方向的限制式都先分別寫下來，兩個方向都要把共用的軌道算進軌道數，然後不考慮方向的問題，把全部列車和軌道都算進來，再寫一次對應的限制式。因為列車既要滿足兩方向各自的限制，也要滿足車站內軌道總數的限制，所以兩組限制式合起來就描述了有共用軌道的情况。不過這樣並沒有考慮到站內月台及軌道配置方式的差異，一個解決的方式是對不同的配置方式給予不同的到開時隔以反映其影響。

6. 表定時刻限制：列車的出發時刻不可以早於時刻表上預定的時刻，所以有  $x_i \geq d_i$ 。在沒有其他限制條件下，通常鐵路運轉時不會要求列車不可以提早抵達，因此抵達時刻可以沒有這項限制。
7. 單線運轉限制：單線運轉時兩站間的區間一次只能有一列車進入，下一列車必須等前列車離開此區間後方能進入。對某個區間而言，從給定的時刻表可以知道所有進入此區間之列車和進入的順序。假設  $E_i$  是某列車離開區間的事件，實際上就是其中一端車站的抵達事件，而  $E_j$  是跟隨  $E_i$  的下一個進入區間的事件，也就是其中一端車站的出發事件。如果  $E_i$  和  $E_j$  是同向列車，那麼令  $a_{ji}$  是號誌作動所需的時間，若  $E_i$  和  $E_j$  是對向列車，則令  $a_{ji}$  是該站的反向到開時隔，於是把限制式寫下來就是  $x_j \geq a_{ji} \otimes x_i$ 。
8. 禁止列車於同一軌道追越：列車不得在同一軌道上追越其他列車，這能模化為列車順序的限制。如果站間只有一軌道可供同一方向列車使用（例如單線或複線運轉），則先離站的列車就要先抵達，於是若在同一車站中  $E_j$  是  $E_i$  的下一個出發事件，且兩出發事件是同一方向的列車， $E_{j'}$  和  $E_{i'}$  為對應的下一站之抵達事件，就必須滿足  $x_{j'} \geq x_{i'}$ 。若站間有二軌道以上可供同方向列車使用時（例如雙單線運轉或複複線），列車可以用另一軌道追越

<sup>1</sup>這裡「追越」指的是列車比較晚抵達卻比較早出發的情況，並不限定要同向的列車。

其他列車,就沒有這個限制。

9. 車輛接續限制: 當不同列車使用同一車輛運行時, 這些列車間就有車輛接續限制, 後續的列車必須等候前列車抵達後並整備完畢方可出發。但是如果有充足的備用車輛時, 則不會有這個限制。
10. 其他限制: 其他的限制式, 例如列車等候旅客轉乘等限制, 也可以仿照前面的作法寫下來, 只要確定是哪個事件要等候另一個事件, 就可以把限制式寫下來。另外和排班時不同的是沒有最長停站時間限制, 延誤發生時一般不可能要求列車滿足最長停站時間限制。依鐵路系統的不同, 前述的模化也可能應修正以適合實際情況。

從上面的討論可以發現, 許多不同的運轉限制在 max-plus 代數的架構下都能表達成相同的形式, 都可以寫成事件發生時刻的線性不等式。因為事件必須在滿足所有限制條件後才可以發生, 所以把所有的限制式加起來 (這裡的「加」是指  $\oplus$ , 取最大值的意), 於是  $\forall E_i \in E$  有

$$x_i \geq \left( \bigoplus_{E_j \in E} a_{ij} \otimes x_j \right) \oplus d_i \quad (4.1)$$

其中  $a_{ij}$  的值由  $E_j$  和  $E_i$  的限制關係決定, 若  $E_j$  不限制  $E_i$  則  $a_{ij} = \varepsilon = -\infty$ 。另外有  $a_{ii} = \varepsilon$ 。於是鐵路系統就可以上述的方式表示成一個 max-plus 代數下的不等式組。

假設事件在滿足所有的條件後立刻發生, 或者說列車在滿足條件後立刻出發或抵達, 並且假定沒有運轉整理, 所有列車均按時刻表定的順序運轉, 那麼式 (4.1) 中的不等號就可以改成等號, 就變成線性方程組。與平常的線性代數相仿, 我們可以將線性方程組以矩陣的方式表達, 寫成矩陣的形式就是

$$\mathbf{x} = A \otimes \mathbf{x} \oplus \mathbf{d} \quad (4.2)$$

其中  $A = (a_{ij})$ ,  $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$  是由事件發生時刻構成的行向量,  $\mathbf{d} = [d_1, d_2, \dots, d_n]^T$  是事件的表定發生時刻構成的行向量。這個方程描述了整個鐵路系統的行為, 我們將以此為基礎分析鐵路系統的穩定性。

下面用一個例子展示如何將一個時刻表寫成 max-plus 代數下的方程組。

**例 4.1** (模化列車時刻表的例子)。表 4.1 是一個虛擬的時刻表, 包含了四個車站和六列車。其中車站 C 到車站 D 之間為單線區間, 餘為雙線區間。每個時刻右方的數字是該事件的編號, 1 就對應於  $E_1$ , 其發生時刻就是  $x_1$ 。注意列車 T3 在車站 B 和 C 其實是通過不停站的。車站 A 和 D 各有兩個上行月台和下行月台, 車站 B 則只有一個上行月台和下行月台, 車站 C 比車站 B 要多一個上下行共用且不臨月台的軌道。假設每站的進站時隔、出發時隔、反向到開時隔均為 2 分, 同股道到開時隔為 1 分, 單線區間列車進站後的號誌作動時間為 1 分。另外列車 T2 和 T7 以及 T5 和 T8 有接續關係, 其最小整備時間為 20 分。站間最小運轉時間比表定時間少 1 分鐘。

表 4.1: 虛擬時刻表

車站\列車	T1 下	T2 上	T3 下	T5 下	T7 下	T8 上
A	7:00 1		7:20 15			
	7:05 2	8:05 9	7:23 16	8:00 23	8:40 29	9:55 35
B	7:15 3	7:55 10	7:30 17	8:10 24	8:50 30	9:45 36
	7:20 4	7:50 11	7:30 18	8:12 25	8:52 31	9:42 37
C	7:30 5	7:40 12	7:37 19	8:22 26	9:02 32	9:32 38
	7:40 6	7:35 13	7:37 20	8:25 27	9:05 33	9:30 39
D	7:50 7	7:25 14	7:44 21	8:35 28	9:15 32	9:20 40
	7:52 8		7:50 22			

接下來考慮系統中的限制式。首先是列車續行和停站限制，以列車 T1 為例，從車站 A 開始，有以下的限制

$$\begin{aligned}
 x_2 &\geq 5 \otimes x_1 & x_3 &\geq 9 \otimes x_2 \\
 x_4 &\geq 5 \otimes x_3 & x_5 &\geq 9 \otimes x_4 \\
 x_6 &\geq 5 \otimes x_5 & x_7 &\geq 9 \otimes x_6 \\
 x_8 &\geq 5 \otimes x_7
 \end{aligned}$$

其他列車都可以仿照上述的方式將限制式寫出來。比較特別的是 T3 在車站 B 及車站 C 不停，所以在這兩站的停站限制就是

$$x_{17} \geq 0 \otimes x_{18} \quad x_{20} \geq 0 \otimes x_{19}$$

列車抵達和出發時隔以 A 站為例如下，左邊是到站時隔，右邊是離站時隔。

$$\begin{aligned}
 x_{15} &\geq 2 \otimes x_1 & x_{16} &\geq 2 \otimes x_2 \\
 x_{35} &\geq 2 \otimes x_9 & x_{23} &\geq 2 \otimes x_{16} \\
 & & x_{29} &\geq 2 \otimes x_{23}
 \end{aligned}$$

因為車站 C 到車站 D 之間是單線區間，因此有單線運轉的限制如下：

$$\begin{aligned}
 x_{20} &\geq 2 \otimes x_{13} & x_6 &\geq 1 \otimes x_{21} \\
 x_{27} &\geq 1 \otimes x_7 & x_{33} &\geq 1 \otimes x_{28} \\
 x_{40} &\geq 2 \otimes x_{34}
 \end{aligned}$$

列車 T2 和 T7 以及 T5 和 T8 有接續關係，其最小整備時間為 20 分，所以列車接續限制是

$$x_{29} \geq 20 \otimes x_9 \quad x_{40} \geq 20 \otimes x_{28}$$



接著是車站內月台及軌道數限制，車站 A、B 和 D 都沒有兩方向共用軌道，比較單純。但車站 C 有一個不臨月台的軌道，要分兩階段來處理。以車站 C 為例，首先是下行方向的限制式

$$x_{26} \geq 1 \otimes x_6 \quad x_{32} \geq 1 \otimes x_{27}$$

上行方向則是  $x_{39} \geq 1 \otimes x_{12}$ 。把全部列車一起考慮時，列車出發的次序是 T3、T1、T2、T5、T7、T8，不過只有 T3 可以使用不臨月台的軌道，因此只需考慮 T3 的限制式即可，就是

$$x_{26} \geq 1 \otimes x_{20}$$

至於不能在同一軌道上追越的限制，仔細寫下來後會發現限制式中的發生時刻變數與列車進站時隔限制式相同，這表示時刻表中列車順序沒有在站間運轉時改變，也就是沒有列車於站間追越其他列車，於是兩種限制式相加後 ( $\oplus$ ) 會等於列車進站時隔之限制式，故可忽略不計。最後還有表定時間的限制，把所有的限制式寫在一起，就得到  $\mathbf{x} = \mathbf{A} \otimes \mathbf{x} \oplus \mathbf{d}$  型式的方程組，而30頁的圖4.1是 A 的 precedence graph。◇

## 4.2 $\mathbb{R}_{\max}^n$ 中的序關係

在開始分析列車時刻表之前，必須先討論一個有用的概念，就是  $\mathbb{R}_{\max}^n$  中的序關係 (order relation)，或者通俗地說就是大小關係。在 max-plus 代數中，因為  $\mathbb{R}_{\max} = \mathbb{R} \cup \{-\infty\}$ ，所以其序關係可以直接繼承自實數系的序關係，雖然有一些性質會不一樣，但不是大問題。然而後文中會顯示出我們也需要在  $\mathbb{R}_{\max}^n$  上，也就是  $\mathbb{R}_{\max}$  向量之集合裡定義序關係。<sup>2</sup>

首先定義何謂一般集合上的序關係。

**定義 4.1** (序關係). 集合  $\mathcal{S}$  上的序關係是  $\mathcal{S} \times \mathcal{S}$  的子集合  $\mathcal{R} \subset \mathcal{S} \times \mathcal{S}$ ，並滿足以下條件：對任意的  $a, b, c \in \mathcal{S}$

