

東 海 大 學

工業工程與經營資訊學系

碩士論文

分散式平行計算應用於以製造廠為中心之  
供應鏈網絡規劃

研 究 生：林詩彥

指 導 教 授：黃欽印 教授

陳武林 教授

中 華 民 國 一〇〇 年 七 月

# **Distributed parallel computing applied to manufactory-centralized supply chain network planning**

By

Shih-Yan Lin

Advisor : Prof. Chin-Yin Huang

Prof. Wu-Lin Chen

A Thesis

Submitted to the Institute of Industrial Engineering and Enterprise

Information at Tunghai University

in Partial Fulfillment of the Requirements

for the Degree of Master of Science

in

Industrial Engineering and Enterprise Information

July 2011

Taichung , Taiwan , Republic of China

# 分散式平行計算應用於以製造廠為中心之供應鏈網絡規劃

研究生：林詩彥

指導教授：黃欽印 博士

陳武林 博士

東海大學工業工程與經營資訊學系

## 摘 要

在現今的全球化競爭環境下，越來越多企業從以往的單廠區生產規劃走向跨國多廠區的生產模式，形成一個供應鏈網絡的供需模式，隨之而來的壓力已經不再只是企業與企業的競爭，而是供應鏈與供應鏈間的競爭，所以，在未來面臨的課題是如何獲取整體供應鏈的最佳利潤。

過去的許多文獻提及，供應鏈規劃為一 NP-Complete 的問題，針對此問題，過去有學者提出利用模擬法以及啟發式演算法來解決之，這些研究往往可以節省許多運算時間，但所求結果皆非最佳解。本研究以製造廠為中心，考量不同特性如產能、物料限制、運輸等，建立適合的數學模型，並且利用 LINGO10.0 數理規劃軟體進行最佳結果規劃。

在規劃 NP-Complete 問題時，時常會花費許多的時間，本研究利用分散式平行計算，將建構的數學模型分割成許多組合，藉由 Java RMI(Java Remote Method Invocation) 將切割後的組合進行平行計算，以達降低運算時間之目的。實驗結果顯示，與運用單一處理器進行運算比較，透過分散式平行計算可以大幅降低運算時間。

**關鍵字詞：**供應鏈網絡、多廠區生產規劃、NP-Complete、平行計算、Java RMI

# **Distributed parallel computing applied to manufactory-centralized supply chain network planning**

Student: Shih-Yan Lin

Advisor: Prof. Chin-Yin Huang

Prof. Wu-Lin Chen

Department of Industrial Engineering and Enterprise Information  
Tunghai University

## **ABSTRACT**

Under the global competition, more and more enterprises change their single-plant production planning to multinational-plant production system. As the supply and demand model of supply chain network, the following pressure is no longer the competition between enterprises but supply chains, so the future issue is how to obtain the best profits in the overall supply chain.

A lot of past literature mentioned that supply chain planning is an NP-Complete problem. For the problem, the past researcher proposed some methods to solve it such as simulation or developing a heuristic algorithm. Those researches often save much computation time, but the result they found is not optimal. This research is manufactory-centralized, which considers different features such as manufactory capacity, material limit, transportation and so on. Using those features construct appropriate mathematical model and planning the optimal solution by using mathematical planning software.

It often spends a lot of time on planning NP-Complete problem. This research uses distributed parallel computing. It divided the mathematical model into a lot of combinations. Then by using Java Remote Method Invocation technique pass the divided combinations to the parallel computers. Parallel computing will achieve the goal 'saving a lot of time.' Experimental results demonstrates that it reduce a lot of solving time by parallel computing significantly. The results are compared with using single computer to solve the mathematical model.

**Keywords: supply chain network; multi-plant production planning; NP-Complete;  
parallel computing; JavaRMI**

## 誌謝

在東海大學已經是第六個年頭，在論文完成的同時，也宣告著將與這再熟悉不過的地方告別，也許，天下沒有不散的筵席，再怎麼熟悉的環境，終須一別，隨之而來的是，即將步入社會的壓力，然而，經由大學四年以及研究所兩年的薰陶，銘記師長的教誨，使我可以勇於面對未來未知的壓力與恐懼。

在碩士班兩年中，首先要感謝黃欽印老師，老師在這兩年中除了在論文與課業方面給予諸多指點外，對於將來就業、做人處事的態度以及生涯規劃都給予寶貴經驗分享，提醒我們要以過去學長姐的經歷作為借鏡，避免重蹈覆轍。還要感謝所有任課老師所給予的指導，因為各位老師的指教，詩彥得以充實自己，使自己學習更多技能與知識。

本論文得以完成，最主要要感謝指導老師黃欽印老師以及陳武林老師，每週兩位老師都給與寶貴的意見，漸漸引導我完成這份論文。從完成論文的過程中，學習發現問題與解決問題的能力，並且透過每週簡報，加強自我表達能力以及培養自己的口條。也感謝口試委員賴奕銓老師以及鄭辰仰老師於口試時給予之寶貴意見，使論文內容更加完備。

兩年的生活中，感謝振誠、弼仁、斯暢、靖雅學長姐的領導與指教，尤其是斯暢學長，給予相當多的指導。亦要感謝玠孝、容慈、鈞翔等研究室夥伴們，在研究路上遇到許多問題，謝謝你們給予各方面的意見。還有 EDA 研究室的好朋友們，靖儀、柏祥、秉群、怡嬪，謝謝你們，在這兩年來有很多很多共同的回憶，一起走過很多地方，很慶幸這兩年有你們的相伴，讓兩年的研究所生涯增添許多色彩。以及感謝學弟妹芸甄、貞翔、思翰及康硯，感謝你們給與諸多的協助，在我埋首於寫論文時無後顧之憂。

最後，最要感謝的就是我的家人，從小到大，都是你們一路支持著我，給予絕對的支持與鼓勵，也時時刻刻叮嚀著我在忙碌之餘，要照顧好自己的身體，謝謝你們的關心與陪伴，讓我這一路走來得以順遂，沒有你們的相挺，就沒有現在的我。

感謝一路上給予幫助的師長與親朋好友們，因為有你們，詩彥得以完成此論文，將此份榮耀，與各位分享，也期待在未來的路上，詩彥能奉獻微薄心力回饋各位。

林詩彥 謹誌於

東海大學工業工程與經營資訊學系

虛擬企業與資料探勘研究室

中華民國一百年七月

# 目錄

摘要 .....	I
ABSTRACT .....	II
誌謝 .....	III
目錄 .....	IV
圖目錄 .....	VI
表目錄 .....	VIII
第一章 緒論 .....	1
1.1 研究背景 .....	1
1.2 研究動機 .....	4
1.3 研究目的與步驟 .....	5
第二章 文獻探討 .....	7
2.1 供應鏈網絡 .....	7
2.2 供應鏈網絡生產規劃 .....	9
2.3 平行計算 .....	15
2.3.1 平行計算架構 .....	15
2.3.2 分割與平行化分析 .....	17
第三章 供應鏈網絡生產規劃模式 .....	20
3.1 供應鏈網絡結構 .....	20
3.1.1 第一階層：供應商供應原物料給製造廠 .....	20
3.1.2 第二階層：製造廠生產產品，並將成品運送至配銷中心 .....	21
3.1.3 第三階層：配銷中心分裝出貨給最終顧客 .....	22
3.2 供應鏈網絡之數學模型 .....	23
3.2.1 假設條件： .....	23
3.2.2 下標說明 .....	24
3.2.3 本研究輸入參數及其對應之符號說明： .....	24
3.2.4 決策變數 .....	26
3.2.5 圖形化描述供應鏈網絡 .....	27
3.2.6 數學模型目標式 .....	28
3.2.7 數學模型限制式 .....	33
3.3 分散式平行計算架構 .....	40

3.3.1. 循序化平行計算 .....	41
3.3.2. LINGO 上界與下界 .....	49
<b>第四章 實驗與分析 .....</b>	<b>62</b>
4.1 實驗環境與參數設定 .....	62
4.1.1 實驗環境 .....	62
4.1.2 實驗說明 .....	63
4.1.3 參數設定 .....	65
4.2 實驗設計及分析 .....	67
4.2.1 小規模全域最佳解 .....	67
4.2.2 小規模平行計算規劃結果 .....	74
4.2.3 中規模全域最佳解 .....	77
4.2.4 中規模平行計算規劃結果 .....	83
4.2.5 大規模全域最佳解與平行計算規劃結果 .....	89
4.3 結果討論 .....	91
<b>第五章 結論與未來發展方向 .....</b>	<b>92</b>
5.1 結論 .....	92
5.2 未來發展方向 .....	92
<b>參考文獻 .....</b>	<b>94</b>

## 圖目錄

圖 1.1、多階多廠區供應鏈網絡示意圖 .....	3
圖 1.2、研究步驟 .....	6
圖 2.1、供應鏈網絡架構 .....	7
圖 2.2、一般性協同規劃與多廠區協同規劃示意圖 .....	9
圖 2.3、供應鏈模式的三種基本類型 .....	10
圖 2.4、單一產品來源、多產品供應鏈網絡示意圖 .....	12
圖 2.5、產品、多廠區、多時期供應鏈網絡圖 .....	13
圖 2.6、共享式記憶體多處理器系統結構圖 .....	16
圖 2.7、分散式記憶體多處理器系統結構圖 .....	17
圖 2.8、平行系統設計 .....	18
圖 2.9、RMI 運作架構 .....	19
圖 3.1、多階多廠區供應鏈網絡示意圖 .....	20
圖 3.2、研究之記憶體模組 BOM 表 .....	21
圖 3.3、圖形化符號對應示意圖 .....	28
圖 3.4、製造廠存貨週期示意圖 .....	30
圖 3.5、製造廠生產週期示意圖 .....	30
圖 3.6、配銷中心節點分裝週期示意圖 .....	31
圖 3.7、平行計算流程 .....	41
圖 3.8、資料分割組合 .....	42
圖 3.9、RMI 架構圖 .....	43
圖 3.10、伺服端計算流程 .....	45
圖 3.11、計算節點計算流程 .....	47
圖 3.12、LINGO 計算上界與下界變化 .....	49
圖 3.13、以給定或截至目前為止所求得之最佳目標函數值控制上界 .....	50
圖 3.14、伺服端計算流程(使用最佳規劃結果控制上界) .....	52
圖 3.15、計算節點計算流程(使用最佳規劃結果控制上界) .....	55



圖 3.16、平行計算後得到較佳之目標函數值.....	56
圖 3.17、伺服端計算流程(取得更佳目標函數值取代原始控制上界值) .....	57
圖 3.18、計算結點計算流程(取得更佳目標函數值取代原始控制上界值).....	60
圖 4.1、多階多廠區供應鏈網絡示意圖 .....	63
圖 4.2、小規模編號一實驗第一次、第二次實驗全域最佳解規劃運輸分配 .....	68
圖 4.3、小規模編號一實驗第三次實驗全域最佳解規劃運輸分配 .....	69
圖 4.4、小規模編號二實驗第一次實驗全域最佳解規劃運輸分配 .....	70
圖 4.5、小規模編號二實驗第二次實驗全域最佳解規劃運輸分配 .....	71
圖 4.6、小規模編號二實驗第三次實驗全域最佳解規劃運輸分配 .....	71
圖 4.7、中規模 2 倍製造廠產能，60%需求量之 LINGO 全域規劃結果 .....	78
圖 4.8、中規模 LINGO 全域規劃上界與下界收斂狀況 .....	78
圖 4.9、中規模 1 倍製造廠產能，60%需求量之 LINGO 全域規劃結果 .....	80
圖 4.10、中規模 2 倍製造廠產能，80%需求量之 LINGO 全域規劃結果 .....	81
圖 4.11、時間-淨利成長幅度圖 .....	82
圖 4.12、中規模 60%需求量實驗規劃時間比較圖 .....	87
圖 4.13、中規模 80%需求量實驗規劃時間比較圖 .....	87
圖 4.14、中規模 2 倍製造廠產能規劃時間比較圖 .....	88
圖 4.15、中規模 1 倍製造廠產能規劃時間比較圖 .....	88
圖 4.16、大規模單一處理器 8 小時處理結果.....	90
圖 4.17、大規模單一處理器 10 小時處理結果.....	90

## 表目錄

表 1.1、記憶體模組廠 2009 年記憶體模組營收排名.....	2
表 2.1、供應鏈網絡生產規劃相關研究整理.....	14
表 3.1、本研究考量生產規劃之特性與限制.....	22
表 3.1、本研究考量生產規劃之特性與限制(續).....	23
表 3.2、分散式平行計算結果.....	48
表 4.1、環境規模 .....	64
表 4.2、實驗因子 .....	64
表 4.3、參數值及其代表意義.....	65
表 4.3、參數值及其代表意義 (續) .....	66
表 4.4、小規模組合製造廠及顧客節點對應數量表 .....	67
表 4.5、小規模編號一實驗三次實驗目標函數規劃結果 (續) .....	70
表 4.6、小規模編號一實驗三次實驗規劃時間.....	70
表 4.7、小規模編號二實驗三次目標函數規劃結果 .....	72
表 4.8、小規模編號二實驗三次實驗規劃時間.....	72
表 4.9、小規模編號三實驗三次目標函數規劃結果 .....	73
表 4.10、小規模編號三實驗三次實驗規劃時間 .....	73
表 4.11、小規模編號四實驗三次實驗規劃時間 .....	73
表 4.12、小規模各水準單一處理器全域最佳規劃時間.....	74
表 4.13、小規模各水準對應選取家數表*註二.....	75
表 4.14、小規模 2 倍製造廠產能、60%需求量單一處理器與多處理器比較 .....	75
表 4.15、小規模 1 倍製造廠產能、60%需求量單一處理器與多處理器比較 .....	76
表 4.16、小規模 2 倍製造廠產能、80%需求量單一處理器與多處理器比較 .....	76
表 4.17、小規模 1 倍製造廠產能、80%需求量單一處理器與多處理器比較 .....	77
表 4.18、中規模各水準對應產能與總需求量表 .....	77
表 4.19、中規模 2 倍製造廠產能，60%需求量規劃時間與淨利增加變化 .....	79
表 4.20、中規模 1 倍製造廠產能，60%需求量規劃時間與淨利增加變化 .....	80

表 4.21、中規模 2 倍製造廠產能，80%需求量規劃時間與淨利增加變化 .....	81
表 4.22、中規模 1 倍製造廠產能，80%需求量規劃時間與淨利增加變化 .....	82
表 4.23、中規模各水準對應選取家數表*註四 .....	83
表 4.24、中規模 2 倍製造廠產能，60%需求量平行計算結果.....	84
表 4.25、中規模 1 倍製造廠產能，60%需求量平行計算結果.....	85
表 4.26、中規模 2 倍製造廠產能，80%需求量平行計算結果.....	85
表 4.27、中規模 1 倍製造廠產能，80%需求量平行計算結果.....	86
表 4.28、中規模 8 與 16 台處理器實際規劃時間 .....	89
表 4.29、大規模 1 倍製造廠產能，60%需求量規劃結果 .....	89

# 第一章 緒論

## 1.1 研究背景

在現階段全球化市場競爭的環境下，許多企業的生產模式從以往的單廠區生產規劃走向跨國多廠區的生產模式，企業所面臨的挑戰愈來愈艱鉅，要考量的因素越來越多，而且往往各個因素間存在著相依關係，常常造成「牽一髮而動全身」的狀況發生，所以該如何有效運用各廠之生產資源，使得獲得的利潤最大化，是企業必定會面臨到的問題。

本篇論文以記憶體模組產業之產品為例，探討以製造廠為中心的生產規劃模式。記憶體模組產業（Memory Module）位於供應鏈中游的位置，屬於製造廠之角色，向上游供應商購買關鍵原物料動態隨機存取記憶體（Dynamic Random Access Memory, DRAM）顆粒，以表面黏著技術（Surface-mount technology, SMT），將顆粒依顧客需求鑲嵌至印刷電路板（PCB）上。近年來，隨著消費性電子產品和智慧手機的需求急遽增加，以及其（消費性電子產品和智慧手機）產品生命週期往往如曇花一現般短暫，使得記憶體模組產品日漸走向多元化，技術也持續進步。

台灣地區為記憶體模組產業的生產重鎮，根據集邦科技 (DRAMeXchange) 資料顯示，2009 年營收前 15 大自有品牌記憶體模組廠商中，台灣廠商佔了 7 家；2009 年全球記憶體模組市場總銷售金額達 68 億 4 千萬美金，較 2008 年成長 14.89%；台灣記憶體模組廠商總營收自 2008 年的 16 億 1 千萬美金提升至 19 億 2 佰萬美金，成長幅度 18.14%，而在 2009 年，台灣記憶體模組廠商總營收佔全球記憶體模組市場總銷售金額的 27.8%；在全球市場佔有率方面，台灣記憶體模組廠商市場佔有率為 27.81%，資料如表 1.1，顯示台灣在記憶體模組產業仍佔有一席之地。