1.  $(a, a) \in \mathcal{R}$  (反身性)
2. 若  $(a, b) \in \mathcal{R}$  且  $(b, c) \in \mathcal{R}$  則  $(a, c) \in \mathcal{R}$  (遞移性)
3. 若  $(a, b) \in \mathcal{R}$  且  $a \neq b$  則  $(b, a) \notin \mathcal{R}$  (反對稱性)

若  $(a, b) \in \mathcal{R}$  則定義  $a \geq b$ ，反之  $(a, b) \notin \mathcal{R}$  則  $a \not\geq b$ 。||

如果集合中的元素可以運算，那麼這個運算經常可以用來定義集合上的序關係，例如實數系上的加法就定義了通常的  $\geq$ ，這種序關係稱為典型序關係 (canonical order relation)。對於

<sup>2</sup>雖然這裡用「向量」一詞，但是  $\mathbb{R}_{\max}^n$  並不是向量空間，只是半模 (semi-module)，因為其加法反元素不存在。關於半模的定義和性質請參閱[18], chap. 5。

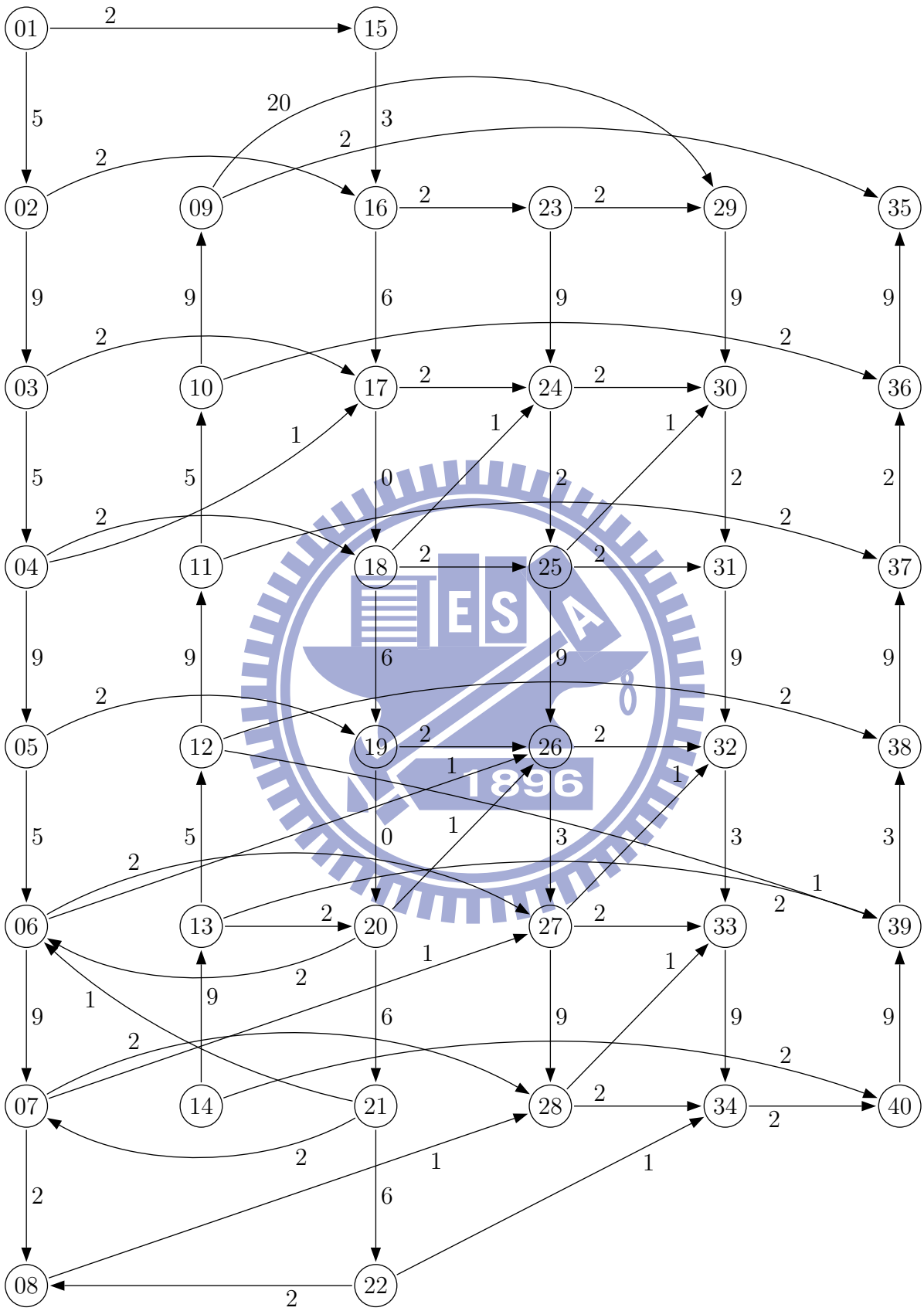


圖 4.1: 虛擬時刻表之 precedence graph

$\mathbb{R}_{\max}^n$  而言, 把  $\mathbb{R}_{\max}^n$  的元素視為行向量, 則矩陣的加法 ( $\oplus$ ) 就是  $\mathbb{R}_{\max}^n$  中的運算, 於是我們也可以作如下的定義。

**定義 4.2** ( $\mathbb{R}_{\max}^n$  中的典型序關係). 對任意的  $\mathbf{a}, \mathbf{b} \in \mathbb{R}_{\max}^n$ , 定義  $\mathbb{R}_{\max}^n$  中的典型序關係  $\geq$  使得  $\mathbf{a} \geq \mathbf{b} \Leftrightarrow \exists \mathbf{c} \in \mathbb{R}_{\max}^n : \mathbf{a} = \mathbf{b} \oplus \mathbf{c}$ . ||

這個定義和通常實數系上的型式是一致的, 以下若沒有特別指明,  $\mathbb{R}_{\max}^n$  中的序關係就是典型序關係。不過這個定義比較難操作。下面給出一個容易操作的等價定義, 並且用它證明定義 4.2 確實定義了一個  $\mathbb{R}_{\max}^n$  中序關係 (也就是滿足定義 4.1), 而且  $\oplus$  運算會保持這個序關係。

**定理 4.1.** 定義 4.2 等價於  $\forall \mathbf{a}, \mathbf{b} \in \mathbb{R}_{\max}^n : \mathbf{a} \geq \mathbf{b} \Leftrightarrow \mathbf{a} = \mathbf{a} \oplus \mathbf{b}$ , 是  $\mathbb{R}_{\max}^n$  上的序關係。並且滿足若  $\mathbf{a} \geq \mathbf{b}$  則  $\Rightarrow \forall \mathbf{c} \in \mathbb{R}_{\max}^n : \mathbf{a} \oplus \mathbf{c} \geq \mathbf{b} \oplus \mathbf{c}$ 。

**證明.** 首先證明兩個定義等價。若  $\mathbf{a} = \mathbf{a} \oplus \mathbf{b}$ , 只要令  $\mathbf{c} = \mathbf{a}$  就得到  $\mathbf{a} = \mathbf{b} \oplus \mathbf{c}$ 。反之若存在  $\mathbf{c}$  使得  $\mathbf{a} = \mathbf{b} \oplus \mathbf{c}$ , 在等式兩邊加上  $\mathbf{b}$  就得到

$$\mathbf{a} \oplus \mathbf{b} = (\mathbf{b} \oplus \mathbf{c}) \oplus \mathbf{b} = \mathbf{b} \oplus (\mathbf{b} \oplus \mathbf{c}) = (\mathbf{b} \oplus \mathbf{b}) \oplus \mathbf{c} = \mathbf{b} \oplus \mathbf{c} = \mathbf{a}$$

於是知道兩個定義是等價的。

利用這個等價的定義接下來證明定義 4.2 的  $\geq$  是一個序關係。因為  $\mathbf{a} = \mathbf{a} \oplus \mathbf{a}$ , 就是  $\mathbf{a} \geq \mathbf{a}$ , 所以有反身性。如果  $\mathbf{a} \geq \mathbf{b}$ , 又有  $\mathbf{b} \geq \mathbf{a}$ , 因為  $\mathbf{a} \oplus \mathbf{b} = \mathbf{b} \oplus \mathbf{a}$ , 所以  $\mathbf{a} = \mathbf{a} \oplus \mathbf{b} = \mathbf{b} \oplus \mathbf{a} = \mathbf{b}$ , 故反對稱性成立。若  $\mathbf{a} \geq \mathbf{b}$  且  $\mathbf{b} \geq \mathbf{c}$ , 則  $\mathbf{a} = \mathbf{a} \oplus \mathbf{b}$  且  $\mathbf{b} = \mathbf{b} \oplus \mathbf{c}$ , 所以

$$\mathbf{a} = \mathbf{a} \oplus \mathbf{b} = (\mathbf{a} \oplus \mathbf{b}) \oplus (\mathbf{b} \oplus \mathbf{c}) = (\mathbf{a} \oplus \mathbf{b} \oplus \mathbf{b}) \oplus \mathbf{c} = (\mathbf{a} \oplus \mathbf{b}) \oplus \mathbf{c} = \mathbf{a} \oplus \mathbf{c}$$

就是  $\mathbf{a} \geq \mathbf{c}$ , 遞移性也成立, 於是證明了  $\geq$  是一個序關係。

最後若  $\mathbf{a} \geq \mathbf{b}$ , 則  $\forall \mathbf{c} \in \mathbb{R}_{\max}^n$  有  $\mathbf{a} \oplus \mathbf{c} = (\mathbf{a} \oplus \mathbf{b}) \oplus (\mathbf{c} \oplus \mathbf{c}) = (\mathbf{a} \oplus \mathbf{c}) \oplus (\mathbf{b} \oplus \mathbf{c})$ , 所以  $\mathbf{a} \oplus \mathbf{c} \geq \mathbf{b} \oplus \mathbf{c}$ . □

**推論 4.1.**  $\mathbf{a}, \mathbf{b} \in \mathbb{R}_{\max}^n$ ,  $\mathbf{a} \geq \mathbf{b}$  若且唯若  $\mathbf{a}$  的每個元素都大於或等於  $\mathbf{b}$  相對應的元素。

**證明.** 直接從  $\mathbf{a} = \mathbf{a} \oplus \mathbf{b}$  就可以知道推論成立。 □

有一點值得注意, 在  $\mathbb{R}_{\max}^n$  中並不是任意兩元素都可以比較大小。考慮  $\mathbb{R}_{\max}^2$  的一個例子, 令  $\mathbf{a} = [1, 2]^T$ ,  $\mathbf{b} = [2, 1]^T$ , 則  $\mathbf{a} \oplus \mathbf{b} = [2, 2]^T$  既不等於  $\mathbf{a}$  也不等於  $\mathbf{b}$ , 所以  $\mathbf{a} \not\geq \mathbf{b}$  且  $\mathbf{b} \not\geq \mathbf{a}$ 。<sup>3</sup> 即使如此, 在  $\mathbb{R}_{\max}^n$  中還是可以有極大和極小的概念, 以下這個定義將會扮演重要的角色。

<sup>3</sup>這種序關係稱為偏序(partial order), 這是相對於全序 (total order) 而言的。簡單的說, 全序就是任兩元素都可以比較大小。

**定義 4.3** (極小元素與極大元素).  $\mathcal{S} \subset \mathbb{R}_{\max}^n$ , 若  $\mathbf{a} \in \mathcal{S}$  滿足  $\forall \mathbf{b} \in \mathcal{S} : \mathbf{b} \geq \mathbf{a}$ , 則稱  $\mathbf{a}$  是  $\mathcal{S}$  中的極小元素(minimal element)。反之若  $\forall \mathbf{b} \in \mathcal{S} : \mathbf{a} \geq \mathbf{b}$ , 則稱  $\mathbf{a}$  是  $\mathcal{S}$  中的極大元素(maximal element)。 ||

### 4.3 求解列車時刻表系統

分析的第一步是探討若鐵路系統中發生延誤, 會如何影響其他列車。首先我們假設所有列車在發生延誤後都依原先的次序運轉, 沒有運轉整理, 只有列車的抵達或出發時刻被迫延後, 且最小運轉時間及停站時間均不變。若  $E_i$  發生  $\delta_i < \infty$  的延誤, 則  $x_i$  就必須滿足

$$x_i \geq \delta_i \otimes d_i$$

令  $\Delta = [\delta_1 \otimes d_1, \dots, \delta_n \otimes d_n]^T$  (若  $E_i$  沒有延誤則  $\delta_i = 0$ ), 於是系統(4.2)可以改寫成

$$\mathbf{x} = A \otimes \mathbf{x} \oplus \Delta \tag{4.3}$$

只要解出方程組(4.3)和原本的表定發生時刻比較, 就知道每個事件受到多少影響。但是這個方程組的解並不一定唯一, 下面給出一個例子。

**例 4.2.** 考慮一個有三個事件的系統如下:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon \\ 2 & \varepsilon & 0 \\ 2 & 0 & \varepsilon \end{bmatrix} \otimes \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

簡單的計算就可以知道  $[1, 3, 3]^T$  和  $[1, 4, 4]^T$  都是這個方程的解, 而且事實上這個方程會有無窮多解。 ◇

雖然方程的解不唯一, 不過因為假設所有事件在滿足所有的條件後立刻發生, 所以事實上要找的是(4.3)的極小解 (minimal solution), 也就是所有解的極小元素。不過即使沒有延誤, 時刻表也不一定就會準點, 所以我們有以下的定義。

**定義 4.4** (可實現的時刻表). 如果  $\mathbf{d}$  是方程(4.2)的解, 則稱此時刻表為可實現的時刻表 (realizable timetable)。 ||

換句話說, 不可實現的時刻表就不可能準點。另外一個不同的概念是可運行的時刻表:

**定義 4.5** (可運行的時刻表). 如果方程(4.2)有解, 則稱此時刻表為可運行的時刻表 (operational timetable)。 ||

可運行與可實現不同，可運行只是列車能依表定的順序運轉，但是可實現則要求要能準點。

本節將說明求解方程組 (4.2) 和 (4.3) 的理論基礎，我們從一個重要的定義和定理開始。

**定義 4.6** (擬反矩陣).  $A \in \mathcal{M}_n(\mathbb{R}_{\max})$ , 若  $\lim_{k \rightarrow \infty} A^{(k)}$  收斂,<sup>4</sup> 則定義  $A$  的擬反矩陣 (quasi-inverse)  $A^* = \lim_{k \rightarrow \infty} A^{(k)}$ . ||

**定理 4.2.**  $A \in \mathcal{M}_n(\mathbb{R}_{\max})$ , 若  $A^*$  存在, 則  $\forall \mathbf{b} \in \mathbb{R}_{\max}^n$ ,  $A^* \otimes \mathbf{b}$  是方程  $\mathbf{x} = A \otimes \mathbf{x} \otimes \mathbf{b}$  的解。

**證明.** 從  $A^{(k)}$  的定義可以知道

$$\begin{aligned} A \otimes (A^{(k)} \otimes \mathbf{b}) \oplus \mathbf{b} &= A \otimes \left( \bigoplus_{i=0}^k A^i \otimes \mathbf{b} \right) \oplus \mathbf{b} = \left( \bigoplus_{i=1}^{k+1} A^i \otimes \mathbf{b} \right) \oplus (I_n \otimes \mathbf{b}) \\ &= \left( \bigoplus_{i=1}^{k+1} A^i \otimes \mathbf{b} \right) = A^{(k+1)} \otimes \mathbf{b} \end{aligned}$$

因為  $A^* = \lim_{k \rightarrow \infty} A^{(k)}$  存在, 如果極限運算對  $\oplus$  和  $\otimes$  都滿足分配律, 則只要對上式兩端取極限  $k \rightarrow \infty$ , 就得到  $A \otimes (A^* \otimes \mathbf{b}) \oplus \mathbf{b} = A^* \otimes \mathbf{b}$ 。要證明極限運算對  $\oplus$  和  $\otimes$  都滿足分配律, 只要證明對在  $\mathbb{R}_{\max}$  中的任意收斂數列  $\{a_k\}$  和  $\{b_k\}$  有

$$\lim_{k \rightarrow \infty} (a_k \oplus b_k) = \lim_{k \rightarrow \infty} a_k \oplus \lim_{k \rightarrow \infty} b_k \quad (4.4)$$

$$\lim_{k \rightarrow \infty} (a_k \otimes b_k) = \lim_{k \rightarrow \infty} a_k \otimes \lim_{k \rightarrow \infty} b_k \quad (4.5)$$

即可。因為  $\otimes$  是通常的加法  $+$ , 所以式 (4.4) 自然成立, 接下來只需要證明  $\oplus$  的情況也成立。設  $\lim_{k \rightarrow \infty} a_k = \alpha$ ,  $\lim_{k \rightarrow \infty} b_k = \beta$ , 不失一般性地可以假設  $\alpha \geq \beta$ , 於是等式右邊就是  $\alpha \oplus \beta = \alpha$ 。這告訴我們  $\alpha \geq \beta$ , 以下分  $\alpha = \beta$  和  $\alpha > \beta$  兩種情況:

1. 若  $\alpha = \beta$ , 由  $\{a_k\}$  和  $\{b_k\}$  的收斂可以知道存在  $K \in \mathbb{N}$  使得對任意的  $\epsilon > 0$ , 若  $k > K$  則  $|a_k - \alpha| < \epsilon$  且  $|b_k - \beta| = |b_k - \alpha| < \epsilon$ 。由  $\lim_{k \rightarrow \infty} (a_k \oplus b_k) = \lim_{k \rightarrow \infty} \max(a_k, b_k)$ , 所以當  $k > K$  時, 不論是  $a_k > b_k$  還是  $a_k \leq b_k$  都有  $|\max(a_k, b_k) - \alpha \oplus \beta| = |\max(a_k, b_k) - \alpha| < \epsilon$ 。
2. 若  $\alpha > \beta$ , 則存在  $K \in \mathbb{N}$  使得對任意滿足  $(\alpha - \beta)/2 > \epsilon > 0$  的  $\epsilon$ , 若  $k > K$  則  $|a_k - \alpha| < \epsilon$  且  $|b_k - \beta| < \epsilon$ 。於是當  $k > K$  時有  $a_k - b_k = a_k - \alpha + \alpha - b_k + \beta - \beta > (a_k - \alpha) + (\beta - b_k) + 2\epsilon > 2\epsilon - |a_k - \alpha| - |\beta - b_k| > 0$  就是  $a_k > b_k$ , 所以  $|\max(a_k, b_k) - \alpha \oplus \beta| = |a_k - \alpha| < \epsilon$ 。

這就證明了式 (4.5), 從而證明了此定理。 □

從定理 4.2 可以知道解方程 (4.2) 的關鍵是  $A^*$  存在, 以下給出一個  $A^*$  存在的充分條件, 也證明定理 4.2 中給出的解在此條件下是極小解。

<sup>4</sup>這裡的收斂是指矩陣的每個元分別收斂, 當然若極限是  $-\infty$  仍然是收斂的, 因為  $-\infty \in \mathbb{R}_{\max}$ 。



**定理 4.3.**  $A \in \mathcal{M}_n(\mathbb{R}_{\max})$ , 若  $A$  的 precedence graph  $G(A)$  沒有長度為正值的圈, 則  $A^*$  存在, 並有  $A^* = A^{(n-1)}$ 。而且  $\forall \mathbf{b} \in \mathbb{R}_{\max}^n$ ,  $A^* \otimes \mathbf{b}$  是方程  $\mathbf{x} = A \otimes \mathbf{x} \oplus \mathbf{b}$  的極小解。<sup>5</sup>

**證明.** 從定義  $A^* = \lim_{k \rightarrow \infty} A^{(k)}$ , 只要證明  $A^{(k)}$  的每個元素當  $k \rightarrow \infty$  時收斂即可。由定理 2.1 可知  $A^k$  的元素就是經過  $k$  個邊的路徑中最長路徑之長度, 而  $A^{(k)} = I_n \oplus A \oplus A^2 \oplus \dots \oplus A^k$ , 所以當  $i \neq j$  時由定理 2.1 知  $[A^{(k)}]_{ij}$  的極限就是從頂點  $j$  到頂點  $i$  最長路徑之長度。但是因為  $G(A)$  沒有權重和為正值的圈, 而且  $G(A)$  中經過  $n$  個邊以上的路徑必包含一個圈, 去掉圈後就得到一個不較短的路徑, 所以可知最長的路徑至少有一條其長度小於  $n$ , 也就是說  $[A^{(k)}]_{ij}$  的極限會在  $k < n$  時達到。至於對角線上的元素(也就是  $i = j$  時), 因為首尾相同的路徑就是圈, 故  $\forall k \in \mathbb{N}$ ,  $[A^k]_{ii} \leq 0$ , 又  $A^{(0)} = I_n$ , 所以其極限就是  $e = 0$ 。因此  $A^* = \lim_{k \rightarrow \infty} A^{(k)}$  存在且在  $k < n$  達到其極限, 故  $A^* = A^{(n-1)}$ 。

最後要證明  $A^* \otimes \mathbf{b}$  是  $\mathbf{x} = A \otimes \mathbf{x} \oplus \mathbf{b}$  的極小解, 假設  $\mathbf{y}$  也是一個解, 把  $\mathbf{y} = A \otimes \mathbf{y} \oplus \mathbf{b}$  代入方程本身, 就是

$$\mathbf{y} = A \otimes \mathbf{y} \oplus \mathbf{b} = A \otimes (A \otimes \mathbf{y} \oplus \mathbf{b}) \oplus \mathbf{b} = (A^2 \otimes \mathbf{y}) \oplus (A \otimes \mathbf{b}) \oplus \mathbf{b} = (A^2 \otimes \mathbf{y}) \oplus (A \oplus I) \otimes \mathbf{b}$$