表 1.1、記憶體模組廠 2009 年記憶體模組營收排名

公司	區域	排名		營收 (美金百萬)			市占率 (%)	
		2009	2008	2009	2008	YoY %	2009	2008
Kingston Technology	US	1	1	2,460	1,470	67.35%	35.96%	24.69%
A-Data Technology	Taiwan	2	3	532	453	17.44%	7.78%	7.61%
Ramaxel	China	3	4	503	420	19.76%	7.35%	7.05%
Crucial Technology	US	4	11	430	160	168.75%	6.29%	2.69%
Transcend Information	Taiwan	5	5	371	360	3.06%	5.42%	6.05%
Smart Modular Technologies	US	6	2	366	573	-36.13%	5.35%	9.62%
Apacer Technology	Taiwan	7	8	302	243	24.28%	4.42%	4.08%
Corsair Memory	US	8	7	235	256	-8.20%	3.44%	4.30%
MA Labs	US	9	6	225	315	-28.57%	3.29%	5.29%
PQI	Taiwan	10	12	215	154	39.61%	3.14%	2.59%
Unifosa	Taiwan	11	10	209	161	29.81%	3.06%	2.70%
Team Group	Taiwan	12	14	138	115	20.00%	2.02%	1.93%
Kingmax Semiconductor	Taiwan	13	13	135	124	8.87%	1.97%	2.08%
Buffalo	Japan	14	9	111	174	-36.21%	1.62%	2.92%
Stronitum	Singapore	15	15	105	95	10.53%	1.54%	1.60%
Others				503	880	-42.85%	7.36%	14.79%
<b>Total DRAM Revenue</b>				<b>6,840</b>	<b>5,953</b>	<b>14.89%</b>	<b>100.00%</b>	<b>100.00%</b>

資料來源：集邦科技(DRAMEXchange)

對於從供應商取得原物料，由製造廠進行生產，最後配銷中心運送貨物交給顧客，以供應鏈結構來看是屬於一個「多階層」的供應鏈結構；而在全球化經營的結構下，往往供應商、製造廠、配銷中心、顧客等各階層都不僅只有單一的選擇性，形成了「多廠區」的供應鏈結構，其多階多廠區供應鏈網絡示意圖如圖 1.1

除上述之「多階多廠區」特性，製造廠之獲利受到原物料波動和存貨多寡影響甚深，當關鍵料充足時，產品會產生供過於求的狀況，廠商面臨庫存壓力，並且可能因為要降低虧損將產品進行降價，以求盡速將庫存脫手；當關鍵料貨源取得不易，造成關鍵料價格上漲也因關鍵料取得不易，間接使得記憶體模組產品有供不應求的狀況。如近期，因受到日本東北地區地

震影響，關鍵料供應商無法短期間內恢復正常運作，造成原物料短缺、原物料顆粒的價格上升，製造廠生產供給狀況出現狀況，直接地影響到企業的獲利狀況。

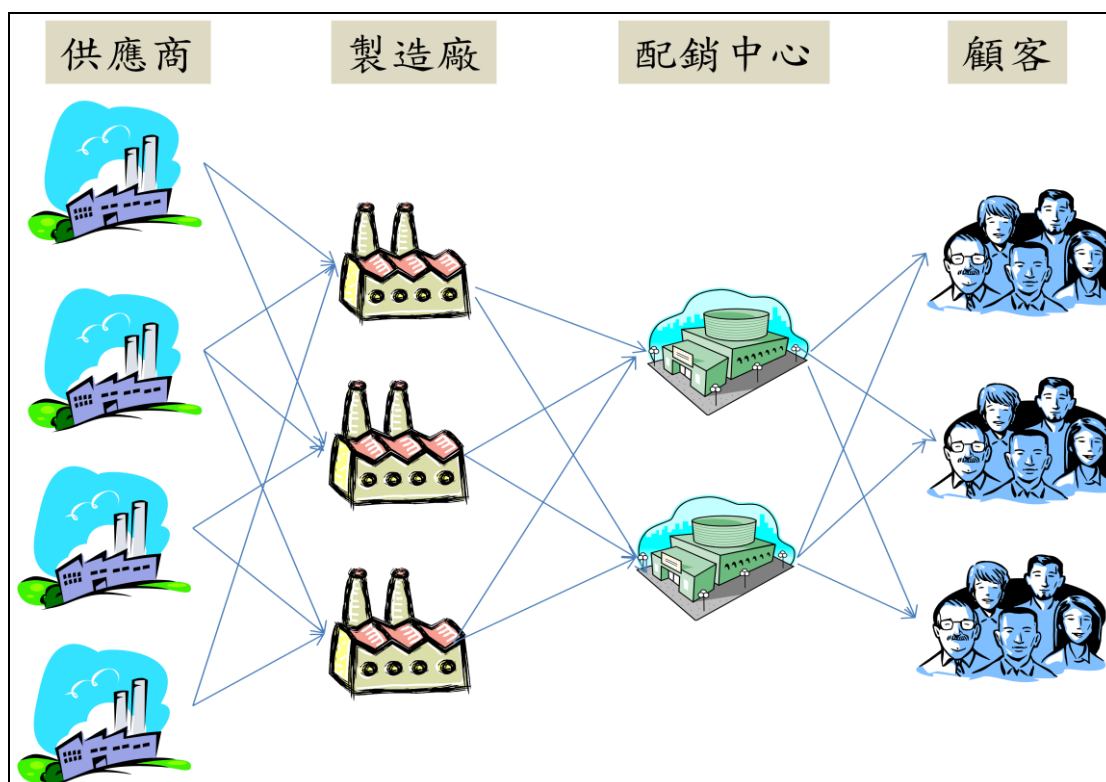


圖 1.1、多階多廠區供應鏈網絡示意圖

由上述可知，對製造廠而言，原物料的用量是一個重要的因素，在此，物料清單 (Bill of material, BOM) 就扮演重要的角色，BOM 表展開後就可以清楚的知道須要向供應商購買何種原物料及數量，針對關鍵以及非關鍵料分項列管，斟酌採購數量以及庫存數量，有助於掌控成本。

為了有效掌控成本，善用各廠資源以追求利潤最大化，製造廠對顧客的需求，除考量到關鍵料取得與否、BOM 表、運輸以及生產的前置時間所造成的庫存狀況外，尚須考量到運輸批量及運輸批量成本，再進行生產規劃。原因在於在有限的規劃時間內，運輸批量的多寡會影響到產品是否能夠準時達交、運輸次數及運輸成本。故製造廠在進行生產規劃，找尋最佳解時，需要考量到下列幾項要點：

1. 多階多廠區供應鏈網絡的生產環境。
2. 產品 BOM 表結構。

3. 供應商各項原物料供給數量限制（此篇論文稱其為供應商產能限制）。
4. 各廠區的產能限制（包含製造廠以及配銷中心）。
5. 運輸及生產所需時間。
6. 運輸批量及運輸批量成本。

## 1.2 研究動機

過去有大量針對「多階多廠區」的存貨管理、訂單分配等生產規劃問題的相關文獻，其相關議題所使用的解決方式多以數學模式、模擬方法以及啟發式演算法來解決相關的問題，而這些方式可能會存在下列問題：

1. 使用數學模式來找尋問題的最佳解，常見到的方法是使用線性規劃及整數規劃等方法，然而隨著供應鏈網絡的規模大小增加（廠區家數、產品種類、需求數量等），規劃的時間呈現指數成長，所以製造廠在規劃時可能花費大量的時間仍無法規劃出結果，在實務上的應用並不方便。
2. 使用模擬方式來模擬真實世界會面臨到的狀況，藉由設定不同的參數評估不同方案的結果，以模擬方式進行規劃，其時間較數學方式節省許多，但模擬方法所提供的結果可能是「局部最佳化」，而非「整體最佳化」，而且規劃人員可能無法從中得知規劃過程，作為評估結果的考量。
3. 使用啟發式演算法進行生產規劃，常見的啟發式演算法有模擬退火法（Simulated Annealing）、塔布搜尋法(Tabu Search)、基因演算法(Genetic Algorithms)等，利用一些簡單邏輯判斷法則，反覆進行演算來求出結果，優點在於可以較數學模式減少許多的運算時間；而其缺點在於所得結果不一定是最佳解，而且每一種啟發式演算法所適用的環境不盡相同，所以儘管解決數學方式所擁有的費時問題，但也如同模擬方法一樣，面臨到「局部最佳化」的問題。

Wu(Wu, 2004)證明在多品項、多廠區及多時期的供應鏈網絡生產環境下，其生產規劃將會是一個 NP-Complete 的問題，即便是在單一品項、多廠區、多時期、無限產能且無運輸成本的狀況下，其生產規劃仍為一個 NP-Complete 的問題。綜合上述，過去針對 NP-Complete 的供應鏈網絡生產規劃問題也有許多學者提出求解方法，Byrne and Bakir (Byrne & Bakir, 1999) 利用數學規劃及系統模擬，求解多期多產品生產規劃問題；Leung et



al(Leung, Wu, & Lai, 2003)考量到產能、人力等因素，探討多地區製造廠的生產規劃問題。但使用分散式平行計算降低數學模式運算時間達最佳規劃結果之相關研究相對稀少。因此，在多階多廠區的供應鏈網絡生產特性下，以製造廠為中心，考量相關的生產特性，建構一套平行計算系統，在規劃週期內，利用各階層各廠區有限之資源，求得最佳之生產規劃結果，作為協助規劃人員做決定的參考依據。

### 1.3 研究目的與步驟

誠如研究背景與動機所述，在多階多廠區的生產環境下，過去文獻對於分散式平行系統應用於供應鏈生產規劃甚少著墨，且在考量追求最佳解的前提下，使用數學模式進行規劃而造成 NP-Complete 問題，造成計算時間過於冗長。本研究將以製造廠為中心，考量製造廠進行生產規劃限制因素，例如 BOM 表、運輸限制、生產時間、產能限制等，藉由記憶體模組為生產產品，提出符合之供應鏈網絡生產規劃數學模式，接著將其分散至多台電腦進行平行運算，以達到下列目的：

1. 降低規劃時間：利用數學模式進行規劃，隨規模擴大產生 NP-Complete 問題，透過平行計算達到大幅降低規劃時間。
2. 求得最佳規劃結果：透過數學模型與數理規劃軟體，求得最佳規劃結果供規劃人員參考。

為了達到上述目的，本研究根據圖 1.2 的研究步驟進行之。

1. 從過去的文獻探討多階多廠的供應鏈網絡生產規劃的特性，瞭解其在規劃時考量到哪些特性與限制，運用何種技術解決問題以及其成效如何。
2. 從相關文獻中瞭解分散式系統以及平行運算的背景以及其運作的模式為何，並探討過去類似之相關應用。
3. 建立出符合多階多廠的供應鏈網絡數學模型，並詳細說明各符號、限制式等意義；利用分散式物件技術以及數理規劃工具將數學模型進行平行運算。
4. 進行實驗設計，說明實驗方式、環境建構、實驗因子等，比較本研究與未進行分散式平行計算之最佳規劃結果之差異，並比較因處理器個數不同所造成時間的差異，藉由在最佳解規劃結果差異不大的狀況下，證明



本研究之運算數學模式時間確實縮減許多。

- 藉由實驗分析的結果，提出結論、建議以及未來可以延續討論的議題。

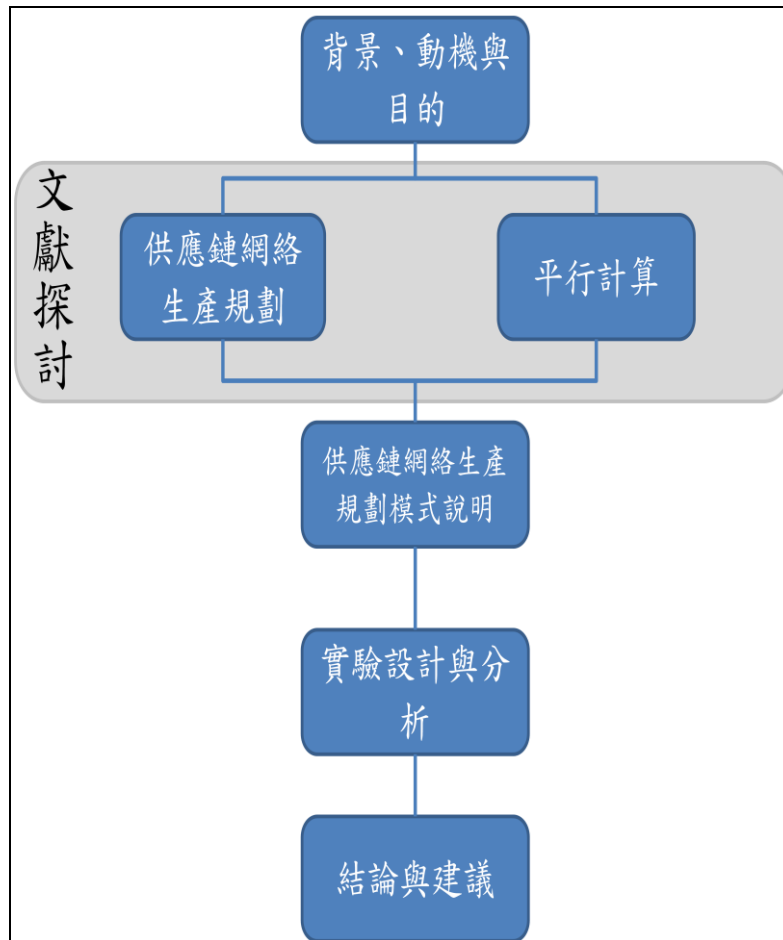


圖 1.2、研究步驟

## 第二章 文獻探討

本研究在探討供應鏈網絡生產規劃的平行計算，故本章將分成三個部分，分別針對供應鏈網絡定義、供應鏈網絡生產規劃以及平行計算三個部分進行探討，了解相關領域的發展近況，以利研究順利進行。

### 2.1 供應鏈網絡

Lambert et al.(Lambert, Cooper, & pagh, 1998) 提出供應鏈管理模式主要由供應鏈網絡結構、供應鏈商業程式以及供應鏈管理元素等三個子系統組成。其中表示供應鏈網絡部分並非一對一與企業對企業的商業鏈，而是多層的商业關係。所以供應鏈是一個複雜的網絡架構，各供應商與製造廠並非單純的一對一關係，而是像樹狀圖的分歧關係，如同圖 2.1 第一階供應商所需物料是向第二階供應商購買而來，而第二階供應商所需物料是向第三階供應商購買直到最原始供應商，且每階層供應商家數並非唯一；反之，第一階購買顧客也會賣給第二階顧客，直到最終消費者。

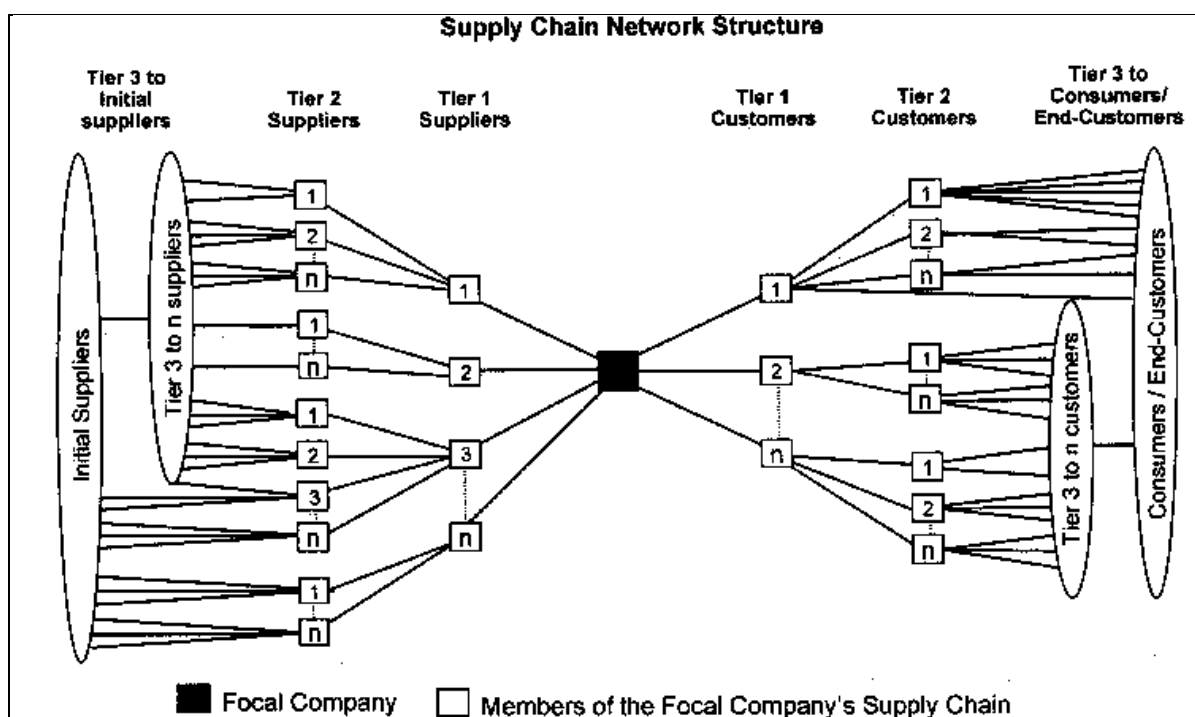


圖 2.1、供應鏈網絡架構

Simchi-Levi et al (Simchi-Levi, Kaminsky, & Simchi-levi, 2001)指出供應鏈也可以稱為物流網路，由供應商、製造廠、倉庫、配銷中心和零售商所構成，可以讓原物料、在製品存貨、成品在上述設施中流動。作者進一步