重覆這個步驟  $n - 1$  次, 就會得到

$$\mathbf{y} = (A^n \otimes \mathbf{y}) \oplus (A^{n-1} \oplus \dots \oplus A \oplus I) \otimes \mathbf{b} = (A^n \otimes \mathbf{y}) \oplus (A^{(n-1)} \otimes \mathbf{b}) = (A^n \otimes \mathbf{y}) \oplus (A^* \otimes \mathbf{b})$$

這說明了  $\mathbf{y} \geq A^* \otimes \mathbf{b}$ , 所以  $A^* \otimes \mathbf{b}$  是極小解。□

如果在系統 (4.2) 中,  $G(A)$  包含了一個正長度的圈, 則在此圈上任一事件  $E_i$  的發生時刻就會沿著這個圈上的邊 (邊就是限制式) 限制其他事件, 最後也限制自己的發生時刻, 但是圈的長度為正值, 這就是說  $x_i$  要大於自己, 這是不可能的。所以一個可以運行的時刻表其  $G(A)$  必定不能包含正長度的圈, 於是依定理 4.2 可知  $A^*$  存在, 從而方程組 (4.2) 和 (4.3) 可利用定理 4.2 求解。總結以上的討論就是以下的推論:

**推論 4.2.** 若方程組  $\mathbf{x} = A \otimes \mathbf{x} \oplus \mathbf{d}$  表示一個可運行的時刻表, 則  $A^*$  存在且  $A^* \otimes \mathbf{d}$  是方程組的極小解。

另外定理 4.2 的證明也給出了一個方程組 (4.2) 有唯一解的充分條件。

**推論 4.3.**  $A \in \mathcal{M}_n(\mathbb{R}_{\max})$ ,  $\mathbf{b} \in \mathbb{R}_{\max}^n$ , 若  $G(A)$  沒有圈, 則  $\mathbf{x} = A \otimes \mathbf{x} \oplus \mathbf{b}$  有唯一解。

**證明.** 若  $G(A)$  沒有圈, 則  $G(A)$  中沒有邊數大於  $n - 1$  的路徑, 從  $A^n$  的圖論性質可知  $A^n = \mathcal{E}$ , 從定理 4.2 的證明中知道如果  $\mathbf{y}$  是方程的解, 則

$$\mathbf{y} = (A^n \otimes \mathbf{y}) \oplus (A^* \otimes \mathbf{b}) = (\mathcal{E} \otimes \mathbf{y}) \oplus (A^* \otimes \mathbf{b}) = A^* \otimes \mathbf{b}$$

所以方程組  $\mathbf{x} = A \otimes \mathbf{x} \oplus \mathbf{b}$  有唯一解  $A^* \otimes \mathbf{b}$ 。□

<sup>5</sup>事實上只要  $A^*$  存在,  $A^* \otimes \mathbf{b}$  就是極小解, 但是為了避免繁複的極限操作, 只證明  $A^* = A^{(n-1)}$  的情況, 這樣對大部分的應用而言就足夠了。進一步的討論見 [13], chap. 3-4。



## 4.4 餘裕矩陣與穩定性指標

從上一節的討論可以知道, 給定列車的原始延誤, 時刻表系統(4.2)和(4.3)就可以求解, 也就知道延誤在系統中如何傳遞以及多久後可以恢復準點。但是在排班的階段並不知道會發生什麼樣的延誤, 也不可能在事前計算所有可能的延誤狀況。因此直接求解(4.2)只能知道這個時刻表是否可實現而已, 就是當  $\mathbf{d}$  不是(4.2)的解則此時刻表不可能準點。所以穩定性分析還是要從時刻表系統本身的性質著手, 而從  $A^*$  的性質可以知道許多有用的訊息。

**性質 4.1.** 若方程組  $\mathbf{x} = A \otimes \mathbf{x} \oplus \mathbf{d}$  表示一個可運行的時刻表, 且  $\mathbf{y}$  是方程組的解, 則對任意的  $E_i, E_j \in E$  有  $y_j \geq [A^*]_{ji} \otimes y_i$ 。

**證明.** 因為  $\mathbf{y} \oplus \mathbf{y} = \mathbf{y}$  且  $\mathbf{y} \oplus \mathbf{d} = \mathbf{y}$ , 從  $\mathbf{y}$  是  $\mathbf{x} = A \otimes \mathbf{x} \oplus \mathbf{d}$  的解可以知道

$$\mathbf{y} = \mathbf{y} \oplus \mathbf{y} = A \otimes \mathbf{y} \oplus \mathbf{d} \oplus \mathbf{y} = A \otimes \mathbf{y} \oplus \mathbf{y}$$

所以  $\mathbf{y}$  也是方程組  $\mathbf{x} = A \otimes \mathbf{x} \oplus \mathbf{y}$  的解。從定理4.3可知  $A^* \otimes \mathbf{y}$  是方程組  $\mathbf{x} = A \otimes \mathbf{x} \oplus \mathbf{y}$  的極小解, 故  $\mathbf{y} \geq A^* \otimes \mathbf{y}$ 。於是依定理4.1的推論就知道

$$y_j \geq \bigoplus_{E_k \in E} [A^*]_{jk} \otimes y_k \geq [A^*]_{ji} \otimes y_i$$

即為所證。 □

用圖論的語言來說, 就是從頂點  $i$  到頂點  $j$  的最遠距離就是  $[A^*]_{ji}$ , 這在列車時刻表系統裡就表示事件  $E_i$  到  $E_j$  至少需要多少時間。所以用  $[A^*]_{ji}$  和表定的時刻比較就知道兩事件間有多少餘裕, 於是可以定義以下的餘裕矩陣。

**定義 4.7** (餘裕矩陣). 若方程組  $\mathbf{x} = A \otimes \mathbf{x} \oplus \mathbf{d}$  表示一個可運行的時刻表, 定義餘裕矩陣  $R = (r_{ij})$ :  $r_{ij} = d_i - d_j - [A^*]_{ij}$ , 如果  $[A^*]_{ij} = \varepsilon$  則定義  $r_{ij} = \infty$ 。 ||

從餘裕矩陣中可以知道每個事件遇到多少延誤時會影響到其他事件, 所以也知道可以容許多少延誤而不影響其他事件。更具體地說,  $R$  的第  $i$  行元素表示當  $E_i$  發生多少延誤時會影響其他事件, 而  $R$  的第  $j$  列元素就表示了事件  $E_j$  如何被其他事件影響。參考 Goverde 的用語,  $R$  的行向量又稱為延誤衝擊向量 (delay impact vector), 列向量則稱為延誤敏感向量 (delay sensitivity vector)。<sup>6</sup>

Pachl 於[28] sec. 6.6 中提到, 穩定度可以用穩定商數(stability quotient)來衡量。但是 Pachl 是在週期時刻表的假設下定義穩定商數, 本研究則依仿照其概念定義穩定商數, 以適合非週期時刻表。

<sup>6</sup>參見14頁。

**定義 4.8** (穩定商數). 對任一時刻表系統的事件  $E_i \in E$ , 定義穩定商數  $q_i$  為系統恢復準點所需時間與初始延誤之比值。 ||

穩定商數代表了系統恢復準點的能力, 穩定商數愈高表示需要愈多時間才能恢復準點, 所以穩定度差, 反之穩定商數低則表示穩定度好。從這個定義可知穩定商數不只與鐵路時刻表系統本身有關, 也和初始延誤的量有關, 給定不同的初始延誤會得到不同的值, 換言之就是初始延誤與恢復準點所需的時間通常不是線性關係。因此在實際計算時須給定初始延誤, 以德國鐵路計算穩定商數為例其初始延誤設定為10分鐘[28]。計算穩定商數的方法有二: 其一是給定初始延誤後直接求解式(4.3); 二是從餘裕矩陣計算, 給定初始延誤  $\delta_0$ , 對任意的  $E_i \in E$  只要在  $R$  的第  $i$  行中找出所有餘裕小於  $\delta_0$  的  $r_{ji}$ , 求出所對應之事件中最晚的發生時刻, 在這個時刻後系統就可以視為恢復準點, 於是就有

$$q_i = \frac{\max_{j:r_{ji} < \delta_0} \{d_j\} - d_i}{\delta_0} \quad (4.6)$$

如果已經求出餘裕矩陣, 那麼利用上式求穩定商數會比直接求解時刻表系統要有效率。

## 4.5 有關演算法的討論

由定理4.2求解 max-plus 系統的關鍵在於求擬反矩陣, 本節討論如何有效地求出一個矩陣的擬反矩陣。以下假設  $A \in \mathcal{M}_n(\mathbb{R}_{\max})$ , 依定理4.2和推論4.2可知若  $A$  代表一個可運行時刻表, 則  $A^* = A^{(n-1)} = I \oplus A^2 \oplus A^3 \oplus \dots \oplus A^{n-1}$ 。如果依此定義進行計算, 對於大型時刻表系統計算時間將非常的長。如果一個數的  $\oplus$  和  $\otimes$  運算時間是  $O(1)$ , 那麼這個計算要有  $n(n-1)/2$  次的矩陣乘法和  $n-1$  次的矩陣加法, 矩陣乘法如果用一般的方法依定義實作其時間複雜度是  $O(n^3)$ , 而矩陣加法是  $O(n^2)$ ,<sup>7</sup>所以總和是  $n(n-1)O(n^3)/2 + (n-1)O(n^2) = O(n^5)$ ! 這實在不是一個好的演算法。

以下的演算法被稱為 Jacobi iteration, 是一個比較有效率的演算法[18]。

**演算法 1** (Jacobi iteration). 給定  $A \in \mathcal{M}_n(\mathbb{R}_{\max})$ , 若  $A$  代表一個可運行的時刻表, 則這個演算法將求出  $A^*$ 。

1. [開始] 令  $x \leftarrow I$ 。
2. [迭代] 令  $y \leftarrow A \otimes x \oplus I$ 。
3. [結束?] 若  $x = y$ , 輸出  $x$  並結束計算, 否則令  $x \leftarrow y$  並回到上一步。

<sup>7</sup>細節請參見[14]。

**定理 4.4.** *Jacobi iteration* 的時間複雜度是  $O(n^4)$ 。

**證明.** 令演算法第一步開始時的  $x$  為  $x_0$ , 第二步的迭代中, 第  $i$  次迭代時的  $x$  為  $x_i$ , 於是  $x_0 = I = A^{(0)}$ ,  $x_1 = A \oplus I = A^{(1)}$ , 依此類推可得  $x_i = A^{(i)}$ 。因為  $A^* = A^{(n-1)}$ , 所以這個演算法只要做  $n - 1$  次迭代, 而每次迭代需一次矩陣乘法和加法, 所以總共是  $n - 1$  矩陣乘法和  $n - 1$  次的矩陣加法, 其時間複雜度就是  $(n - 1)(O(n^3) + O(n^2)) = O(n^4)$ 。□