定義，供應鏈管理是利用一連串有效率的方法，來整合供應商、製造廠、配銷中心與零售商，使得產品可以在交期範圍內送正確的數量到正確的地點，滿足顧客需求，以達成整體系統利潤最大化之目標。

Shapiro (Shapiro, 2001)指出供應鏈架構有三個階層，包括供應商、製造商或工廠以及配送網路。所謂的供應商通常是指直接將原物料提供給負責生產與組裝工廠者；製造商或工廠主要是透過特定的生產程序，將原物料轉換成最終產品或服務者；配送網路包含配銷中心、零售商以及最終顧客等，此網路之目的在於將前述製造商或工廠生產之產品或服務運送給最終顧客。

而供應鏈架構中各階層之間的關係，則必須考慮各個企業所處的競爭位置為何，若整個供應鏈掌握在某個企業或財團，而供應鏈上的其他成員只能扮演追隨者的角色，這樣的供應鏈結構可是為一種集中式的供應鏈。然而現今的供應鏈生產環境逐漸走向全球化，由甲地獲取原物料，乙地設計製造，丙地配銷出貨，供應鏈的各個個體都存在著相依關係，彼此之間的關係密切，彼此間透過實體物料流、金流、資訊流等進行合作關係，此種供應鏈體系可以視為分散式的供應鏈。

供應鏈管理的目標是使整個供應鏈網絡具有效率以及利潤最大化，但在供應鏈中，不同成員經常會有不同或是相互衝突的目標，而在全球市場或策略產生變化時，企業不可只依附在相同的供應鏈網絡中，必須調整其供應鏈，以確保競爭力。關於供應鏈規劃問題，Bhatnagar and Chandra (Bhatnagar & Chandra, 1993)將供應鏈規劃區分成兩個層次，分成一般協同規劃與多廠區協同規劃，如圖 2.2 所示。一般協同規劃指的是同時考慮企業在不同功能的整合，如生產/配送(Production/Distribution)、庫存/配送(Inventory/Distribution)等，不同功能同時進行規劃的範圍包括前段採購、生產裝配到後段配送；而多廠區協同規劃則是針對企業內，同一功能但不同階層間的決策連結，好比一家企業有數家製造廠，製造廠間在生產的這個功能存在著順序性，製造廠完成半成品後再由其他製造廠繼續加工，做成完成品，對於製造廠間的生產規劃與決策必須要有良好的整合。本研究是以製造廠的角度最為考量進行供應鏈網絡生產規劃，一方面有考量自上游供應商至下游配銷中心的規劃，另一方面也有考量到多廠區的特性，屬於

兩者之綜合型協同規劃。

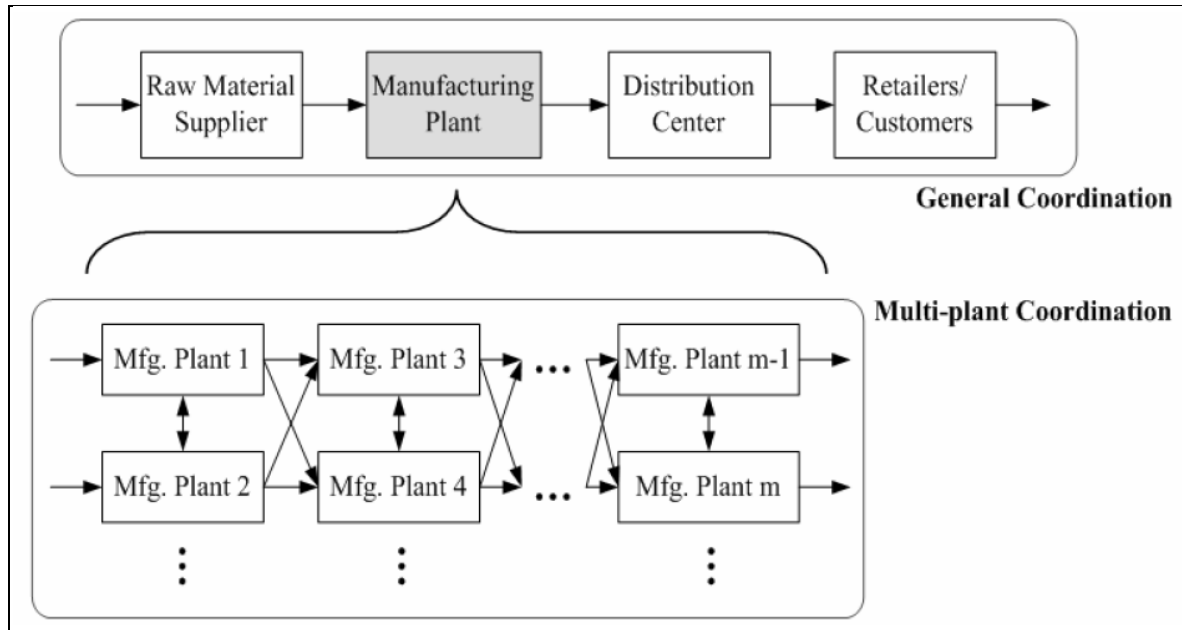


圖 2.2、一般性協同規劃與多廠區協同規劃示意圖

Guinet (Guinet, 2001)在多廠區規劃的問題上，提出以兩階層生產的方法，將規劃分成多廠規劃與單廠排程問題。前者問題是考量到實際的限制狀況下，如何將所有訂單安排到各廠進行生產；後者是當前者將訂單處理到各廠後，在單廠的環境下進行生產排程，安排生產的時辰與數量。本研究僅考量前者，將訂單數量適當分配到各廠，而各廠內在接到訂單後如何進行排程作業並不加以討論。

Sambasivan and Schmidt (Sambasivan & Schmidt, 2002)提出多廠區規劃問題所規劃的主要內容與產出為(1)每個廠區所要生產的產品種類與數量，(2)每個廠區在各個規劃期末所需持有的存貨數量，(3)廠區與廠區之間的轉換數量。

在了解供應鏈網絡的說明後，接下來將針對過去文獻對此領域的研究來做探討，了解各研究所考量的限制條件，且用何種方法解決供應鏈網絡相關問題。

## 2.2 供應鏈網絡生產規劃

近年來，有許多研究使用不同的方法來解決供應鏈網絡生產規劃問題，較為常見的就是利用數理規劃、模擬、啟發式演算法，其他亦有利用代理人、模糊理論等方式解決相關問題。Taylor (Taylor, 2003)提出建構供應鏈

模式的三種基本類型：概念模式（Conceptual models）、數學模式（Mathematical models）與模擬模式（Simulation models）。概念模式是以圖形化的方式表達供應鏈網絡架構；數學模式以公式以及限制條件符號描繪供應鏈網絡架構，在一些假設前提下藉由公式求解結果；模擬模式使用模擬軟體建立物件來表達各企業體的元件（供應商、製造廠、配銷中心等），透過模擬軟體了解各物件之間的互動所發生情形來尋求可能的結果，三種類型比較如圖 2.3：

	1.觀念模式	2.數學模式	3.模擬模式
供應鏈代表方式	圖形與描述	公式與數學系統	物件與互動關係
求解方法	口語推理與論證	數學求解與電腦運算	蒙地卡羅實驗
最適使用時機	共識建立	尋求最佳績效	實用性評估

圖 2.3、供應鏈模式的三種基本類型

Thierry et al.(Thierry et al, 1995)將用於限制滿足問題（Constraint Satisfaction Problem）方法應用於多廠規劃線性規劃模式求解。並考慮到換模換線時需要較長設置時間的特性。限制滿足問題的求解法是藉由限制來一步步縮小求解空間，等到無法再縮小時，就利用各種啟發式方法來快速找到好的解。此研究有考慮到當某些生產單位在換模換線時需要較長設置時間的特性，所以各生產單位在規劃時的時間區間大小不同。

Arntzen et al.(Arntzen et al, 1995) 提出利用混合整數線性規劃法建構全球供應鏈模式。此模式為多產品、多階層級及多時期，整合製造廠與配銷中心之規劃，目標在求解生產、運輸與存貨等最小成本。

Sauer et al.(J. Sauer et al, 1998)使用模糊理論的概念來處理一般會自行假設固定值或是一些忽略不記的不確定因素如訂單數量及運輸時間等，使得整體排程規劃更具彈性。Sauer and Appelrath,(Sauer & Appelrath, 2000)更進一步將各廠之間的運輸問題納入多廠規劃的求解，將運輸問題轉換成類

似生產排程問題，之後再和其餘各廠一起規劃求解。

Guinet (Guinet, 2001)指出多廠規劃問題可以被建構成受限於變動和固定成本的流動問題(flow problem)。而其在考量產能限制和機台設置特性，以及生產、設置和運輸等成本最小化的情形下，提出了一個「主要-對偶線性規劃法」來求解多廠規劃問題。

郭瑞祥 等(郭瑞祥等, 2004)針對供應鏈多廠區生產規劃，運用混合型整數線性規劃，建構「單階決策模式」，而為了降低問題的複雜度及運算時間，乃進而建構「二階決策模式」。在單階決策模式中，考量製造廠產能與物料限制，追求成本極小化，但其運算耗時，不適合實務應用；二階決策模式則是將單階決策模式中所考量的成本與限制分成兩個階段考量：第一階段為「訂單分配階段」；第二階段為「工廠規劃階段」，此模式降低了單階段決策模式的運算時間而求得接近最佳的生產規劃結果。

Shen et al.(Shen et al, 2003)利用網際網路為基礎的網路協同代理人系統，運用多廠區的生產環境，解決生產規劃問題。 Chan and Chung (Chan & Chung, 2004)研究在訂單分配的需求供應鏈架構下發展出一個多準則基因演算程序來求解訂單分配的問題。而此需求供應鏈架構只有需求端和生產端兩階層，並以生產、運輸成本最小且平衡各製造廠的產能為目標；Chan et al. (Chan et al, 2005)進一步考量作業成本、服務水準及資源使用率，採用AHP為準則權重和基因演算法來解決生產及配銷問題。就上述文獻而言，使用不同方式來解決供應鏈之問題，但解決之問題都屬於單階層多廠區的規劃問題，並未考量到供應鏈網絡「多階層」之概念。

以下文獻將探討考量多階多廠生產環境之供應鏈網絡生產規劃問題。Lendermann et al.(Lendermann et al , 2001) 提出以模擬的方式，建立多階多廠的供應網絡架構，考量多種品項、物料替代關係及製造廠間之物料調撥，然而，此研究並未考量各製造廠之產能限制。Jayaraman and Pirkul (Jayaraman & Pirkul, 2001)探討多階層供應鏈生產模式，並考量工廠及零售商的產能限制提出混合整數線性規劃及運用拉式鬆弛法(Lagrangian Relaxation)為基礎的啟發式演算法來解決供應鏈多廠生產的問題。

Yeh (Yeh, 2005)提出了混合啟發式演算法 (hybrid heuristic algorithm, HAA) 結合貪婪策略(Greedy method, GM)、線性規劃(Linear Programming,

LP)區域搜尋法(Local Search method, LSMs)求解，並且修正過去 Syarif et al. (2002)錯誤的地方再進行多階多廠區生產環境訂單分配規劃，但並未考量時間因素，單純以實體物料流量來做規劃。Yeh (Yeh, 2006)再次提出以結合基因演算法、貪婪啟發式和區域搜尋法的瀾集演算法(Memetic algorithm, MA)，解決多階多廠生產環境訂單分配問題，而本文獻依舊沒有考量到時間因素，如規劃週期、作業前置時間、運輸時間等。

Altıparmak et al.(Altıparmak et al, 2006)利用基因演算法求解供應鏈網絡多目標函數之問題，在已知顧客、供應商家數、顧客需求數和供應商、製造廠、配銷中心產能，且顧客僅能從單一配銷中心獲得成品之前提下，為滿足各製造廠及配銷中心產能以及顧客需求，利用基因演算法決定製造廠及配銷中心所需家數，並求解最小成本以及各製造廠及配銷中心的產能使用率，圖 2.4 為單一產品來源、多產品，三階層的供應鏈網絡示意圖。

Altıparmak et al.(Altıparmak et al, 2009)利用上述相同問題，提出穩態基因演算法 (Steady-state Genetic Algorithm)，此方法將網絡分成三個階層如圖 2.4 所示，前兩階層使用 Priority-based 進行編碼，最後一階使用整數編碼進行啟發式演算法求解，求解結果將與使用 CPLEX 數理規劃軟體、模擬退火法(Simulated Annealing)、混合式基因演算法(Hybrid Genetic Algorithm)等方式求解結果比較，結果顯示穩態基因演算法所求的解較其他啟發式演算法好，而與數理規劃軟體比較則是節省許多運算時間，而本方法之缺點即在於所求解並非最佳解。

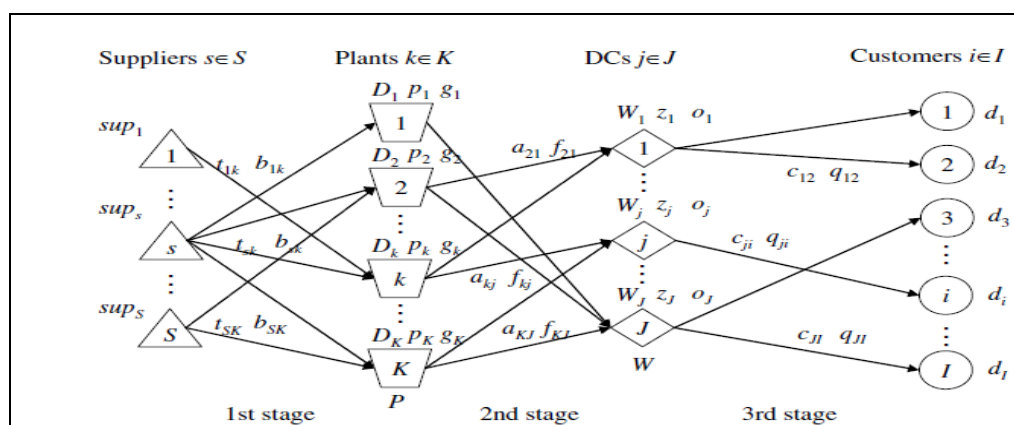


圖 2.4、單一產品來源、多產品供應鏈網絡示意圖

Chern and Hsieh (Chern & Hsieh, 2007)利用有向圖形描繪供應鏈網絡環境，提出一多目標主排程演算法(Multi-objective Master Planning

Algorithm, MOMPA), 目的在於求解缺貨成本、外包情況、原物料、生產、運輸及存貨持有成本最小化但此研究以逐筆訂單滿足的方式，進行訂單分配，其結果無法達到整體最佳解化。因部分產業如記憶體模組產業，其關鍵原物料價格深受時間所影響，故在進行規劃時，亦須考量時間的這項因子。Wu (Wu, 2004)提出考量多產品、多廠區、多時期等因素將不同的廠區、不同的品項及不同的時期視為不同的節點，如圖 2.5, 代表在四個不同時期，三條生產線生產三種不同產品的供應鏈網絡。且考量品項優先順序的狀況下於多階多廠環境進行生產規劃，利用拉式鬆弛法(Lagrangian Relaxation)進行生產規劃，以滿足各時期的需求節點，但此研究並未考量運輸前置時間及原物料供給限制。

此外，Wu (Wu, 2004)進一步證明即使為單一產品、多廠區、多時期、無線產能且無運輸成本的供應鏈網絡生產環境下，其生產規劃問題將會是一個 NP-Complete 的問題，為此，作者提出了以最短路徑(Shortest Path Algorithm)的啟發式演算法來搜尋成本最低的路線，其結果與使用混合正數規劃進行比較，結果顯示使用最短路徑啟發式演算法與混合整數規劃的結果平均誤差在 0.23%，證明最短路徑法是有效解決的方法，唯一美中不足的是，結果仍不是最佳解。

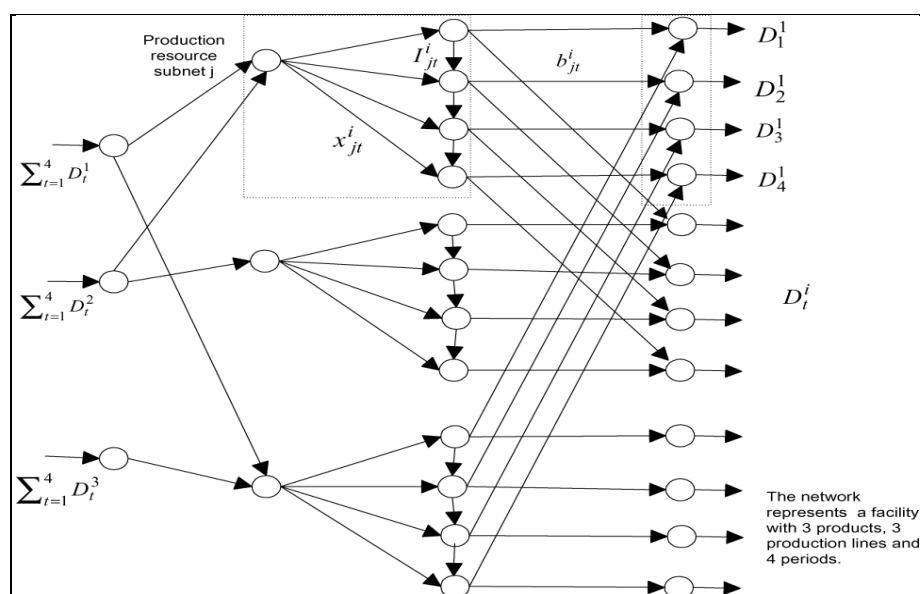


圖 2.5、產品、多廠區、多時期供應鏈網絡圖

以下針對過去文獻對於供應鏈網絡的相關研究其考量特性、限制及求解方法做整理，結果如表 2.1 所示：



表 2.1、供應鏈網絡生產規劃相關研究整理

文獻	求解工具	規劃範圍	規劃特性					
			物料供給限制	訂單交期	產能限制	運輸前置時間	運輸批量成本	產品結構
Thierry et al (1995)	啟發式演算法	單階多廠		✓	✓			
Arntzen et al (1995)	混整數線性規劃	單階多廠	✓		✓			✓
Sauer et al (2000)	混整數線性規劃	單階多廠			✓	✓		
Jayaraman et al(2001)	混整數線性規劃	多階多廠	✓		✓			
Guinet (2001)	對偶線性規劃	單階多廠	✓		✓	✓		✓
Lendermann et al(2001)	模擬方法	多階多廠		✓				
Shen et al. (2003)	代理人	單階多廠		✓		✓		
郭瑞祥(2004)	混合整數規劃	單階多廠	✓		✓			
Wu (2004)	混整數線性規劃	單階多廠		✓	✓			✓
Chan et al. (2005)	基因演算法	單階多廠			✓	✓		
Yeh (2005)	啟發式演算法	多階多廠	✓		✓			
Altiparmak et al. (2006)	基因演算法	多階多廠	✓	✓	✓			
Lim et al. (2006)	模擬方法	多階多廠	✓	✓	✓			
Aliev et al. (2007)	基因演算法	多階多廠		✓	✓			
葉斯暢(2010)	分散式平行計算	多階多廠	✓	✓	✓	✓		

綜合上述而言，針對供應鏈網絡相關研究都各自針對不同的假設提出適當的方法，但針對以數學模型為基礎的分散式平行計算應用於供應鏈網絡的相關研究相當稀少，僅有葉斯暢(葉斯暢, 2010)提出利用分散式平行計算方法應用於供應鏈網絡規劃，該論文證明平行計算可明顯降低運算時間。然而該研究所提出之數學模型並無考量 BOM 表結構，在無考量 BOM 表結構下，製造廠無法精確掌握原物料的狀況；且該論文所考量的多階多廠的生產架構為三階層，與現實的供應鏈網絡模型有所差異；此外，該論文雖有考量運輸成本，但忽略實際狀況存在著因運輸所產生的批量成本；最後，本研究以製造廠為中心，較該論文多考量製造廠開始生產的時間，可較精確掌握製造廠的生產狀況。

## 2.3 平行計算

為了更有效率的解決科學、工程等各方面的需求，以及為了解決更複雜的問題，電腦處理器的運算速度日益進步。然而也因為各界所面臨的問題可能越來越複雜，處理器更新的速度可能跟不上問題複雜的演變程度，也就是說，以往的單一處理器可能已經不足以滿足計算的需求，所以開始有了叢集電腦的產生。叢集電腦藉由一套中介軟體 (Middleware)，此軟體能連結多部電腦具備三大特性：高效能、高可靠度以及簡易操作。透過叢集電腦的概念，利用平行化的技術，解決需要在每台電腦裝置中介軟體的困擾，以獲取單一處理器無法獲得的計算效果。

### 2.3.1 平行計算架構

如上述，平行計算之所以會被提出加以應用，乃是因為現實問題中的複雜度增加了，也因為複雜度增加，若使用單一處理器進行運算，不但運算時間較多處理器長，其結果可能因尚未求得最佳解而較多處理器差。Hwang and Faye(K Hwang & Faye, 1984)提出了早期平行電腦的論述。Hwang and Xu(K Hwang & Xu, 1998)進一步提出「Phase Parallel Model, PPM」將平行計算分成三個階段，包括 Parallelism Phase、Computation Phase 和 Commutation Phase 三個部分。以下針對此三階段做簡短敘述：

1. Parallelism Phase：此階段主要功能是在做前置處理，將欲求解之問題切割成許多部分，並且分散到各個處理器上。
2. Computation Phase：此階段為主要計算過程，透過前階層將問題分割成許多部分分散至多台處理器上進行運算。
3. Commutation Phase：此階層為主要計算過程中進行資料交換、處理資料同步以及資料整合的部分。

綜合上述三階層，作者指出，應盡可能降低第一階層以及第三階的出現的頻率以及時間，原因無他，為了使整體的計算效果達到最好，所以應盡可能降低非運算階層的時間。

平行處理有兩種應用層面：(1) 多個處理器值型不同工作。由許多不同的處理器同時執行不同的工作，使其效率增加，減少因為單處理器造成執行不同工作拖慢整體時間的狀況，這樣的電腦稱之為 OpenMosix。(2) 多

個處理器執行同一個工作。常見的平行計算即為此種狀況，因為單一問題的複雜程度高，所需要花費的計算時間非常的長，藉由透過平行計算的方式，節省整體的運算時間。有別於傳統的循序處理方式（Sequential Processing Method），平行計算是利用多處理器，將一個擁有龐大計算量的工作，加以切割並且將這些分割分散到各個處理器單元上進行運算，藉由多個處理器同時運算，來降低其整體的運算時間。

Wilkinson and Allen (Wilkinson & Allen, 1998)將目前的平行電腦依記憶體的配置分為兩大類：

### 1. 共享式記憶體多處理器系統(Shared memory multi-process system)

又稱為對稱式多處理器系統(Symmetric Memory Multi-Processor System, SMP)，也就是目前所見到的多核心技術，利用系統匯流排（system bus）將系統的CPU、記憶體、I/O設備連接起來，結構如圖 2.6。因其是共享同一記憶體，雖皆可執行循序式以及平行式程式，但當執行程序過於龐大時，記憶體的使用可能就會遇到瓶頸，甚至使整體的執行速度更加緩慢

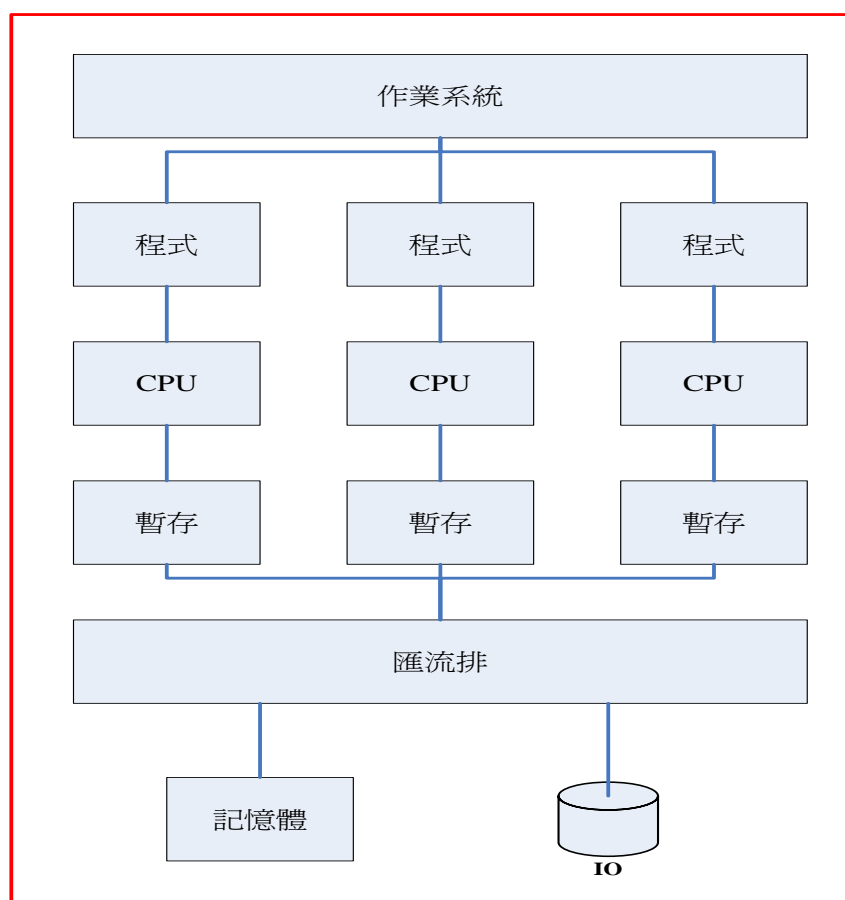


圖 2.6、共享式記憶體多處理器系統結構圖

## 2. 分散式記憶體多處理器系統(Distributed memory multi-process system)

又稱為大量平行處理器系統(Massive Parallel Processor System, MPP)，系統是由多台的電腦組合而成，每台電腦皆有屬於自己的記憶體單元，各自儲存所屬之資料。若欲與他台電腦進行資料存取時，就必須透過訊息傳遞介面網路(Message Passing Interconnection Network)機制來達成溝通、存取資料的目的。在此系統進行平行計算較為複雜，常藉由中介軟體MPI、PVM(Parallel Virtual Machine)來進行資料的交換與溝通，其結構如圖 2.7

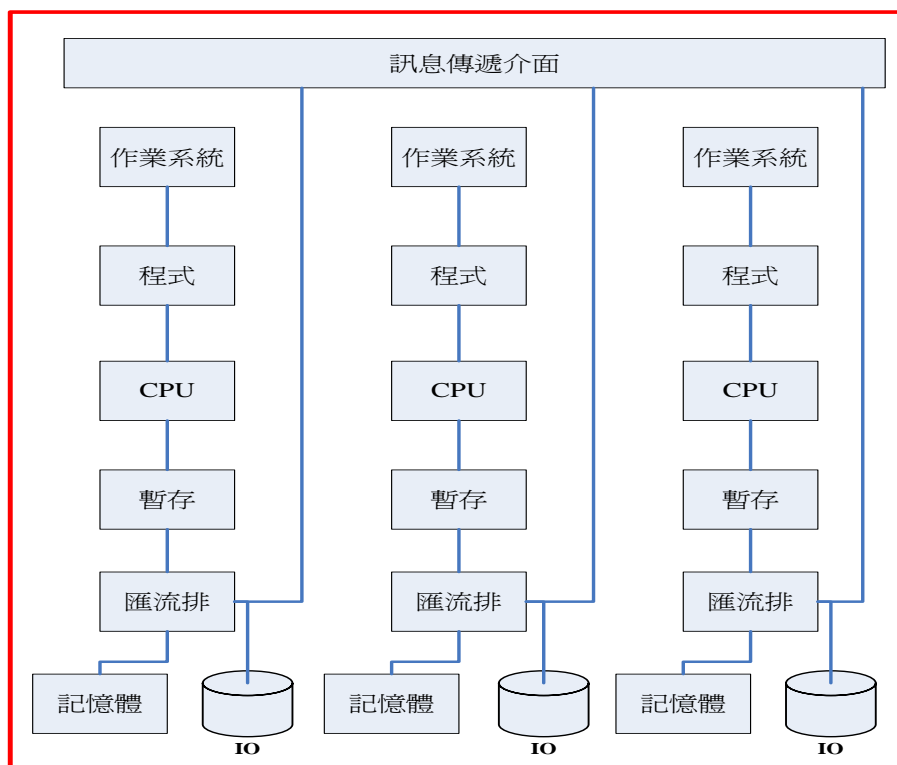


圖 2.7、分散式記憶體多處理器系統結構圖

### 2.3.2 分割與平行化分析

個人叢集電腦系統一般皆屬於分散式記憶體的平行電腦，每台電腦皆有其獨立之記憶體、作業系統、匯流排、CPU 等，對於這樣的平行化電腦，選擇區域分割法進行平行計算。區域分割法是將循序程式中整體區塊進行切割後，分配給數台電腦進行同時運算，讓龐大且複雜的問題平均給各台電腦分擔，使各處理器可以獨立且同時進行運算，減少許多不必要的閒置時間，應用較為廣泛，不受演算法或數學模型限制。而本法所面臨較大的問題是要如何將較為複雜的問題做適當的切割，因為所面臨到的問題可能牽涉到多維度。

如上所述，進行切割也是循序程式與平行程式的差異，平行程式需要將問題進行切割，並且透過多次的通訊以及資料交換，Ian (Ian, 1995)提出一個設計平行系統的方法，稱為有條理的設計 (Methodical Design)，可分為四個部分，分別為切割 (Partitioning)、通訊 (Communication)、凝聚 (Agglomeration) 與對應 (Mapping)，如圖 2.8 所示。將較為繁複的問題進行切割後，透過訊息傳遞介面進行資料傳遞的動作，將分割後的問題對應到不同的處理器上。進行平行計算後，將資料回傳給主控端，但此時會有同步化的問題 (先被分派的工作晚傳回，晚分派的工作早傳回的狀況)，所以作者也建議可待資料都回傳後再進行資料整合，如此一來可以避免同步化產生的問題。

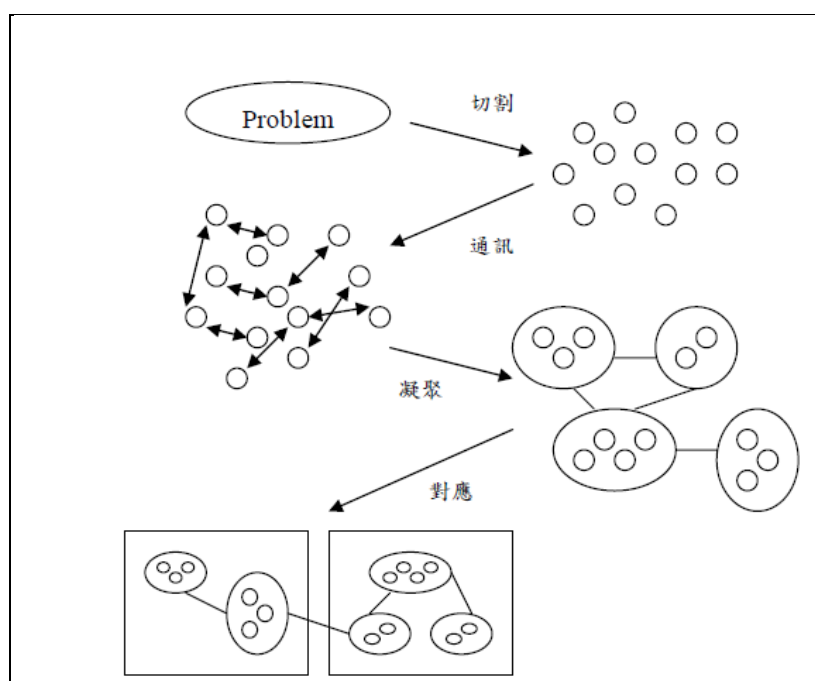


圖 2.8、平行系統設計

陳權(陳權, 2008)說明平行化系統發展速度相對於追逐高速計算的電腦平台來的緩慢，其原因有下列幾項因素：

1. 平行化之程式較傳統序列程式更為複雜，且具有更高技巧與挑戰性。
2. 序列程式要轉換成平行程式會因平行編譯系統不同而有不同的模型。
3. 平行化之除錯程式較少，且除錯的環境難以模擬。
4. 各計算節點的計算量不均或 I/O 與網路寬頻造成系統更複雜。

陳權(陳權, 2008)也指出，除了切割使用的方法外，採用何種平行程式編

譯策略也是影響平行效能的重要因素。訊息傳遞 (MPI) 與 socket 兩種方式，當資料量大時，其平行化效率皆明顯下降。

針對此部分，Java 提供了 RMI (Remote Method Invocation) 的套件，可不必藉由 MPI 與 socket，讓不同主機上的 JAVA 物件進行資料交換，其運作架構圖如圖 2.9