實際操作這個演算法時,  $A$  通常是稀疏矩陣(sparse matrix), 也就是說大部分的元都是零元 ( $\varepsilon$ ), 如果設計特別的資料結構以儲存  $A$  使得在進行矩陣乘法時只要計算非零元的部分, 則計算時間可以減少許多。如果  $A$  中非零元的數量是  $m \ll n^2$ , 則計算時間可以降到  $O(mn^2)$ [18]。

另外 *Jacobi iteration* 還有一個優點是它可以利用平行計算加速, 從演算法的步驟3可以知道這個迭代的步驟能逐行或是逐列計算, 所以能用多部電腦分別進行計算。如果有  $n$  部電腦, 且當  $A$  為稀疏矩陣時, 計算的時間複雜度可以降到  $O(mn)$ 。進一步如果已知  $A$  表示一可實現的時刻表, 將能更有效率地計算  $A^*$ 。

**定理 4.5.** 若  $A$  表示一可實現的時刻表, 定義  $S = (s_{ij}) \in \mathcal{M}_n(\mathbb{R}_{\max})$  使得  $s_{ij} = d_j - d_i + a_{ij}$ , 則  $S$  滿足

1.  $\forall 1 \leq i, j \leq n, s_{ij} \leq e$
2.  $G(S)$  沒有正長度的圈
3.  $\forall 1 \leq i, j \leq n, [S^*]_{ij} = d_j - d_i + [A^*]_{ij}$

**證明.** 因為  $A$  表示一可實現的時刻表, 必有  $d_i \geq d_j + a_{ij}$ , 所以  $s_{ij} = d_j - d_i + a_{ij} \leq e$ 。由此可推論  $G(S)$  的每個邊權重都小於  $e$ , 故沒有正長度的圈。最後考慮  $G(S)$  中由頂點  $j$  到頂點  $i$  的任一路徑  $p = (j, k_1, \dots, k_l, i)$ , 其長度為

$$\begin{aligned} \sum_{(\alpha, \beta) \in p} s_{\beta\alpha} &= s_{k_1 j} + \dots + s_{i k_l} \\ &= d_j - d_{k_1} + a_{k_1 j} + \dots + d_{k_l} - d_i + a_{i k_l} \\ &= d_i - d_j + \sum_{(\alpha, \beta) \in p} a_{\beta\alpha} \end{aligned}$$

所以所有  $G(S)$  中從  $j$  到  $i$  的路徑其長度與  $G(A)$  中相對應之路徑只差一常數  $d_i - d_j$ , 而  $[S^*]_{ij}$  是  $G(S)$  中從  $j$  到  $i$  最長的路徑長, 就等於  $G(A)$  中從  $j$  到  $i$  最長的路徑長加上  $d_i - d_j$ , 就是  $d_j - d_i + [A^*]_{ij}$ 。□

從這個定理可知求  $A^*$  可以轉換成求  $S^*$ , 而  $G(S)$  沒有正長度的圈, 所以求  $[S^*]_{ij}$  就是在找  $G(S)$  中從  $j$  到  $i$  的最長路徑長, 這又相當於找  $G(-S)$  中從  $j$  到  $i$  的最短路徑長。因為

$s_{ij} \leq e$ , 所以  $G(-S)$  所有的邊均不為負值, 於是可以利用 Dijkstra 演算法找  $G(-S)$  中的最短路徑。如果以 Fibonacci heap 和 min-priority queue 來實作 Dijkstra 演算法, 則計算時間只要  $O(n^2 \log n + mn)$ , 比 Jacobi iteration 的  $O(mn^2)$  要來得好。<sup>8</sup>

## 4.6 穩定性方析方法比較

本研究參考 Goverde 的方法[19, 20], 去除週期時刻表的假設, 發展了可應用於非週期時刻表的穩定性分析, 這一節將比較兩方法的異同。

Goverde 將每個週期視為狀態,  $A(\gamma)$  就是狀態轉移的規則, 從 spectral analysis (也就是求廣義特徵值與特徵向量) 得到時刻表系統的定性指標, 由此可以判定時刻表的穩定性與可實現性。而本研究的對象是非週期時刻表, 沒有漸近穩定的概念, spectral analysis 並不適用, 因此本研究從求解時刻表系統的角度出發, 從求解的結果可以判定時刻表是否可實現與可運行。

在定量的指標方面兩方法可說是殊途同歸, 都是從求解最長路徑問題得到系統的餘裕量。雖然在 Goverde 的論文中明確假設時刻表是週期時刻表, 並且得到恢復矩陣  $r_{ij} = d_i^0 - d_j^0 - [A_T^+]_{ij}$ , 但是如果時刻表不是週期時刻表時, 系統(2.2)就是一個零階系統,<sup>9</sup> 那麼就有  $A_T^+ = A(T^{-1}) = A_0$ , 矩陣  $A_0$  會與本研究系統(4.2)中的  $A$  一致, 於是理論上, Goverde 的恢復矩陣(recovery matrix) 應該就會和本研究的餘裕矩陣相同。不過由於兩者模化的方式並不完全一致, 所以實際上計算出來的結果還是會不一樣。由此觀之可以說本研究的方法在這一部分是 Goverde 方法的一種特例, 只是 Goverde 的研究中沒有說明非週期時刻表的情況。不過由於本研究沒有週期時刻表的假設, 所以模化時就顯得單純許多, 理論推導也比較簡單。

最後一提的是兩個方法採用不同的角度進行模化。從[19]可以看出 Goverde 是用圖論的方式模化並建構時刻表系統, 將時刻表模化成 timed event graph, 再從 timed event graph 建立 max-plus 系統, 其實作的演算法也都是圖論演算法; 而本研究則直接從代數的角度切入, 直接將時刻表的限制寫成 max-plus 代數的方程組, 實作時則採用 Jacobi iteration。兩種方式各有其長處, 圖論方法具有直觀的解釋, 容易理解其原理; 而代數方法則是比較直接明快, 容易實際操作。

<sup>8</sup>Dijkstra 演算法的細節請參閱[14], chap. 25。

<sup>9</sup>參見第 10 頁。



## 第五章 實例研究與應用

### 5.1 研究範圍概況

本研究以臺鐵基隆到新竹路段時刻表為實例進行穩定性分析。時刻表取自國立交通大學鐵道研究會繪製之列車運行圖[3]，版本是第五版，為九十八年六月十六日的時刻表；路線、股道、月台及運轉規則等資料主要參考交通部運輸研究所出版之一系列有關臺鐵系統排班與容量的研究報告[7, 8, 9, 10, 11, 12]。

基隆-新竹段全長106.4公里，全線皆為電化區間，號誌系統為雙單線號誌系統，但是在排班時均假定為複線運轉，故本研究亦假定此路段為複線運轉。此路段中共有二十七個車站，<sup>1</sup>上下行合計共有379車次，其中部分車次僅在一星期中的部分日期行駛，包括東、西部幹線之長途列車和區間車，為全臺環島路網中最繁忙的一段。

本研究參考[8]將各站軌道的數量及運用方式整理成表5.1。由於近年臺鐵工程頻繁，運研所報告中部分資料已與現況不符，以舊的路線資料配合後來的時刻表進行分析，恐有不盡合理之處，故本研究參考目前現況更新部分路線資料。依臺鐵之運轉規章規定，站內只有正線能辦理列車到開，因此表5.1中之軌道數均只包括正線。其中又只有主正線能辦理列車通過，故表中將主正線數另列一欄。雖然假定為複線運轉，但是路線中部分車站的軌道運用並沒有將上下行完全分開，站內部分的軌道是上下行共用，表中的上行與下行軌道數均包含共用的軌道。

### 5.2 模化與計算

首先根據資料，依4.1節所述之方法將基隆-新竹段的時刻表模化為 max-plus 線性系統。以下說明模化的過程和考慮的事項。

1. 列車續進限制：在這個限制中，原則上應該要知道每列車在站間的最小運轉時間，但是最小運轉時間通常要藉由運轉曲線的模擬或多車測試來得到[12]，並不能直接從時刻表中知道。由荷蘭的經驗，時刻表定的運轉時間平均含有7%的餘裕[19]，如果表定的站間運轉時間為7分鐘，餘裕大概就是30秒。考慮到30秒是現階段臺鐵排班時的最小時間單位，且在此路段中鮮少有站間運轉時間超過7分鐘，本研究採取較寬容的估計，假設每列車在每兩站間運轉時間的餘裕均為30秒，所以最小運轉時間就是表定時間減去30秒。
2. 列車停站限制：同樣的，原則上也應該用最小停站時間，但是這同樣不能從時刻表中得到，

<sup>1</sup>依[3]中的運行圖有二十七個車站，其中「新貨」旅客列車並不停靠。

表 5.1: 臺鐵基隆—新竹段各車站軌道數資料

車站	軌道總數	上行	下行	共用	上行臨月台	下行臨月台	上行主正線	下行主正線	到開線 <sup>†1</sup>
基隆	8	8	8	8	3	3	1	1	0
三坑	2	1	1	0	1	1	1	1	0
八堵 <sup>†2</sup>	4	2	2	0	2	2	2	2	0
東	2	1	1	0	1	1	1	1	0
西	2	1	1	0	1	1	1	1	0
七堵	4	2	2	0	2	2	1	1	4
百福	2	1	1	0	1	1	1	1	0
五堵	2	1	1	0	1	1	1	1	0
汐止	3	2	2	1	2	2	1	2	0
汐科	2	1	1	0	1	1	1	1	0
南港	4	2	2	0	2	2	2	2	0
松山	5	2	3	0	2	3	1	1	0
台北	5	3	3	1	2	2	2	1	0
萬華	3	2	2	1	2	2	1	1	0
板橋	5	3	2	0	2	2	1	1	0
樹林	4	2	2	0	2	2	1	1	1
山佳	4	2	2	0	2	2	1	1	0
鶯歌	4	2	2	0	2	2	1	1	0
桃園	4	3	2	1	2	2	1	1	0
內壢	3	2	2	1	2	2	1	1	0
中壢	4	2	2	0	2	2	1	1	0
埔心	3	2	2	1	2	2	1	1	0
楊梅	4	2	2	0	2	2	1	1	0
富岡	4	2	2	0	2	2	1	1	0
湖口	3	2	2	1	2	2	1	1	0
新豐	4	2	2	0	2	2	1	1	0
竹北	3	2	2	1	2	2	1	1	0
新貨 <sup>†3</sup>	2	1	1	0	0	0	1	1	0
新竹	5	4	4	3	4	4	1	1	0