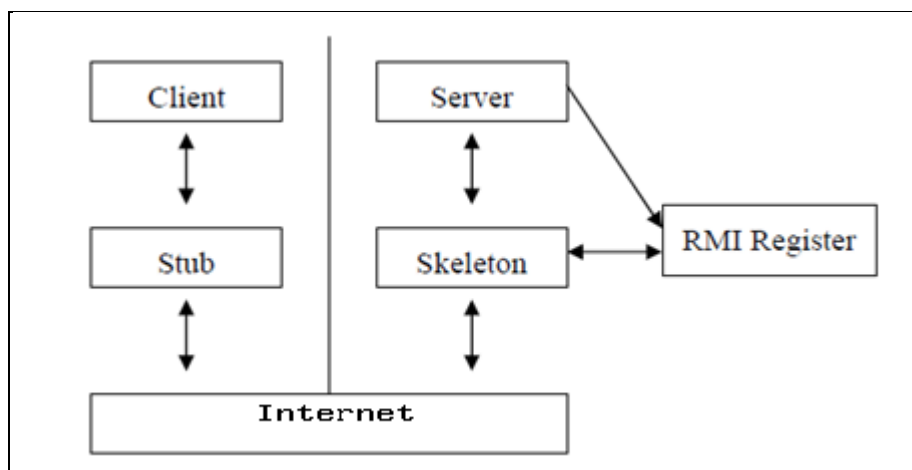


圖 2.9、RMI 運作架構

JavaRMI 存在遠端物件(remote object)，遠端物件存放在各個伺服器端，而各有其一個遠端界面(remote interface)與其對應，透過 RMI 連線時，需要輸入節點的 IP 做為判別，如此才能順利連接其他網路節點。而用戶端可以引用這些已定義好的遠端物件，當呼叫遠端物件的方法時，stub 將封裝傳入方法參數，透過網路傳輸已先註冊 IP 的伺服器端，伺服器端接收到物件後，由 skeleton 拆解參數，呼叫物件方法；反之，回傳資料也是一樣。使用 JavaRMI 減化了使用 socket 傳遞資料的方式 (socket 需解決程式間通訊協定集資料交換的方式，JavaRMI 則省去這方面問題)。

## 第三章 供應鏈網絡生產規劃模式

### 3.1 供應鏈網絡結構

本節將從製造廠的立場，考量供應鏈網絡各階層之間的關係以及產品構。

常見的供應鏈網絡結構大致上可以分成三個階層，如圖 3.1 所示，第一階層為供應商供應原物料給製造廠，而這些供給來源的情形，對於後續階層生產決策的制訂有相當大的影響；第二階層為製造廠生產產品，並將成品運送至配銷中心，其生產產品的數量依據來自各配銷中心；第三階層為配銷中心出貨給最終顧客，而每個階層中又有「多廠」的狀況。以下將針對各個階層所需考量到的特性進行說明：

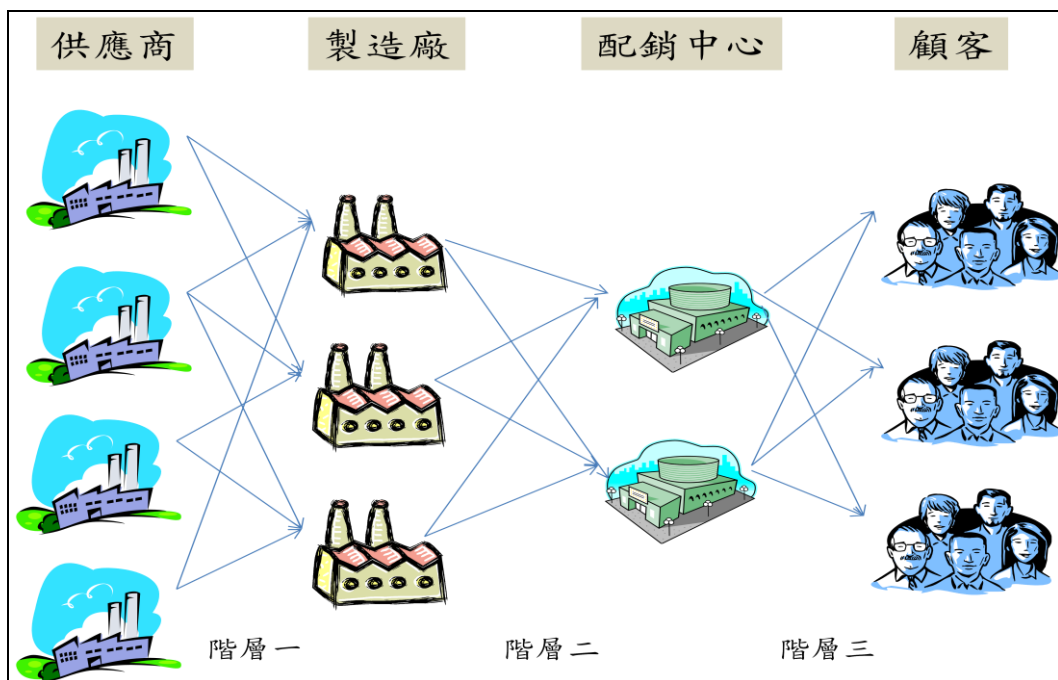


圖 3.1、多階多廠區供應鏈網絡示意圖

#### 3.1.1 第一階層：供應商供應原物料給製造廠

1. 原物料供給限制：各種原物料在規劃週期內，其可供給的數量並非無限上綱，是有限制的，故在進行生產規劃時必須考量到各個供應商及不同原物料階應有供給限制，本研究稱之為供應商產能限制。
2. 作業前置時間：在進行供應鏈網絡生產規劃時，亦須考量到各個供應商的作業前置時間，此時間可能包含供應商之備料、生產以及運輸原物料至製造廠的前置時間。



3. 考量各項成本：第一階層尚須考量到供應商供給原物料給製造廠時所產生的各項成本，其中包含原物料採購成本、運輸成本及因批量造成之運輸批量成本（舉例而言，運送批量設定為 15，但是為了提升服務水準，儘管未達運送批量，依然進行運送的動作，如此一來就會產生運輸批量小於 15 的狀況一種運輸批量成本，運輸量大於 15 的狀況是另一種運輸批量成本）。

### 3.1.2 第二階層：製造廠生產產品，並將成品運送至配銷中心

1. 製造廠產能限制：因所有的製造廠機台設備及人力皆有限的狀況下，各個製造廠的產出能力是有限制的，即便有足夠原物料，但是在產能無法負荷的狀況下，仍然是無法進行生產。
2. 時間因素：進行生產規劃時必須考量時間因素，這些時間因素包括製造廠的生產時間以及自製造廠運送成品至配銷中心的運輸時間，避免規劃出不合理的交貨時間。
3. 考量各項成本：與製造廠相關的成本因素包括製造廠的生產成本、存貨成本、運送成品至配銷中心之運輸成本，以及如前所述之運輸批量成本。
4. 產品結構：在進行生產規劃時，以製造廠的角色是要生產產品，但要這些產品需要哪些原物料就有賴於 BOM 表，從 BOM 表展開後即可以了解要向供應商購買多少原物料。本研究採用 BOM 表為記憶體模組產品，其 BOM 表結構如圖 3.2，以 1G 的 RAM 來說，將由一片的印刷電路板及兩片 512MB 記憶晶片所組成。

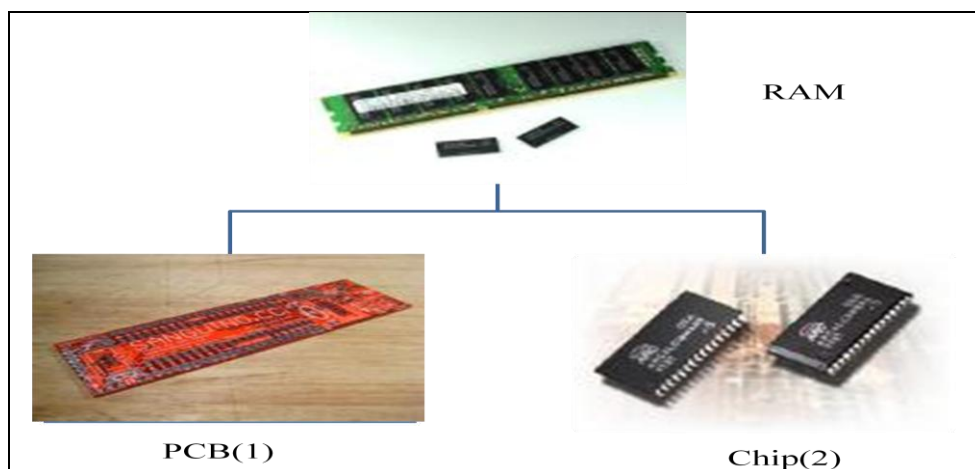


圖 3.2、研究之記憶體模組 BOM 表



### 3.1.3 第三階層：配銷中心分裝出貨給最終顧客

1. 訂單交期時間與產品售價：配銷中心自顧客下訂單時即了解交期時間以及產品單位售價，而在進行供應鏈網絡規劃時，考量各階層的時間因素以及訂單數量與產品售價，採前推的方式，決定出配銷中心及製造廠的交貨時間、製造廠開始生產的時間以及採購的時間。
2. 成本因素，在配銷中心要考量的成本因素有缺貨成本、存貨成本、運送給顧客的運輸成本及運輸批量成本。

綜合上述，在進行供應鏈網絡生產規劃時要考量的特性與限制有很多，而本研究所考量的特性與限制整理如表 3.1，在考量以下限制建立出適合之供應鏈網絡生產規劃數學模型。

表 3.1、本研究考量生產規劃之特性與限制

階層	生產特性及限制條件
第一階層：供應商供給原物料至製造廠	原物料供給限制（供應商產能限制）
	作業前置時間(生產/運輸時間)
	運輸批量
	考量各項成本(原物料/運輸/運輸批量成本)
第二階層：製造廠生產產品，並將成品運送至配銷中心	製造廠產能限制
	作業前置時間(生產/運輸時間)
	運輸批量
	考量各項成本(生產/存貨/運輸/運輸批量成本)
	BOM 表

表 3.1、本研究考量生產規劃之特性與限制(續)

階層	生產特性及限制條件
第三階層：配銷中心分裝 出貨給最終顧客	訂單交期時間
	配銷中心產能限制
	產品售價
	運輸批量
	考量各項成本(缺貨/存貨/運輸/運輸批量成本)
求解型態	整體規劃最佳

### 3.2 供應鏈網絡之數學模型

說明完本研究在進行供應鏈網絡生產規劃模式考量的特性及限制後，接下來的部分將進行本研究的數學模式說明，首先說明本研究的假設條件以已知資訊，透過考量到這些資訊，將其以簡單的圖示化表現出來，接著進一步建立適當的數學模型。建立完數學模型後帶入數理規劃軟體 LINGO 進行運算以及利用分散式平行計算解決供應鏈網絡生產規劃問題，以供規劃人員進行訂單分配時之參考。

#### 3.2.1 假設條件：

1. 製造廠廠區間不進行調撥。
2. 供應商僅提供生產原物料，且各家供應商皆有滿足製造廠原物料種類。
3. 雖然模式容易改變，假設供應商間的各项成本、時間皆為相同(例如：原物料成本、運輸成本、運輸時間等)。
4. 雖然模式容易改變，假設製造廠間的各项成本、時間皆為相同(例如單位生產成本、運輸成本、存貨成本、生產時間、運輸時間等)。
5. 雖然模式容易改變，假設配銷中心間的各项成本、時間皆為相同(例如單位組裝成本、運輸成本、存貨成本、運輸時間等)。

6. 不考慮各廠區安全庫存。
7. 生產良率皆假設成 100%。
8. 製造廠生產時間不受數量影響，設為固定常數，雖然模式容易改變。
9. 本研究製造廠直接生產完成品，交由配銷中心分裝運送給顧客。但考量部分產業配銷中心仍會進行加工之動作，故本研究將製造廠以及配銷中心的成品下標分開表示。
10. 本研究產品結構以 1G 記憶體模組為例，由一片 PCB 及兩片 512MB 的 CHIP 所組成。

### 3.2.2 下標說明

$i$	供應商節點編號	$(i = 1, 2, \dots, I)$
$j$	製造廠節點編號	$(j = 1, 2, \dots, J)$
$k$	配銷中心節點編號	$(k = 1, 2, \dots, K)$
$l$	顧客節點編號	$(l = 1, 2, \dots, L)$
$r$	原物料種類編號	$(r = 1, 2)$
$s$	製造廠成品種類編號	$(s = 1)$
$p$	配銷中心成品種類編號	$(p = 1)$

### 3.2.3 本研究輸入參數及其對應之符號說明：

本小節將從需求面、供給面與成本面三跟層面，分別對於各項輸入參數及其對應符號進行說明，詳細說明如下：

需求面：

規劃週期內訂單數量：

$q_{pl}^R$ ：表示顧客  $l$  購買產品  $p$  之數量。

成品單位售價：

$e_{pl}^R$ ：表示顧客  $l$  購買產品  $p$  之單位售價。

配銷中心訂單交期時間：

$t_k^{SHIP}$ ：表示配銷中心  $k$  之出貨時間。

供給面：

供應商至製造廠之運輸時間：

$t_{ij}^{RT}$ ：表示供應商  $i$  至製造廠  $j$  之運輸時間。

製造廠至配銷中心之運輸時間：

$t_{jk}^{FD}$ ：表示製造廠  $j$  至配銷中心  $k$  之運輸時間。

製造廠生產時間：

$t_j^P$ ：表示製造廠  $j$  之生產時間。

供應商對原物料的生產前置時間：

$t_{ri}^{SP}$ ：表示供應商  $i$  對原物料  $r$  的生產前置時間。

供應商對原物料的產能限制：

$cap_{ri}^S$ ：表示供應商  $i$  供給原物料  $r$  之產能限制。

製造廠對成品的產能限制：

$cap_{sj}^F$ ：表示製造廠  $j$  生產成品  $s$  之產能限制。

配銷中心對成品的產能限制：

$cap_{pk}^D$ ：表示配銷中心  $k$  供應成品  $p$  之產能限制。

原物料對成品的產品結構：

$B_{rs}^{RB}$ ：表示組成成品  $s$  所需要原物料  $r$  之個數。

成本面：

配銷中心缺貨懲罰成本：

$C_{pl}^{SH}$ ：表示顧客  $l$  對成品  $p$  之缺貨懲罰成本。

製造廠原物料採購單位成本：

$C_{rij}^B$ ：表示製造廠  $j$  向供應商  $i$  購買原物料  $r$  之單位成本。

製造廠生產單位成本：

$C_{sjk}^P$ ：表示製造廠  $k$  生產成品  $s$  至配銷中心  $k$  之生產成本。

配銷中心分裝單位成本：

$C_{pkl}^A$ ：配銷中心  $k$  分裝成品  $p$  給顧客  $l$  之分裝成本。

供應商至製造廠運輸成本：

$C_{rij}^{ST}$ ：表示原物料  $r$  自供應商  $i$  運送至製造廠  $k$  之運輸成本。

製造廠至配銷中心運輸成本：

$C_{sjk}^{FT}$ ：表示成品  $s$  自製造廠  $j$  運送至配銷中心  $k$  之運輸成本。

配銷中心至顧客運輸成本：

$C_{pkl}^{DT}$ ：表示成品  $p$  自配銷中心  $k$  運送至顧客  $l$  之運輸成本。

製造廠原物料存貨成本：

$C_r^{RI}$ ：表示原物料  $r$  之存貨成本。

製造廠成品存貨成本：

$C_s^{SI}$ ：表示成品  $s$  之存貨成本。

配銷中心成品存貨成本：

$C_p^{DI}$ ：配銷中心  $p$  之存貨成本。

### 3.2.4 決策變數

1.  $Q_{rij}^{RB}$ ：表示原物料  $r$  自供應商  $i$  運送至製造廠  $j$  之數量。
2.  $Q_{sjk}^{SP}$ ：表示成品  $s$  自製造廠  $j$  運送至配銷中心  $k$  之數量。
3.  $Q_{pkl}^{FA}$ ：表示成品  $p$  自配銷中心  $k$  運送至顧客  $l$  之數量。
4.  $Q_{pl}^{SH}$ ：表示缺貨後補數量
5.  $T_j^{BP}$ ：表示製造廠  $j$  之開始生產時間。

6.  $T_{ri}^S$  : 表示原物料  $r$  對供應商  $i$  之出貨時間。
7.  $T_j^F$  : 表示製造廠  $j$  之出貨時間。
8.  $C_{rij}^{STS}$  : 表示供應商至製造廠運輸批量成本(if 運輸量 $<15 \rightarrow$  成本 10 else 成本 15)。
9.  $C_{sjk}^{FTS}$  : 表示製造廠至配銷中心運輸批量成本(if 運輸量 $<15 \rightarrow$  成本 10 else 成本 15)。
10.  $C_{pkl}^{DTS}$  : 表示配銷中心至顧客運輸批量成本(if 運輸量 $<15 \rightarrow$  成本 10 else 成本 15)。
11.  $P_{rij}^{SF}$  : 表示 0 或 1 的整數變數，當供應商節點  $i$  運送原物料  $r$  至製造廠節點存在時，其值為 1，反之為 0。
12.  $P_{sjk}^{FD}$  : 表示 0 或 1 的整數變數，當製造廠節點  $j$  運送成品  $s$  至配銷中心節點  $k$  存在時，其值為 1，反之為 0。
13.  $P_{pkl}^{DC}$  : 表示 0 或 1 的整數變數，當配銷中心節點  $k$  運送成品  $p$  至顧客  $l$  存在時，其值為 1，反之為 0。