<sup>†1</sup> 到開線只供以該站為起點或終點之列車到開使用，總軌道數不包含到開線的數量。

<sup>†2</sup> 八堵站是縱貫線(西部幹線)與宜蘭線(東部幹線)之交會站，兩線分別使用不同的軌道和月台。

<sup>†3</sup> 新貨為新竹貨運站，在[8]中沒有其月台軌道資料，而在[3]中有標示出來，因為旅客列車並不會停靠，故設定該站沒有月台，僅有兩通過線。



而且當列車發生延誤時經常導致列車為了等候旅客上下車使得停站時間增加。因為在列車續進限制中已經偏向給予較多餘裕，所以列車停站限制就設定為最小停站時間就是表定的停站時間，另外如果有停站待避其他列車時則設為3分鐘。

3. 運轉時隔限制: 簡單而言運轉時隔限制可分為三類: 進站時隔、離站時隔和到開時隔, 由於是複線運轉, 進站與離站時隔均只要考慮同方向的列車。研究報告[12]設定同向離站時隔設定為2.5分鐘, 同向到站時隔設定為3分鐘, 同股道到開時隔則設定為5分鐘。但是如果用這個數字去計算就會有許多事件間的餘裕為負值, 仔細檢視時刻表後發現實際排班允許的時隔是比上述數字要小, 因此計算結果就會顯得列車間隔不足。然而研究報告[12]所設定的運轉時隔是供排班用, 已經包含了若干餘裕, 實際運轉允許的時隔應當比較小一些。在不知道實際最小運轉時隔的情況下, 本研究將上述的值略為減少, 設定同向離站時隔為2分鐘, 同向到站時隔為2.5分鐘, 同股道到開時隔則為4分鐘。特別值得一提的是八堵站的情況, 因為八堵是縱貫線與宜蘭線的交會站, 兩線的列車分別使用不同的月台和軌道, 實際的交會點在車站的下行側, 所以在進站與離站時隔的設定上, 上行側縱貫線與宜蘭線的列車應分開考慮, 而下行側要一起考慮。
4. 站內軌道與月台數限制: 由於軌道配置的關係, 許多站都有上下行共用的軌道, 除了要滿足個別方向的軌道數限制外, 也要滿足總軌道數的限制。另外在七堵和樹林有到開線, 以七堵或樹林為起終點的列車可以不占用正線軌道。另外由於從時刻表中不能知道列車實際的運用狀況, 也不知道抵達終點的列車是否會持續占用軌道, 只好假設列車在抵達終點後5分鐘後即駛離並空出軌道。雖然臺鐵規定主正線方能辦理列車通過, 不過站間每個方向只有一軌道, 所以也不會有兩列車同時通過的狀況。如果發生延誤使得站內的軌道都被前方列車占用, 在運轉順序不變的前提下, 那麼要通過的列車也要於站內停等, 因此是不是使用主正線也就不重要了。
5. 禁止列車於同一軌道追越: 因為假設系統為複線運轉, 所以這個限制自然就表現為禁止列車於站間追越, 沒有問題。
6. 不考慮的限制:
  - (a) 因為是複線運轉, 當然沒有單線運轉的限制。
  - (b) 因為不知道列車實際的運用狀況, 因此假設上下行列車間均沒有接續關係。
7. 其他: 有些列車是假日的加班車, 另外有些則例假日停駛, 以致於一週中每天的時刻表都有一些不同。但是仔細檢視運行圖時發現除了星期二之外, 其他運行圖都有不合理之處, 大多是加班的列車於站間追越其他列車以及沒有保持運轉時隔。不合理的時刻表可能會導致擬反矩陣不存在, 在計算時餘裕矩陣會有部分的元為  $\infty$ 。因為不考慮運轉整理, 所以本研究決定以可運行的星期二時刻表進行分析。

在上述的考慮之下，本研究以 C++ 語言撰寫電腦程式進行計算。使用 GCC 編譯器及 Dev-C++ 4.9.9.2 開發環境，在 Windows XP™ SP3 作業系統下執行，並於編譯時開啓所有最佳化選項以節省計算時間。計算的步驟和說明如下：

1. 模化時刻表：將時刻表中所有的到開事件編號，然後依 4.1 節所述的規則確認每個事件間的限制式，以建立描述這個系統的 max-plus 矩陣。本研究分析之時刻表共有 316 列車，模化為 11932 個事件以及 30633 個限制式。
2. 計算擬反矩陣：本研究以 Jacobi iteration 計算擬反矩陣。因為時刻表系統龐大，所以將矩陣以稀疏矩陣的方式儲存（即只存非零元），並採逐列計算的方式以免耗盡記憶體。
3. 求餘裕矩陣和穩定商數：依定義計算出餘裕矩陣後，利用式 (4.6) 計算穩定商數。本研究設定的初始延誤為 5 分鐘。

使用的電腦為 Intel® Core™ 2 Duo E8200，以及 2GB DDR2 800 的記憶體，大約耗時 16 分鐘，其中的 13 分鐘以上都用於計算擬反矩陣。就靜態的時刻表分析而言完全是可以接受的時間，如果要用於動態的運轉整理評估，應可利用較好的電腦及平行運算來減少計算時間，對於更大的系統而言，實作更有效率的資料結構也是可以考慮的方法。

### 5.3 結果分析與討論

計算結果主要是餘裕矩陣與每個事件的穩定商數，本節討論這些數據的意義。

時刻表模化後共有 11932 個事件，所以餘裕矩陣是一個  $11932 \times 11932$  的矩陣  $R$ ，矩陣中的每個元素  $r_{ij}$  代表了事件  $E_j$  到事件  $E_i$  有多少餘裕。舉例來說，如果台北站某列車的出發事件編號為 10，桃園站的另一列車之抵達事件編號為 20，則矩陣中的  $r_{20,10}$  就代表前一個事件到後一個事件間的餘裕。換句話說就是前一事件可以延誤  $r_{20,10}$  而不影響到後一事件。如果  $r_{ij}$  是  $\infty$ ，即餘裕為無窮大，則表示事件  $E_j$  的延誤不會影響到  $E_i$ 。若餘裕是負值，則表示系統會自己產生延誤，該事件就無法準點。另外值得一提的是，如果有在列車在站間追越其他列車或於軌道數不足的站進行追越等，則擬反矩陣不存在，<sup>2</sup> 這個情況下 Jacobi iteration 中不收斂的元可視為  $\infty$ ，於是相對應的餘裕就是  $-\infty$ 。餘裕是負無窮大也就表示時刻表根本無法運行，這一點可以用來找出時刻表中不合理之處。

在 43 頁的表 5.2 是一部分的餘裕矩陣。事件 1–5 依序分別為南下 2133 次區間車七堵的出發事件、百福的抵達事件、百福的出發事件、五堵的出發事件和五堵的抵達事件。事件 50–55 為南下 11 次莒光號以及事件 98–102 是南下 1049 次自強號從七堵到五堵的出發與抵達事件。以表 5.2 的第 1 行為例，這一行是事件 1 的延誤衝擊向量，表示事件 1 可以發生多少延誤而不影響其他事

<sup>2</sup>參見 34 頁的討論。

表5.2: 餘裕矩陣示意

$r_{ij}$	1	2	3	4	5	...	50	51	52	53	54	55	...
1	0	$\infty$	$\infty$	$\infty$	$\infty$		$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	
2	30	0	$\infty$	$\infty$	$\infty$		$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	
3	30	0	0	$\infty$	$\infty$	...	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	...
4	60	30	30	0	$\infty$		$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	
5	60	30	30	0	0		$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	
⋮			⋮			⋮			⋮				
50	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$		0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	
51	660	$\infty$	$\infty$	$\infty$	$\infty$		0	0	$\infty$	$\infty$	$\infty$	$\infty$	
52	570	540	540	$\infty$	$\infty$		30	30	0	$\infty$	$\infty$	$\infty$	
53	570	540	540	$\infty$	$\infty$	...	30	30	0	0	$\infty$	$\infty$	...
54	540	510	510	480	480		60	60	30	30	0	$\infty$	
55	540	510	510	480	480		60	60	30	30	0	0	
⋮			⋮						⋮				
98	1680	$\infty$	$\infty$	$\infty$	$\infty$		1020	1020	$\infty$	$\infty$	$\infty$	$\infty$	
99	1530	1500	1500	$\infty$	$\infty$		990	990	960	$\infty$	$\infty$	$\infty$	
100	1530	1500	1500	$\infty$	$\infty$	...	990	990	960	990	$\infty$	$\infty$	...
101	1500	1470	1470	1440	1440		1020	1020	990	990	960	$\infty$	
102	1500	1470	1470	1440	1440		1020	1020	990	990	960	990	
⋮			⋮						⋮				⋮

表5.3: 事件說明、穩定商數與容許延誤示意表

事件編號	1	2	3	4	5	50	51	52	53	54	55	98	99	100	101	102
車次	2133					11						1049				
車種	區間					莒光						自強				
車站	七堵	百福	百福	五堵	五堵	七堵	七堵	百福	百福	五堵	五堵	七堵	百福	百福	五堵	五堵
事件類型	出發	抵達	出發	抵達	出發	抵達	出發	抵達	出發	抵達	出發	出發	抵達	出發	抵達	出發
表定時刻	4:33.0	4:36.5	4:37.0	4:40.0	4:40.5	4:43.0	4:46.0	4:50.0	4:50.0	4:52.5	4:52.5	5:05.0	5:08.5	5:08.5	5:11.0	5:11.0
容許延誤	540	510	510	480	480	810	810	780	780	750	750	120	90	150	150	210
穩定商數	8.6	9.1	9.0	9.3	9.2	9.7	9.1	9.7	9.7	10.0	10.0	8.0	8.4	8.4	8.7	8.7

附註: 餘裕矩陣之單位為秒, 表5.3的表定時刻中.0為00秒, .5為30秒。容許延誤為在不影響其他列車的條件下該事件可以發生多少延誤, 單位亦為秒。

件。例如第1行的第51個元素是660, 所以2133次區間車如果在七堵延誤超過660秒出發就會導致11次莒光號在七堵出發也隨之延誤。第1行的第98個元素為1680, 表示若2133次區間車在七堵的出發延誤超過1680秒就會導致1049次自強號在七堵出發也延誤。第1列是事件1的延誤敏感向量, 表示其他事件可以發生多少延誤而不影響事件1。餘裕矩陣的第1列除了第1行外都是 $\infty$ , 表示其他事件的延誤都不會影響事件1, 這其實是因為事件1是整個時刻表中最早的事件。

在餘裕矩陣的同一行中找出最小的值, 就可以知道該行對應的事件可以允許多少延誤而不影響其他事件。不過實務上更在意的是可以允許多少延誤而不影響其他「列車」, 這只要在找最小值時去除同一列車的那幾列, 這樣就得到表5.3最後一列的容許延誤, 即在不影響其他列車的條件下該事件可以發生的延誤量。