### 3.2.5 圖形化描述供應鏈網絡

以圖式化的方便表達本研究供應鏈網絡各符號所對應的節點以及連接兩階層的連接線，圖形如 3.3 所示，節點部分包含節點編號、節點作業前置時間、節點產能、作業成本、存貨成本等；在連接線的部分包含的有兩階層間的運輸數量、節點的出貨時間以及單位運輸成本。圖 3.3 僅以單廠多階的形式表現之，然而此模型可進一步進化成為多廠的結構，每階層皆有大於一家的節點，在追求利潤最大化的基礎下，決定滿足訂單的最佳規劃路徑，自供應商開始，依據製造廠所需的原物料運送足夠的數量至製造廠，經過製造廠生產後由配銷中心進行分裝，最後交付給最終顧客。以下將針對本研究之數學模型進行詳細說明

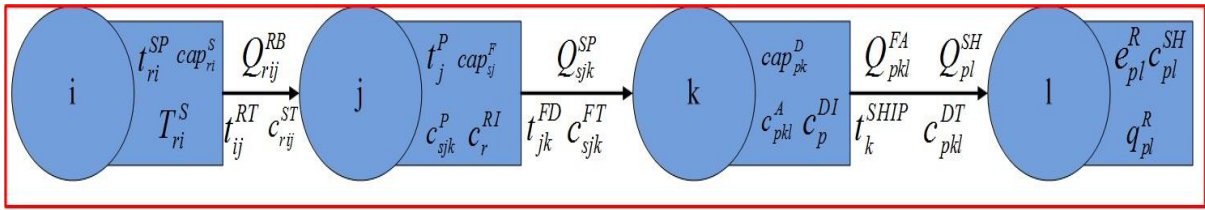


圖 3.3、圖形化符號對應示意圖

### 3.2.6 數學模型目標式

本研究考量四階的供應鏈網絡，包含有 I 間供應商、J 間製造廠、K 間配銷中心以及 L 位最終顧客。而其目標式目的在求取數學模型的利潤最大化，影響利潤的多寡的因素包含總營收以及總成本，其定義如下目標式，接著分別對總營收以及總成本分別做說明：

1. 目標式：

MAX = 規劃週期內總淨利(Gross Profit) = 規劃週期內總營收(Order Revenue)－規劃週期內總成本(Total Cost)。

2. 總營收：

總營收 = 最終顧客需求數量 × 單位售價。

$$\text{總營收} = \sum_{p=1}^P \sum_{l=1}^L (q_{pl}^R \times e_{pl}^R) \quad (1)$$

$q_{pl}^R$ ：表示最終顧客 l 的需求數量。

$e_{pl}^R$ ：表示最終顧客 l 的單位售價。

3. 總成本：

成本的部分本研究將其區分為採購成本 (Purchase Cost)、生產成本 (Production Cost)、存貨持有成本 (Inventory Holding Cost)、運輸成本 (Transportation Cost)、運輸批量成本 (Transportation Setup Cost)、缺貨成本 (Out of Stock Cost)，相關整理如下：

(1) 製造廠採購成本：

製造廠採購成本 = 供應商節點運送原物料種類之數量 × 供應商節點運送原物料種類之單價。

$$\text{製造廠採購成本} = \sum_{r=1}^R \sum_{i=1}^I \sum_{j=1}^J (Q_{rij}^{RB} \times c_{rij}^B) \quad (2)$$

$Q_{rij}^{RB}$ ：表示供應商節點  $i$  運送原物料  $r$  至製造廠節點  $j$  的數量。

$c_{rij}^B$ ：表示供應商節點  $i$  運送原物料  $r$  至製造廠節點  $j$  的單價。

(2) 製造廠製造成本：

製造廠製造成本 = 製造廠節點生產產品種類數量 × 製造廠節點產品種類單位生產成本。

$$\text{製造廠製造成本} = \sum_{s=1}^S \sum_{j=1}^J \sum_{k=1}^K (Q_{sjk}^{SP} \times c_{sjk}^P) \quad (3)$$

$Q_{sjk}^{SP}$ ：表示製造廠節點  $j$  生產產品  $s$  至配銷中心  $k$  的數量。

$c_{sjk}^P$ ：表示製造廠節點  $j$  生產產品  $s$  至配銷中心  $k$  的單位生產成本。

(3) 配銷中心分裝成本：

配銷中心分裝成本 = 配銷中心節點分裝種類數量 × 配銷中心節點分裝種類單位成本。

$$\text{配銷中心分裝成本} = \sum_{p=1}^P \sum_{k=1}^K \sum_{l=1}^L (Q_{pkl}^{FA} \times c_{pkl}^A) \quad (4)$$

$Q_{pkl}^{FA}$ ：表示配銷中心節點  $k$  分裝產品  $p$  至最終顧客  $l$  的數量。

$c_{pkl}^A$ ：表示配銷中心節點  $k$  分裝產品  $p$  至最終顧客  $l$  的單位成本。

(4) 製造廠原物料存貨持有成本：

製造廠原物料存貨持有成本 = 供應商節點運送至製造廠節點之原物料種類數量 × 製造廠節點存貨週期 × 製造廠節點單位存貨持有成本。

$$\text{製造廠原物料存貨持有成本} = \sum_{r=1}^R \sum_{i=1}^I \sum_{j=1}^J [Q_{rij}^{RB} \times (T_j^{BP} - T_{ri}^S - t_{ij}^{RT}) c_j^{RI}] \quad (5)$$

$Q_{rij}^{RB}$ ：表示供應商節點  $i$  運送原物料  $r$  至製造廠節點  $j$  的數量。

$T_j^{BP}$ ：表示製造廠節點  $j$  開始生產的時間。



$T_{ri}^S$  : 表示供應商節點  $i$  供應原物料  $r$  的出貨時間

$t_{ij}^{RT}$  : 表示供應商節點  $i$  運送至製造廠節點  $j$  的前置時間(包含生產及運輸時間)。

$c_j^{RI}$  : 表示製造廠節點  $j$  的單位存貨成本。

製造廠存貨週期示意圖如圖 3.4 所示：

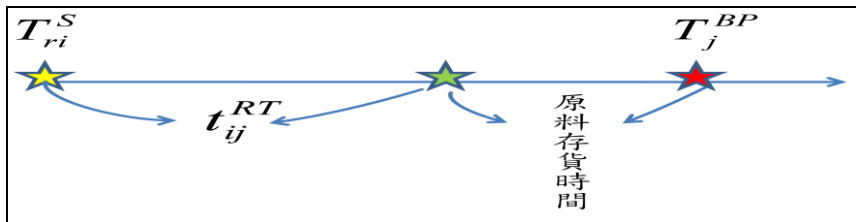


圖 3.4、製造廠存貨週期示意圖

(5) 製造廠產品存貨持有成本：

製造廠產品存貨持有成本 = 製造廠節點生產產品數量 × 製造廠節點生產週期 × 製造廠節點產品單位存貨持有成本。

$$\text{製造廠產品存貨持有成本} = \sum_{s=1}^S \sum_{j=1}^J \sum_{k=1}^K [Q_{sjk}^{SP} \times (T_j^F - T_j^{BP} - t_j^P) c_j^{SI}] \quad (6)$$

$Q_{sjk}^{SP}$  : 表示製造廠節點  $j$  生產產品  $s$  至配銷中心  $k$  的數量。

$T_j^F$  : 表示製造廠節點  $j$  產品出貨時間。

$T_j^{BP}$  : 表示製造廠節點  $j$  開始生產的時間。

$t_j^P$  : 表示製造廠節點  $j$  生產的時間。

製造廠生產週期示意圖如圖 3.5 所示：

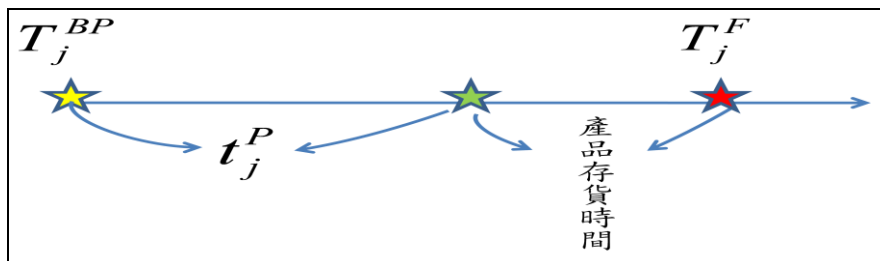


圖 3.5、製造廠生產週期示意圖

(6) 配銷中心存貨持有成本：

配銷中心存貨持有成本 = 製造廠節點生產數量 × 配銷中心節點分裝週期 × 配銷中心節點單位存貨持有成本。

$$\text{配銷中心存貨持有成本} = \sum_{s=1}^S \sum_{j=1}^J \sum_{k=1}^K [Q_{sjk}^{SP} \times (t_k^{SHIP} - T_j^F - t_{jk}^{FD}) c_k^{DI}] \quad (7)$$

$Q_{sjk}^{SP}$  : 表示製造廠節點  $j$  生產產品  $s$  至配銷中心  $k$  的數量。

$t_k^{SHIP}$  : 表示配銷中心節點  $k$  出貨時間。

$T_j^F$  : 表示製造廠節點  $j$  產品出貨時間。

$t_{jk}^{FD}$  : 表示製造廠節點  $j$  運送至配銷中心節點  $k$  的運輸時間。

配銷中心節點分裝週期示意圖如圖 3.6

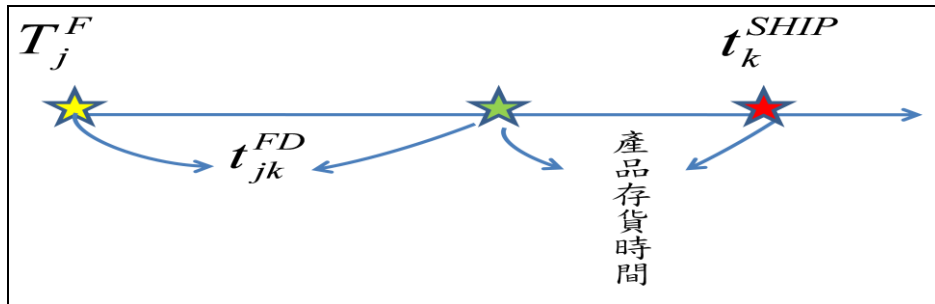


圖 3.6、配銷中心節點分裝週期示意圖

(7) 供應商至製造廠的運輸成本：

供應商至製造廠的運輸成本 = 供應商節點運送至製造廠節點之原物料種類數量 × 供應商節點產品種類單位運輸成本。

$$\text{供應商至製造廠的運輸成本} = \sum_{r=1}^R \sum_{i=1}^I \sum_{j=1}^J (Q_{rij}^{RB} \times c_{rij}^{ST}) \quad (8)$$

$Q_{rij}^{RB}$  : 表示供應商節點  $i$  運送原物料  $r$  至製造廠節點  $j$  的數量。

$c_{rij}^{ST}$  : 表示供應商節點  $i$  運送原物料  $r$  至製造廠節點  $j$  的單價。

(8) 製造廠至配銷中心的運輸成本：

製造廠至配銷中心的運輸成本 = 製造廠節點生產產品數量 × 製造廠節點產品種類單位運輸成本。

$$\text{製造廠至配銷中心的運輸成本} = \sum_{s=1}^S \sum_{j=1}^J \sum_{k=1}^K (Q_{sjk}^{SP} \times c_{sjk}^{FT}) \quad (9)$$

$Q_{sjk}^{SP}$ ：表示製造廠節點  $j$  生產產品  $s$  至配銷中心  $k$  的數量。

$c_{sjk}^{FT}$ ：表示製造廠節點  $j$  運送產品  $s$  至配銷中心  $k$  的單位運送成本。

(9) 配銷中心至最終顧客的運輸成本：

配銷中心至最終顧客的運輸成本 = 配銷中心節點分裝種類數量 × 配銷中心節點種類單位運輸成本。

$$\text{配銷中心至最終顧客的運輸成本} = \sum_{p=1}^P \sum_{k=1}^K \sum_{l=1}^L (Q_{pkl}^{FA} \times c_{pkl}^{DT}) \quad (10)$$

$Q_{pkl}^{FA}$ ：表示配銷中心節點  $k$  分裝產品  $p$  至最終顧客  $l$  的數量。

$c_{pkl}^{DT}$ ：表示配銷中心節點  $k$  運送產品  $p$  至最終顧客  $l$  的單位運輸成本。

(10) 供應商至製造廠的運輸批量成本：

$$\text{供應商至製造廠的運輸批量成本} = \sum_{r=1}^R \sum_{i=1}^I \sum_{j=1}^J c_{rij}^{STS} \quad (11)$$

$c_{rij}^{STS}$ ：表示供應商節點  $i$  運送原物料  $r$  至製造廠節點  $j$  的運輸批量成本

(if  $Q_{rij}^{RB} < 15, c_{rij}^{STS} = 10, \text{ else } c_{rij}^{STS} = 15$ )。

(11) 製造廠至配銷中心的運輸批量成本：

$$\text{製造廠至配銷中心的運輸批量成本} = \sum_{s=1}^S \sum_{j=1}^J \sum_{k=1}^K c_{sjk}^{FTS} \quad (12)$$

$c_{sjk}^{FTS}$ ：表示製造廠節點  $j$  運送產品  $s$  至配銷中心節點  $k$  的運輸批量成本

(if  $Q_{sjk}^{SP} < 15, c_{sjk}^{FTS} = 10, \text{ else } c_{sjk}^{FTS} = 15$ )。

(12) 配銷中心至最終顧客的運輸批量成本：

$$\text{配銷中心至最終顧客的運輸批量成本} = \sum_{p=1}^P \sum_{k=1}^K \sum_{l=1}^L c_{pkl}^{DTS} \quad (13)$$

$c_{pkl}^{DTS}$ ：表示配銷中心節點  $k$  運送產品  $p$  至最終顧客  $l$  的運輸批量成本 (if  $Q_{pkl}^{FA}$

$\langle 15, c_{pkl}^{DTS} = 10, \text{ else } c_{pkl}^{DTS} = 15 \rangle$ 。

綜合方程式 (1) – (13)，本研究供應鏈網絡生產規劃目標函數如下所示：

$$\begin{aligned}
MAX = & \sum_{p=1}^P \sum_{l=1}^L (q_{pl}^R \times e_{pl}^R) - \sum_{r=1}^R \sum_{i=1}^I \sum_{j=1}^J (Q_{rij}^{RB} \times c_{rij}^B) - \sum_{s=1}^S \sum_{j=1}^J \sum_{k=1}^K (Q_{sjk}^{SP} \times c_{sjk}^P) \\
& - \sum_{p=1}^P \sum_{k=1}^K \sum_{l=1}^L (Q_{pkl}^{FA} \times c_{pkl}^A) - \sum_{r=1}^R \sum_{i=1}^I \sum_{j=1}^J [Q_{rij}^{RB} \times (T_j^{BP} - T_{ri}^S - t_{ij}^{rt}) \times c_j^{RI}] \\
& - \sum_{s=1}^S \sum_{j=1}^J \sum_{k=1}^K [Q_{sjk}^{SP} (T_j^F - T_j^{BP} - t_j^p) \times c_j^{SI}] - \sum_{s=1}^S \sum_{j=1}^J \sum_{k=1}^K [Q_{sjk}^{SP} (t_k^{SHIP} - T_j^F - t_{jk}^{FD}) \times c_k^{DI}] \\
& - \sum_{r=1}^R \sum_{i=1}^I \sum_{j=1}^J (Q_{rij}^{RB} \times c_{rij}^{ST}) - \sum_{s=1}^S \sum_{j=1}^J \sum_{k=1}^K (Q_{sjk}^{SP} \times c_{sjk}^{FT}) - \sum_{p=1}^P \sum_{k=1}^K \sum_{l=1}^L (Q_{pkl}^{FA} \times c_{pkl}^{DT}) \\
& - \sum_{r=1}^R \sum_{i=1}^I \sum_{j=1}^J c_{rij}^{STS} - \sum_{s=1}^S \sum_{j=1}^J \sum_{k=1}^K c_{sjk}^{FTS} - \sum_{p=1}^P \sum_{k=1}^K \sum_{l=1}^L c_{pkl}^{DTS} - \sum_{p=1}^P \sum_{l=1}^L (Q_{pl}^{SH} \times c_{pl}^{SH})
\end{aligned}$$

### 3.2.7 數學模型限制式

除了以上所述之目標式外，在考量不同條件限制下，將這些限制以數學符號表示之，以下將針對這些條件限制式做說明：

#### 1. 供需平衡

在規劃週期內，由於供應商供給原物料的種類和數量有限、製造廠以及配銷中心的產能也有限制，所以在對於顧客的下單數量可能不盡然會全部滿足，也就是在配銷中心分裝的總數量 ( $\sum_{k=1}^K \sum_{l=1}^L Q_{pkl}^{FA}$ ) 可能無法滿足顧客訂單 ( $q_{pl}^R$ )，故各配銷中心節點將有缺貨數量 ( $Q_{pl}^{SH}$ ) 存在。限制式如下所示：

配銷中心各節點分裝總數量 + 顧客缺貨總數量 = 顧客下單數量

$$\sum_{k=1}^K \sum_{l=1}^L Q_{pkl}^{FA} + \sum_{l=1}^L Q_{pl}^{SH} = q_{pl}^R \quad \forall l = 1, \dots, L, p = 1, \dots, P \quad (14)$$

$\sum_{k=1}^K \sum_{l=1}^L Q_{pkl}^{FA}$ ：表示配銷中心節點  $k$  供應給顧客  $l$  的數量總和。

$\sum_{l=1}^L Q_{pl}^{SH}$ ：表示顧客端缺貨數量總和。

$q_{pl}^R$ ：表示顧客端的需求數量。

#### 2. 供應商的產能限制

本研究考量供應商在規劃週期內不同原物料的供應有數量上的限制，所以各供應商節點供應各製造廠的總數量 $(\sum_{j=1}^J Q_{rij}^{RB})$ 必須小於該節點可供給的最大數量。且本研究設定廠內不同種類的原物料產能不同，但各廠之間相同原物料的產能相同。限制式表示如下：

$$\sum_{j=1}^J Q_{rij}^{RB} \leq cap_{ri}^S \quad \forall r = 1, \dots, R, i = 1, \dots, I \quad (15)$$

$\sum_{j=1}^J Q_{rij}^{RB}$ ：表示供應商節點*i*供給原物料*r*至製造廠節點*j*的總數量。

$cap_{ri}^S$ ：表示供應商*i*對原物料*r*的產能限制。

### 3. 製造廠的產能限制

本研究考量製造廠在規劃週期內其生產的產能有數量上的限制，所以各製造廠所能生產配送給配銷中心的總數量 $(\sum_{k=1}^K Q_{sjk}^{SP})$ 必須小於該節點可生產的最大數量。限制式表示如下：

$$\sum_{k=1}^K Q_{sjk}^{SP} \leq cap_{sj}^F \quad \forall s = 1, \dots, S, j = 1, \dots, J \quad (16)$$

$\sum_{k=1}^K Q_{sjk}^{SP}$ ：表示製造廠節點*j*生產產品*s*至配銷中心節點*k*的總數量。

$cap_{sj}^F$ ：表示製造廠*j*生產產品*s*的產能限制。

### 4. 配銷中心的產能限制

本研究考量配銷中心在規劃週期內有供給數量的限制，所以各配銷中心節點所能配送給顧客的總數量 $(\sum_{l=1}^L Q_{pkl}^{FA})$ 必須小於該節點可供給的最大數量。限制式表示如下：

$$\sum_{l=1}^L Q_{pkl}^{FA} \leq cap_{pk}^D \quad \forall p = 1, \dots, P, k = 1, \dots, K \quad (17)$$

$\sum_{l=1}^L Q_{pkl}^{FA}$ ：表示配銷中心點*k*分裝配送產品*p*至顧客*l*的總數量。

$cap_{pk}^D$ ：表示配銷中心*k*分裝配送產品*p*的產能限制。

## 5. 供應商至製造廠路徑運輸量及運輸批量成本

因有考量運輸批量，當供應商節點  $i$  運送原物料  $r$  至製造廠節點  $j$ ，若此兩節點的運輸存在時，其運輸數量必須大於等於 10；若節點間的運輸不存在時，其運輸數量則為 0，為判定節點運輸存在與否，在此引進等於 0 或 1 的整數變數 ( $P_{rij}^{SF}$ )。又在考量到運輸批量成本的狀況下，當運輸批量小於 15 時，給定運輸批量成本為 10，若批量大於 15，則其運輸批量成本為 15，為此，引進中介變數 ( $S_{rij}^{TK}$ )，再搭配前述之等於 0 或 1 的整數變數 ( $P_{rij}^{SF}$ )，求得運輸批量成本 ( $C_{rij}^{STS}$ )，其限制式如下所示：

$$\begin{aligned}
 &10 \leq Q_{rij}^{RB} \times P_{rij}^{SF} \\
 &Q_{rij}^{RB} \times P_{rij}^{SF} \leq M \\
 &\text{if } Q_{rij}^{RB} \times P_{rij}^{SF} < 15, S_{rij}^{TK} = 10 \\
 &\text{if } Q_{rij}^{RB} \times P_{rij}^{SF} \geq 15, S_{rij}^{TK} = 15 \\
 &C_{rij}^{STS} = S_{rij}^{TK} \quad \forall r = 1, \dots, R, i = 1, \dots, I, j = 1, \dots, J \quad (18)
 \end{aligned}$$

$Q_{rij}^{RB}$ ：表示供應商節點  $i$  運送原物料  $r$  至製造廠  $j$  的數量。

$P_{rij}^{SF}$ ：表示 0 或 1 的整數變數，當供應商節點  $i$  運送原物料  $r$  至製造廠節點  $j$  存在時，其值為 1，反之為 0。

$M$ ：為 Big M 係數，當運輸存在時，為一個限制不住 ( $Q_{rij}^{RB}$ ) 的係數。

$S_{rij}^{TK}$ ：為一中介變數，以 15 做為基準，判斷存在之運輸量 ( $Q_{rij}^{RB} \times P_{rij}^{SF}$ ) 超過，並給定不同的運輸批量成本。

$C_{rij}^{STS}$ ：表示存在運輸的運輸批量成本，當上述的中介變數成立且存在運輸的狀況下，即可獲得自供應商節點  $i$  運輸原物料  $r$  至製造廠節點  $j$  的運輸批量成本。

## 6. 製造廠至配銷中心路徑運輸量及運輸批量成本

如前限制式所述，因有考量運輸批量，當製造廠節點  $j$  運送成品  $s$  至配銷中心節點  $k$ ，若此兩節點的運輸存在時，其運輸數量必須大於等於 10；若節點間的運輸不存在時，其運輸數量則為 0，為判定節點運輸存在與否，

在此引進等於 0 或 1 的整數變數 ( $P_{sjk}^{FD}$ )。又在考量到運輸批量成本的狀況下，當運輸批量小於 15 時，給定運輸批量成本為 10，若批量大於 15，則其運輸批量成本為 15，為此，引進中介變數 ( $f_{sjk}^{TK}$ )，再搭配前述之等於 0 或 1 的整數變數 ( $P_{sjk}^{FD}$ )，求得運輸批量成本 ( $C_{sjk}^{FTS}$ )，其限制式如下所示：

$$\begin{aligned}
10 &\leq Q_{sjk}^{SP} \times P_{sjk}^{FD} \\
Q_{sjk}^{FD} \times P_{sjk}^{FD} &\leq M \\
\text{if } Q_{sjk}^{SP} \times P_{sjk}^{FD} < 15, & f_{sjk}^{TK} = 10 \\
\text{if } Q_{sjk}^{SP} \times P_{sjk}^{FD} \geq 15, & f_{sjk}^{TK} = 15 \\
C_{sjk}^{FTS} &= f_{sjk}^{TK} \quad \forall s = 1, \dots, S, j = 1, \dots, J, k = 1, \dots, K \quad (19)
\end{aligned}$$

$Q_{sjk}^{SP}$ ：表示製造廠節點  $j$  運送成品  $s$  至配銷中心節點  $k$  的數量。

$P_{sjk}^{FD}$ ：表示 0 或 1 的整數變數，當製造廠節點  $j$  運送成品  $s$  至配銷中心節點  $k$  存在時，其值為 1，反之為 0。

$M$ ：為 Big M 係數，當運輸存在時，為一個限制不住 ( $Q_{sjk}^{SP}$ ) 的係數。

$f_{sjk}^{TK}$ ：為一中介變數，以 15 做為基準，判斷存在之運輸量 ( $Q_{sjk}^{SP} \times P_{sjk}^{FD}$ ) 超過，並給定不同的運輸批量成本。

$C_{sjk}^{FTS}$ ：表示存在運輸的運輸批量成本，當上述的中介變數成立且存在運輸的狀況下，即可獲得自製造廠節點  $j$  運送成品  $s$  至配銷中心節點  $k$  的運輸批量成本。

## 7. 配銷中心至顧客路徑運輸量及運輸批量成本

與前面兩階層的限制式相同，因考量運輸批量，當配銷中心節點  $k$  運送成品  $p$  至顧客  $l$ ，若此兩節點的運輸存在時，其運輸數量必須大於等於 10；若節點間的運輸不存在時，其運輸數量則為 0，為判定節點運輸存在與否，在此引進等於 0 或 1 的整數變數 ( $P_{pkl}^{DC}$ )。又在考量到運輸批量成本的狀況下，當運輸批量小於 15 時，給定運輸批量成本為 10，若批量大於 15，則其運輸批量成本為 15，為此，引進中介變數 ( $d_{pkl}^{TK}$ )，再搭配前述之等於 0 或 1

的整數變數 ( $P_{pkl}^{DC}$ )，求得運輸批量成本 ( $C_{pkl}^{DTS}$ )，其限制式如下所示：

$$\begin{aligned}
10 &\leq Q_{pkl}^{FA} \times P_{pkl}^{DC} \\
Q_{pkl}^{FA} \times P_{pkl}^{DC} &\leq M \\
\text{if } Q_{pkl}^{FA} \times P_{pkl}^{DC} &< 15, d_{pkl}^{TK} = 10 \\
\text{if } Q_{pkl}^{FA} \times P_{pkl}^{DC} &\geq 15, d_{pkl}^{TK} = 15 \\
C_{pkl}^{DTS} &= d_{pkl}^{TK} \quad \forall p = 1, \dots, P, k = 1, \dots, K, l = 1, \dots, L \quad (20)
\end{aligned}$$

$Q_{pkl}^{FA}$ ：表示配銷中心節點  $k$  運送成品  $p$  至顧客  $l$  的數量。

$P_{pkl}^{DC}$ ：表示 0 或 1 的整數變數，當配銷中心節點  $k$  運送成品  $p$  至顧客  $l$  存在時，其值為 1，反之為 0。

$M$ ：為 Big M 係數，當運輸存在時，為一個限制不住 ( $Q_{pkl}^{FA}$ ) 的係數。

$d_{pkl}^{TK}$ ：為一中介變數，以 15 做為基準，判斷存在之運輸量 ( $Q_{pkl}^{FA} \times P_{pkl}^{DC}$ ) 超過，並給定不同的運輸批量成本。

$C_{pkl}^{DTS}$ ：表示存在運輸的運輸批量成本，當上述的中介變數成立且存在運輸的狀況下，即可獲得自配銷中心節點  $k$  運送成品  $p$  至顧客  $l$  的運輸批量成本。

## 8. 供應商出貨時間限制

供應商節點  $i$  的出貨時間 ( $T_{ri}^S$ ) 會大於等於存在運輸的供應商生產前置時間 ( $P_{rij}^{SF} \times t_{ri}^{SP}$ )；並小於供應商節點  $i$  的生產前置時間 ( $t_{ri}^{SP}$ ) 加上存貨週期時間 ( $t_{ri}^{SI}$ )；若供應商各節點運送原物料至製造廠的運輸完全不存在時 ( $\sum_{j=1}^J P_{rij}^{SF} = 0$ )，供應商節點  $i$  的出貨時間為 0，限制式如下：

$$\begin{aligned}
P_{rij}^{SF} \times t_{ri}^{SP} &\leq T_{ri}^S \quad \forall r = 1, \dots, R, i = 1, \dots, I, j = 1, \dots, J \\
T_{ri}^S &\leq t_{ri}^{SP} + t_{ri}^{SI} \quad \forall r = 1, \dots, R, i = 1, \dots, I \\
T_{ri}^S &\leq M \times \sum_{j=1}^J P_{rij}^{SF} \quad \forall r = 1, \dots, R, i = 1, \dots, I \quad (21)
\end{aligned}$$



$T_{ri}^S$ ：表示供應商節點  $i$  對原物料  $r$  的出貨時間。

$t_{ri}^{SP}$ ：表示供應商結節  $i$  對原物料  $r$  的生產前置時間。

$t_{ri}^{SI}$ ：表示供應商結節  $i$  對原物料  $r$  的存貨週期時間。

$P_{rij}^{SF}$ ：表示 0 或 1 的整數變數，當供應商節點  $i$  運送原物料  $r$  至製造廠節點  $j$  存在時，其值為 1，反之為 0。

$M$ ：為 Big M 係數，當  $\sum_{j=1}^J P_{rij}^{SF}$  大於等於 1 時，為一個限制不住( $T_i^S$ )的係數。

## 9. 製造廠出貨時間限制

製造廠節點開始製造的時間( $T_j^{BP}$ )會大於等於供應商節點運輸存在的狀況下，供應商節點  $i$  出貨時間( $T_{ri}^S$ )加上運輸至製造廠節點  $j$  的運輸時間( $t_{ij}^{RT}$ )。製造廠節點的出貨時間( $T_j^F$ )會大於等於製造廠節點開始製造的時間( $T_j^{BP}$ )加上製造廠的生產作業時間。若製造廠各節點運送成品至配銷中心的運輸完全不存在時( $\sum_{k=1}^K P_{sjk}^{FD} = 0$ )，製造廠節點  $j$  的出貨時間為 0，限制式如下：

$$T_j^{BP} \geq (T_{ri}^S + t_{ij}^{RT}) - M \times (1 - P_{rij}^{SF}) \quad \forall r=1, \dots, R, i=1, \dots, I, j=1, \dots, J,$$

$$T_j^F \geq (T_j^{BP} + t_j^P) - M \times (1 - P_{rij}^{SF}) \quad \forall r=1, \dots, R, i=1, \dots, I, j=1, \dots, J$$

$$T_j^F \leq M \times \sum_{k=1}^K P_{sjk}^{FD} \quad \forall s=1, \dots, S, j=1, \dots, J \quad (22)$$

$T_j^{BP}$ ：表示製造廠節點  $j$  開始進行生產的時間。

$T_{ri}^S$ ：表示供應商節點  $i$  對原物料  $r$  的出貨時間。

$t_{ij}^{RT}$ ：表示字供應商節點  $i$  至製造廠節點  $j$  的運輸時間。

$t_j^P$ ：表示製造廠節點  $j$  的生產時間。

$P_{rij}^{SF}$ ：表示 0 或 1 的整數變數，當供應商節點  $i$  運送原物料  $r$  至製造廠節點  $j$  存在時，其值為 1，反之為 0。

$P_{sjk}^{FD}$ ：表示 0 或 1 的整數變數，當製造廠節點  $j$  運送成品  $s$  至配銷中心節點  $k$  存在時，其值為 1，反之為 0。

M：為 Big M 係數，當  $\sum_{k=1}^K P_{sjk}^{FD}$  大於等於 1 時，為一個限制不住( $T_j^F$ )的係數。

## 10. 配銷中心出貨限制

配銷中心節點的出貨時間( $t_k^{SHIP}$ )會大於製造廠的出貨時間( $T_j^F$ )加上製造廠運送成品至配銷中心的運輸時間( $t_{jk}^{FD}$ )，若配銷中心各節點運送成品給顧客的運輸狀況完全不存在時( $\sum_{l=1}^L P_{pkl}^{DC} = 0$ )，配銷中心的出貨時間=0，限制式表示如下：

$$\begin{aligned} t_k^{SHIP} &\geq (T_j^F + t_{jk}^{FD}) - M(1 - P_{sjk}^{FD}) \quad \forall s=1, \dots, S, j=1, \dots, J, k=1, \dots, K, \\ t_k^{SHIP} &\leq M \times \left( \sum_{l=1}^L P_{pkl}^{DC} \right) \quad \forall p=1, \dots, P, k=1, \dots, K \end{aligned} \quad (23)$$

$t_k^{SHIP}$ ：表示配銷中心節點  $k$  的出貨時間。

$T_j^F$ ：表示製造廠節點  $j$  的出貨時間。

$t_{jk}^{FD}$ ：表示製造廠節點  $j$  至配銷中心節點  $k$  的運輸時間。

$P_{sjk}^{FD}$ ：表示 0 或 1 的整數變數，當製造廠節點  $j$  運送成品  $s$  至配銷中心節點  $k$  存在時，其值為 1，反之為 0。

$P_{pkl}^{DC}$ ：表示 0 或 1 的整數變數，當配銷中心節點  $k$  運送成品  $p$  至顧客  $l$  存在時，其值為 1，反之為 0。

M：為 Big M 係數，當  $\sum_{l=1}^L P_{pkl}^{DC}$  大於等於 1 時，為一個限制不住( $t_k^{SHIP}$ )的係數。

## 11. 產品結構限制

製造廠節點欲購買的原物料種類數量( $Q_{rij}^{RB}$ )會等於製造廠節點欲生產之數量( $Q_{sjk}^{SP}$ )乘以產品的 BOM 表結構( $B_{rs}^{RB}$ )，表示如下

$Q_{rij}^{RB} = B_{rs}^{RB} \times Q_{sjk}^{SP}$ ，但考量其限制式之周延，將其等號拆作成大於等於以及小於等於兩個部份，在此引進表示 0 或 1 的整數變數，表示 BOM 表結構存在與否，數學式以下列方式呈現

$$\begin{aligned}
Q_{sjk}^{SP} &\leq \frac{Q_{rij}^{RB}}{B_{rs}^{RB}} \\
-Q_{sjk}^{SP} &\geq -\frac{Q_{rij}^{RB}}{B_{rs}^{RB}} + M \times (1 - Y_v) \\
\forall r=1, \dots, R, i=1, \dots, I, j=1, \dots, J, s=1, \dots, S, v=1, \dots, V
\end{aligned} \tag{24}$$