進一步如果某個事件發生了延誤, 就在相對應的那一行中找出餘裕小於延誤量的元素, 這些元素所在的位置就是會被影響的事件。用這個想法就可以依式(4.6)計算穩定商數。從定義4.8穩定商數就表示系統當恢復準點的快慢, 本研究計算時設定初始延誤為5分鐘, 於是事件的穩定商數就是當該事件發生5分鐘的延誤時系統恢復準點所需的時間除以5分鐘, 所以從穩定商數可以看出當該事件發生延誤時對整個系統的影響有多大。表5.2最後一列列出一部分事件的穩定商數, 例如事件1的穩定商數為8.6, 表示當事件1發生5分鐘的延誤時, 在沒有運轉整理的情況下系統要 $5 \times 8.6$ 即43分鐘後才會恢復準點。

穩定商數可以作為評估時刻表穩定性的一個指標, 例如全體事件的穩定商數平均就可以用來表示時刻表整體的穩定度。<sup>3</sup>本研究的時刻表全體事件穩定商數平均值為7.271, 也就是說如果系統隨機發生一個5分鐘的延誤, 在沒有運轉整理的情況下平均約36分鐘後系統才能恢復到準點。也可以將每個車站的事件分開來看, 從不同車站的事件穩定商數之差異來找出比較不穩定的區間。表5.4是每個車站所有列車穩定商數之平均值, 從表中也可以見到在中間車站的穩定商數明顯比較高, 靠近基隆和新竹兩端的車站穩定商數比較低, 這是因為在終點站延誤不會傳播到其他列車, 如果考慮列車的運用時可能就不會是這樣的結果。從各站穩定商數平均值可以觀察可能的瓶頸車站, 例如桃園站及前幾站的上行部分穩定商數明顯比鶯歌高出許多, 表示桃園站可能是上行方向的瓶頸。

除了從車站的角度分析, 還可以檢視每一列車的穩定商數最大值, 找出使時刻表不穩定的列車以謀求改善之道。例如結果中有不少列車的穩定商數最大值都在12左右, 這表示發生5分鐘的延誤, 系統在一小時後才能恢復準點, 對於這些列車就可以考慮調整時刻表或事先準備運轉整理方案以減少延誤傳遞。

實際應用時由於餘裕矩陣十分龐大, 很難逐一去檢視, 本研究建議計算時可以將結果標記於

<sup>3</sup>如果能知道初始延誤的機率分佈當然更好, 就可以用模擬來得到整個時刻表的平均穩定商數的估計值。



表 5.4: 各車站穩定商數平均值

車站	基隆	三坑	八堵	七堵	百福	五堵	汐止	汐科	南港	松山
下行	7.085	8.025	7.808	8.351	9.027	9.040	9.177	9.911	9.986	9.979
上行	0.131	1.048	1.210	2.266	3.182	3.772	4.543	4.999	5.795	6.409
平均	3.634	4.562	4.509	5.320	6.126	6.425	6.877	7.473	7.906	8.204
車站	台北	萬華	板橋	樹林	山佳	鶯歌	桃園	內壢	中壢	埔心
下行	9.121	8.630	8.248	8.352	8.989	8.991	8.017	7.488	7.167	6.454
上行	7.594	8.358	8.725	9.022	9.210	9.627	10.669	10.908	10.934	10.549
平均	8.360	8.509	8.486	8.687	9.100	9.309	9.343	9.198	9.050	8.502
車站	楊梅	富岡	湖口	新豐	竹北	新貨	新竹	平均		
下行	5.734	4.676	3.731	2.741	1.733	0.781	0.213	7.390		
上行	9.826	9.799	9.492	9.408	9.369	8.586	7.756	7.151		
平均	7.780	7.237	6.612	6.074	5.541	4.663	3.984	7.271		

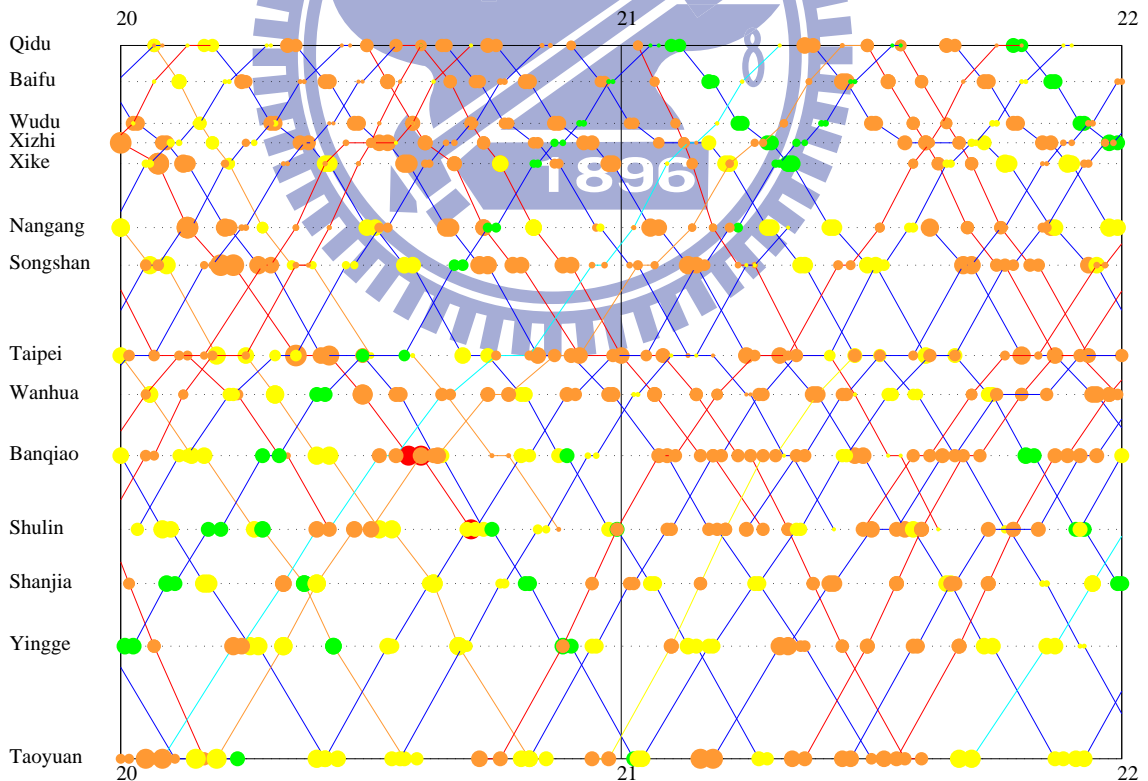


圖 5.1: 帶有穩定性標記的運行圖

運行圖上，如此一來從運行圖中就能直接得到相關的資訊。標記的方式可以視需要自行設計，以下是一個例子，圖 5.1 是七堵到桃園一部分的運行圖，圖中加上了有關穩定性的標記。標記顏色表示容許延誤的多寡，綠色為 10 分鐘以上，黃色為 5-10 分鐘，橘色是 0-5 分鐘，紅色則為負值！負值表示該事件不僅不能延誤，還要提早發生才不會影響其他列車，這不是很合理，可能的原因包括計算時設定之運轉時隔過小或是排班太緊湊。標記的大小則表示穩定商數的量，愈大穩定商數就高。從圖中各列車的穩定性可以一目瞭然，有助於尋找時刻表不穩定之處。

## 5.4 穩定性分析之應用

以下討論本研究發展之穩定性分析方法一些可能的應用方式，可供鐵路營運單位參考。

1. 時刻表選擇、偵錯及調整：在排班階段可能會產生一些候選的時刻表，穩定性分析可以提供一個決策時的參考依據，在沒有其他考量下，選擇穩定度較高的時刻表比較好。另外在 5.3 節提到穩定性分析方法也可以用於偵測不合理的排班以及找出使時刻表不穩定的事件，然後可以針對它們進行調整與改善。具體地說如果擬反矩陣的某些元是  $\infty$ ，也就是說擬反矩陣不存在，則依定理 4.2 此時刻表一定不可運行；另外如果餘裕矩陣的元是負值也表示時刻表排班有可能太緊湊。檢視相應的事件可以找出時刻表的錯誤與不恰當之處。
2. 運轉整理：在實際營運時，如果有列車發生延誤，馬上可以從餘裕矩陣估計延誤擴散的程度，以決定是否要進行運轉整理。如果有足夠的計算資源，也可以研擬運轉整理策略，例如變更列車順序或取消列車等，然後求解運轉整理後之時刻表系統來評估各種策略的效果。
3. 設施改善方案評估：鐵路營運單位在評估各種設施設備的改善方案時，穩定性分析可以作為方案效益的指標之一。應用的方式可能是用現行時刻表，考慮各種改善方案模化為不同的系統，比較穩定度的差異。實際應用時應結合成本效益分析來決定最佳的方案。
4. 列車服務可靠度分析：可靠度一般都用列車的準點率來表示，而本研究的方法也可以用於估計準點率。首先要由過去的資料估計列車延誤的機率分佈，<sup>4</sup> 然後隨機產生延誤，求解式 (4.3) 然後統計準點事件的比率，重覆許多次後就可以得到準點率的估計值。如果假設沒有運轉整理，則每一次的計算都可以用相同的擬反矩陣，可以很迅速地計算出結果，但是實際運轉時遇到大的延誤時都會進行運轉整理，因此計算的結果並不會接近實際的準點率。如果參考 [1] 的作法，考慮一些簡單的運轉整理規則，會更貼近實際的狀況，但是這樣一來每次都要重新計算擬反矩陣，需耗費較多的時間。<sup>5</sup> 如何將運轉整理的因素考慮進來而且還能迅速計算出結果尚待進一步研究。

<sup>4</sup>當然列車也很可能會早到，所以更合理的做法是估計運轉時間的機率分佈。也可以參考 [1] 的做法，允許延誤值為負值，不失為一種簡便的好方法。

<sup>5</sup>實際上擬反矩陣不用全部重算，只要算延誤發生後的部分即可，這是因為 Jacobi iteration 可以逐行計算。



## 第六章 結論與建議

### 6.1 結論

穩定的時刻表對鐵路營運而言相當重要，本研究參考 Goverde 先前的成果，去除其週期時刻表的假設，並略為改進模化方式，以 max-plus 代數建立可用於傳統鐵路的時刻表穩定性分析方法，從理論的角度證明了可運行的時刻表系統可以求解並能計算出餘裕矩陣與穩定商數，而餘裕矩陣和穩定商數可以用於衡量時刻表的穩定度。

本研究以臺鐵基隆—新竹路段之時刻表為實例，分析其穩定性。以臺鐵的時刻表實際計算時發現餘裕矩陣中有不合理的值，經檢視時刻表後確有不合理之衝突或排班過度密集的現象，這表示從餘裕矩陣的計算結果還可以找出時刻表中不合理之處，可以用於時刻表的檢驗與偵錯。

在選擇合理的時刻表後計算出穩定度指標，這些指標代表時刻表的穩定程度。計算結果表明 max-plus 方法確實可以有效地應用於實際的鐵路系統中，作為鐵路營運單位的決策輔助工具。本研究並指出穩定性分析還能應用於時刻表選擇及調整、運轉整理、設施改善方案評估以及列車服務可靠度分析。

### 6.2 後續研究建議

由於研究時程與人力的限制，本研究的成果還有許多可以改進與深入探討之處。

首先從實際應用層面來看，由於資料不充分，例如站間運轉時間、最小停站時間與運轉時隔都是本研究自行推估設定，還有本研究也沒有車輛運用的資料，因此本研究無法對實際鐵路系統的穩定性分析做出結論。這方面有賴鐵路營運單位進行測試與分析以提出適當的數值，還需要更詳細的排班資料，才能使穩定性分析結果更貼近鐵路系統的真實狀態，得到真正有用的結果。分析的結果也顯示臺鐵的時刻表有不合理之處，傳統鐵路的排班問題相當複雜，如何改善臺鐵的排班，使得時刻表可以合理地運行是個重要的課題。

就理論的角度而言，雖然在 max-plus 代數的架構下，許多列車運轉的限制都可以模化成 max-plus 的方程組，但是仍有一些限制沒有被納入這個架構。例如車站內的軌道佈設有平面交叉時，列車可能在平面交叉處發生衝突，這一點就無法被反應在模化後的方程組中。如何使模化更接近真實的情況需要再研究。

本研究假設每列車的站間運轉時間與停站時間都是固定值，如果將站間運轉時間及停站時

間視為隨機變數會更符合實際情況，這就導出隨機 max-plus 系統的概念。這方面有一些理論結果[23]，但能實際應用到鐵路系統上的成果似乎還相當少，可能是因為隨機 max-plus 系統相當複雜，也比較難計算。要如何引入隨機的因子又不能犧牲太多計算時間，還有待研究。

模化時刻表時一個重要假定是列車的運轉順序固定，所以遇到列車順序改變的情況就要修改限制式重新求解一次時刻表系統，餘裕矩陣的計算也是假設沒有運轉整理。但是運轉整理經常發生，每次都要重新計算則效率不好。對於時刻表系統中少許的變動，如果可以從原本的結果中迅速推算變動後的結果，則更能應用於實際營運時的即時決策。

一個重要但困難的問題是：如何應用 max-plus 代數理論到實際的排班問題中，更具體地說就是如何在 max-plus 代數上進行最佳化，例如排班時在滿足旅客需求的條件下，求取穩定度最好的時刻表。但是何謂最佳穩定度也需要進一步探討。

最後要提出來的是穩定性與容量的關係，以及如何以 max-plus 代數進行軌道容量的研究。這兩個問題其實是一體兩面，如果知道如何以 max-plus 代數分析容量，則透過 max-plus 代數穩定性與容量的關係應該就不難建立；反之如果知道穩定性與容量的關係，則 max-plus 代數於穩定性分析的成果也能應用於容量的研究。容量分析一直都是軌道研究中的一個重要問題，也有一些 max-plus 代數應用於容量研究的成果[15]，只是看起來其架構都還相當簡單，如何應用到更複雜的鐵路系統中還尚待努力。

Max-plus 代數方法應用於鐵路時刻表研究與常見的模擬模式相較，具有模化過程簡單、可以直接分析時刻表結構以及容易計算等優點，但相關的研究成果不論是理論或實際應用都還不多，希望本研究可以達到拋磚引玉的效果，使國內有更多人投入這個領域。

## 參考文獻

- [1] 周學怡,「列車運行可靠性分析」, 國立成功大學交通管理學系, 碩士論文, 民國八十五年。
- [2] 容志輝,「基本線性系統理論」, 全華科技圖書股份有限公司, 民國九十二年。
- [3] 國立交通大學鐵道研究會,「臺灣鐵路旅客列車運行圖 Ver. 5」, 台灣鐵道故事館, 民國九十八年。
- [4] 張有恆,「現代運輸學」, 華泰文化事業股份有限公司, 民國九十四年。
- [5] 黃士弦、吳廷忠等,「97年度自行研究號誌故障, 如何縮短變更閉塞時間, 以減少延誤時分研究」, 中華民國交通部臺灣鐵路管理局運務處, 民國九十八年。
- [6] 黃民仁,「新世紀鐵路工程學——基礎篇」, 文笙書局, 民國九十六年。
- [7] 鍾志成、李治綱等,「軌道容量研究——臺鐵系統容量模式之建構分析(一)」, 中華民國交通部運輸研究所, 研究報告, 民國九十四年。
- [8] 鍾志成、李治綱等,「軌道容量研究——臺鐵系統容量模式之建構分析(二)」, 中華民國交通部運輸研究所, 研究報告, 民國九十五年。
- [9] 鍾志成、李治綱等,「運輸系統容量分析暨應用研究——軌道系統(1/4)」, 中華民國交通部運輸研究所, 研究報告, 民國九十六年。
- [10] 鍾志成、李治綱等,「運輸系統容量分析暨應用研究——軌道系統(2/4)」, 中華民國交通部運輸研究所, 研究報告, 民國九十七年。
- [11] 鍾志成、李治綱等,「運輸系統容量分析暨應用研究——軌道系統(3/4)」, 中華民國交通部運輸研究所, 研究報告, 民國九十八年。
- [12] 鍾志成、李治綱等,「臺鐵車輛排程最適化之研究」, 中華民國交通部運輸研究所, 研究報告, 民國九十五年。
- [13] Baccelli, F. ; Cohen, G. ; Olsder, G. J. ; Quadrat, J.-P., *Synchronization and Linearity: An Algebra for Discrete Event Systems*. London: Wiley, 1992. A new web edition is released at 2001.
- [14] Cormen, T. H. ; Leiserson, C. E. ; Rivest, R. L. ; Stein, C. *Introduction to Algorithms*, second edition. Massachusetts: The MIT Press, 2001.

- [15] De Kort A. F. ; Heidergott B. ; Ayhan H., “A probabilistic (max, +) approach for determining railway infrastructure capacity”. *European Journal of Operational Research* 148, pp. 644–661, 2003.
- [16] Delorme, X. ; Gandibleux, X. ; Rodriguez, J., “Stability evaluation of a railway timetable at station level”, *Information Control Problems In Manufacturing 2006: A Proceedings Volume from the 12th IFAC International Symposium (INCOM'06)*, pp. 379–384, St Etienne: Elsevier Science, 2006.
- [17] Delorme, X. ; Gandibleux, X. ; Rodriguez, J., “Stability evaluation of a railway timetable at station level”, *European Journal of Operational Research* 195, pp. 780–790, 2009.
- [18] Gondran, M. ; Minoux, M., *Graphs, Dioids and Semirings: New Models and Algorithms*. New York: Springer Science+Business Media, 2008.
- [19] Goverde, R. M. P., *Punctuality of Railway Operations and Timetable Stability Analysis*, PhD thesis, TRAIL Thesis Series T2005/10, Delft University of Technology, 2005.
- [20] Goverde, R. M. P., “Railway timetable stability analysis using max-plus system theory”, *Transportation Research Part B* 41, pp. 179–201, 2007.
- [21] Hansen, I. A., Current State-of-the-Art of Railway operations research, 2005.
- [22] Hansen, I. A. ; Pachl, J., *Railway Timetable & Traffic*. Hamburg: Eurailpress, 2008.
- [23] Heidergott, B., Olsder, G. J., Woude, J. W. van der, *Max Plus at Work*. New Jersey: Princeton University Press, 2006.
- [24] Huisman, T. ; Boucherie, R. J., “Running times on railway sections with heterogeneous train traffic”, *Transportation Research Part B* 35, pp. 271–292, 2001.
- [25] Mattsson, L-G., “Railway Capacity and Train Delay Relationships”, *Critical Infrastructure*, pp. 129–150, Berlin Heidelberg: Springer-Verlag, 2007.
- [26] McEneaney, W. M., *Max-Plus Methods for Nonlinear Control and Estimation*. Boston: Birkhäuser, 2006.
- [27] Middelkoop, D. ; Bouwman, M., “Simone: large scale train network simulations”, *Proceedings of the 33rd conference on Winter simulation*, 2001.

- [28] Pachl, J., *Railway Operation and Control*, second edition, Terrace: VTD Rail Publishing, 2002.
- [29] Rudin, W., *Principles of Mathematical Analysis*, third edition, New York: McGraw-Hill, 1976.
- [30] Salido, M. A. ; Barber, F. ; Ingolotti, L., “Robustness in Railway Transportation Scheduling”, *Proceedings of the 7th World Congress on Intelligent Control and Automation*, June 25–27, 2008, Chongqing, China.
- [31] Vromans, M. J. C. M. ; Dekker, R. ; Kroon, L. G., “Reliability and heterogeneity of railway services”, *European Journal of Operational Research* 172, pp. 647–665, 2006.







# 索引

- 7 劃
- 序關係, 29
  - 典型, 29
- 8 劃
- 延誤
  - 次生, 21
  - 原始, 21
- 延誤敏感向量, 14, 35
- 延誤衝擊向量, 14, 35
- 表定等候時間, 21
- 9 劃
- 恢復矩陣, 14
- 相鄰矩陣, 8
- 10 劃
- 時刻表
  - 功能, 1
  - 以 max-plus 代數模化, 25–27
  - 可運行, 32
  - 可實現, 14, 32
  - 可靠度, 23
  - 排班限制, 20–21
  - 週期性
    - 延誤傳遞, 15
    - 模化, 9–10
    - 穩定性分析, 13
  - 穩定性, 1, 13, 22–23
  - 穩定度, 23
- 11 劃
- 閉塞區間, 20
- 12 劃
- 循環恢復時間, 14
- 稀疏矩陣, 37
- 14 劃
- 漸近穩定, 22
- 15 劃
- 餘裕, 21, 22
- 餘裕矩陣, 35, 42
- 17 劃
- 擬反矩陣, 33
- 演算法, 36–38
- 19 劃
- 穩定商數, 36, 44
- 21 劃
- 鐵路系統, 19
- 行車制度, 20
- 鐵路運轉, 19–20
- Jacobi iteration, 36
- max-plus 代數, 6–9
- 矩陣運算, 7–8
- $\mathbb{R}_{\max}^n$ 
  - 序關係, 31
  - 極大元素, 32
  - 極小元素, 32
- timed event graph, 11

## 附件

這是上傳於圖書館之電子檔版本，不含附件光碟，附件請逕向國立交通大學浩然圖書館借用。附件光碟內含論文全文電子檔、穩定性分析程式、程式原始碼、論文與程式之完整授權條款，利用光碟前請務必仔細閱讀使用說明並瞭解授權條款之內容，不得有違反授權條款之行為，否則將依法追究。