$Q_{sjk}^{SP}$ ：表示製造廠節點  $j$  生產產品  $s$  至配銷中心節點  $k$  的數量。

$Q_{rij}^{RB}$ ：表示供應商節點  $i$  運送原物料  $r$  至製造廠  $j$  的數量。

$B_{rs}^{RB}$ ：表示生產產品  $s$  需要原物料  $r$  的 BOM 表

$Y_v$ ：表示 0 或 1 的整數變數，當 BOM 存在時，其值為 1，反之為 0。

## 12. 總量守恆限制

各階層的所傳遞的數量必須守恆，如第一階層供應商節點傳遞的數量總和  $(\sum_{i=1}^I Q_{rij}^{RB})$  會等於製造廠節點欲運送至配銷中心節點之總數量  $(\sum_{j=1}^J \sum_{k=1}^K Q_{sjk}^{SP})$  與 BOM 表  $(B_{rs}^{RB})$  的乘積。而第二階層生產之總數量  $(\sum_{j=1}^J Q_{sjk}^{SP})$  會等於顧客向配銷中心節點所訂購的數量總和  $(\sum_{k=1}^K \sum_{l=1}^L Q_{pkl}^{FA})$ 。

$$\begin{aligned}
\sum_{i=1}^I Q_{rij}^{RB} &= \sum_{j=1}^J \sum_{k=1}^K Q_{sjk}^{SP} \times B_{rs}^{RB} & \forall s=1, \dots, S, r=1, \dots, R \\
\sum_{j=1}^J Q_{sjk}^{SP} &= \sum_{k=1}^K \sum_{l=1}^L Q_{PKL}^{FA} & \forall s=1, \dots, S, p=1, \dots, P
\end{aligned} \tag{25}$$

以上為本研究所建立以製造廠為中心，考量 BOM 表、產能、運輸時間等之供應鏈網絡生產規劃數學模型，首先所考量到的供應商、製造廠、配銷中心以及顧客數分別為 3 家供應商、3 家製造廠、3 家配銷中心以及 2 為顧客作為小規模的供應鏈網絡，利用數理規劃軟體 LINGO 10.0 版進行求解。而本模型利用數理規劃軟體進行求解，其計算時間會隨網絡規模變大而有大幅度的增加，甚至在規劃週期時間內無法求得最佳解，相當不具效率。運算條件，兼具降低求解的時間以及求得最佳解的目的。

## 3.3 分散式平行計算架構

### 3.3.1. 循序化平行計算

如前所述，本研究屬於 NP-Complete 的問題，在利用數理規劃軟體 LINGO 10.0 版進行運算，整體的規劃時間隨著規模變大而呈現指數的增加。所以，本節將針對前述之數學模型進行平行化，利用數理規劃軟體 LINGO 所提供的動態連結資料庫 (Dynamic Link Library, DLL) 的功能，搭配 JavaRMI 套件，將數學模型進行分割、分配組合、平行計算、合理的終止，完成平行計算，流程如圖 3.7。

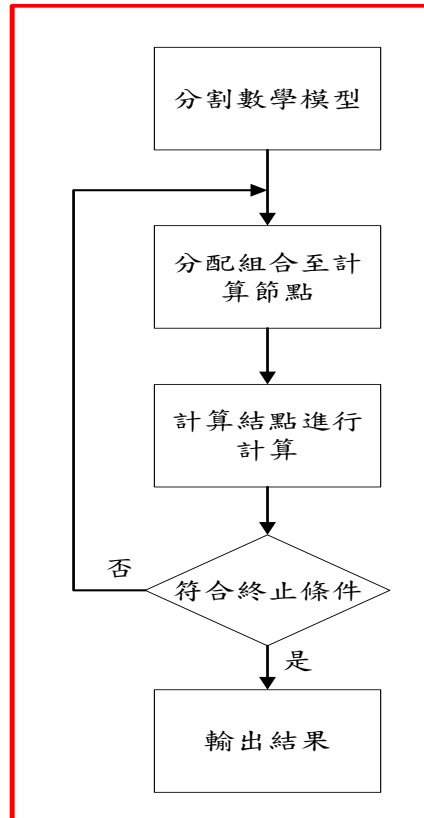


圖 3.7、平行計算流程

從分割數學模型開始，為了使各個計算節點可同時進行計算，必須先將數學模型進行分割，本研究是採取由伺服器端將資料進行分割後，再將分割組合分配至各個計算節點進行運算，而本研究所使用的分割模式是將全域供應鏈網絡的規劃分割成許多較小的區域行供應鏈網絡的組合，分割示意圖如圖 3.8 所示

全域供應鏈網絡			
組合編號	供應商節點編號	製造廠結點編號	配銷中心節點編號
1	1、2、3	1、2、3	1、2、3

↓  
資料分割

區域供應鏈網絡			
組合編號	供應商節點編號	製造廠結點編號	配銷中心節點編號
1	1、2	1、2	1、2、3
2	1、2	1、3	1、2、3
3	1、2	2、3	1、2、3
4	1、3	1、2	1、2、3
5	1、3	1、3	1、2、3
6	1、3	2、3	1、2、3
7	2、3	1、2	1、2、3
8	2、3	1、3	1、2、3
9	2、3	2、3	1、2、3

圖 3.8、資料分割組合

以現實狀況說明，供應商與製造廠的供給產能可能遠大於顧客需求量，顧客之需求量可能僅需由部分供應商以及部分製造廠即可完全供給。舉例來說，假設各家供應商的供給產能為 106 單位，且供應商總共有 3 家（3 家供給產能皆相同）；各家製造廠產能為 281 單位，總共亦有 3 家製造廠（3 家製造廠產能皆相同）。又假設顧客的總需求為 191 單位。以此案例來看，供應商以兩家就可以滿足所有需求，製造廠一家即可生產足夠產品。以此概念做為我們將資料進行分割的方式，在找尋最佳解時，供應商家數即以 3 家中任選取 2 家開始，搭配製造廠家數從 1 家開始，其資料分割組合數即為  $(C_2^3 \times C_1^3)$  開始找尋起，在剛好滿足需求量之前提，開始找尋在更大供給量的狀況下是否有更家的規劃結果，即嘗試  $(C_2^3 \times C_2^3)$ 、 $(C_2^3 \times C_3^3)$  等組合方式，透過資料分割，選取適當參數在相同數學模型進行規劃，找尋出最佳規劃結果。

本研究資料分割由伺服器端進行，而圖 3.8 表示供應商節點家數有 3 家、製造廠節點家數有 3 家、配銷中心節點家數有 3 家，在進行分割後供應商

家數從三家中選取兩家，製造廠也是選取兩家，配銷中心維持三家，如此之分割方式會產生九種的組合數 ( $C_2^3 \times C_2^3 = 9$ )。如圖 3.9，在伺服器端資料分割後，透過 JavaRMI 作為呼叫物件橋樑，伺服器端將分割後的結果以物件的方式傳遞，伺服器端在傳遞資料前，會先判斷計算節點運算狀況。因物件的傳遞藉由網路傳送，所以物件傳送的大小以及次數會影響到整體的規劃時間，所以本研究設計一次傳送一組組合數及其相關參數資料給單一計算節點進行計算，降低可能因資料傳遞所產生的時間

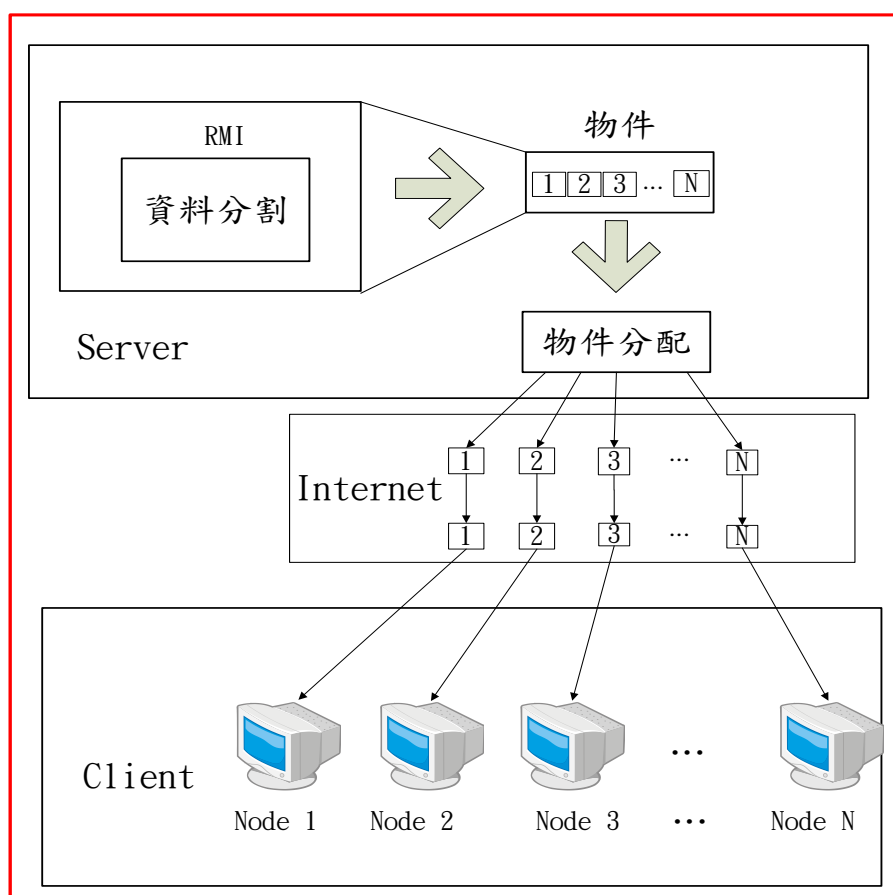


圖 3.9、RMI 架構圖

首先在模型切割的部分，使用者於伺服器端輸入欲選擇的供應商以及製造廠家數在伺服器端中，在輸入供應商以及製造廠的家數有一考量，「所選取的家數其總產能必須盡可能大於等於需求數量」，若供應商及製造廠節點選取家數之總產能小於需求，將會出現缺貨成本，導致平行計算後之所求之目標值與考量全域規劃之目標函數值有所差異。當確定欲選取之供應商以及製造廠家數後，設計一物件計算所有組合數。完成組合數計算後，伺服器端將進行呼叫計算節點之動作，呼叫計算節點端物件的方式是採用多執行緒

(Multithreading)的設計，如此可以不用等待前一個計算節點端計算完畢後才能呼叫第二個計算節點端進行計算；以類別方法 `isAlive()` 回傳之布林值作為測試計算節點是否為閒置的依據，若回傳值為 `true`，表示計算節點目前正進行運算，伺服器端則不分配分割組合之資料；若回傳值為 `false`，表示該計算節點處於閒置狀態，可以進行資料傳遞進行運算。此測試計算節點是否為閒置之動作持續至所有組合數皆計算完畢（所有節點 `isAlive()` 回傳值皆為 `false`）。

計算節點計算完畢時，其結果會保留於計算節點，並且將複製 回傳一份至伺服器端。換言之，當所有的計算節點都計算完畢，伺服器端將會獲取到所有組合數的計算結果，伺服器端依照這些結果，再找出最佳的規劃結果。

伺服器端與計算節點間存在物件互動關係，在 JavaRMI 必須在計算節點端必須事先定義介面，並且類別檔必須執行介面，其介面中所定義的方法物件如下所示：

1. `void indexTransfer(String fName, byte[] fByte)`：此物件主要功能為伺服器端傳送數學模型的參數給計算節點，其中 `fName` 為文字檔的檔案名稱，`fByte` 為文字檔轉換成的位元陣列。
2. `void modelConstruct(int nI, int nJ)`：此物件功能為計算節點根據伺服器端傳送的參數建立數學模型，其中 `nI` 為供應商的選取數量，`nJ` 為製造廠的選取數量。本研究採用的方式為利用 JAVA 建立 LINGO 所需要讀取的 `Ing` 文字檔。
3. `void combinationTransfer(String COMBI_IJ)`：此物件主要功能為計算節點接收伺服器端所分配的組合，其中 `COMBI_IJ` 為依照組合所轉換成的字串。計算節點端將會依照此組合的組合方式進行計算。
4. `byte[] modelSolve(int cCount)`：此物件的主要功能為伺服器端呼叫此物件並且傳送參數給計算節點端後，計算節點端針對所接收到的組合進行數學模型的求解，求解完得到的結果寫入文字檔，轉換成位元陣列並且回傳給伺服器端。其中 `cCount` 為組合的編號，作為用來建立文字檔的檔案名稱的依據。

計算節點端的類別檔執行介面後，伺服器端的程式才能利用 RMI 的方式

進行物件上的呼叫，而計算完的結果是以位元陣列的方式回傳給伺服器端。接著開始說明伺服器端以及計算結點的計算流程，圖 3.10 為伺服器端的運算流程；圖 3.11 為計算節點之計算流程。

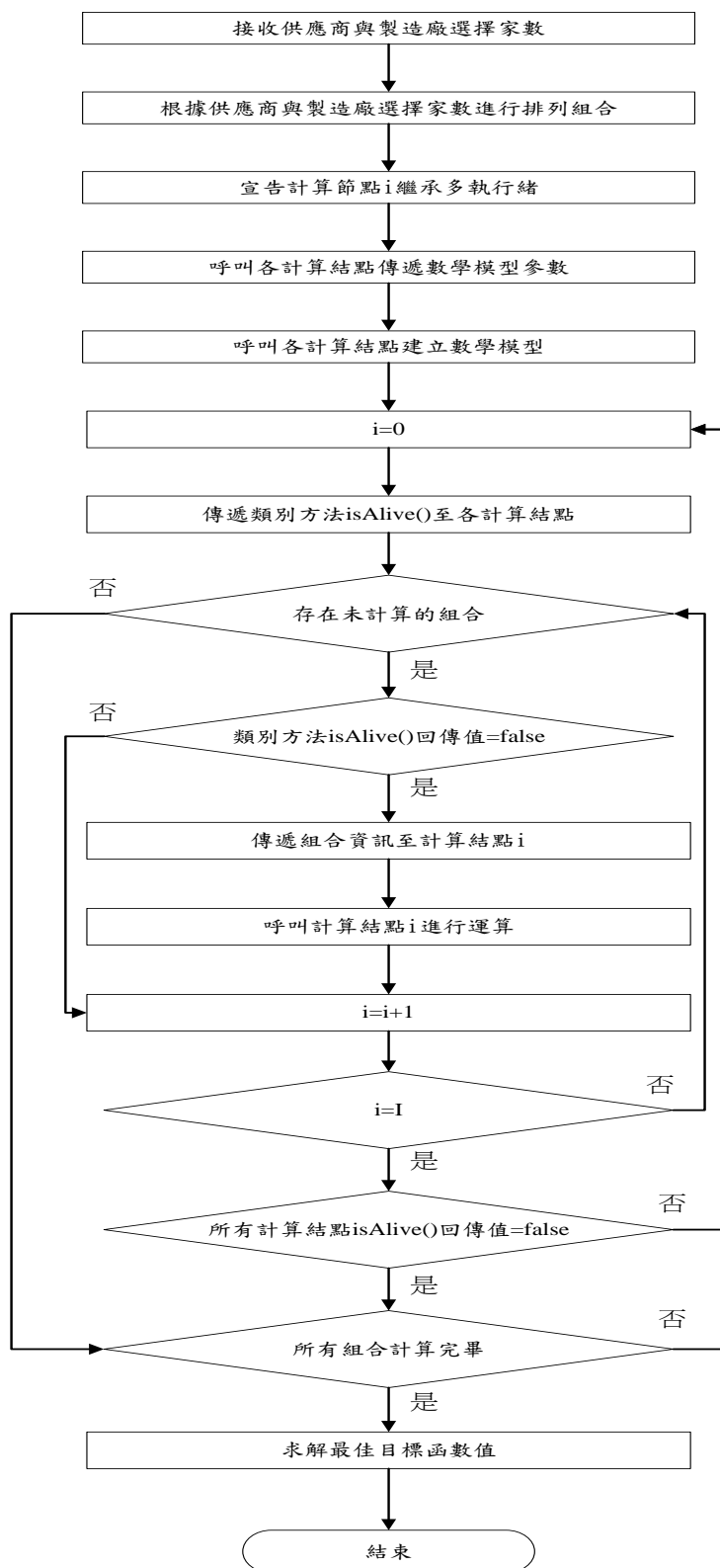


圖 3.10、伺服器端計算流程



對於圖 3.10 說明伺服端的計算流程：

1. 規劃者輸入最小選取的供應商家數  $nI$  與製造廠選取家數  $nJ$ 。
2. 根據實際供應商(I)、製造廠家數(J)與規劃者輸入之  $nI$ 、 $nJ$  產生所有排列組合數，共有  $C_{nI}^I \times C_{nJ}^J$  種組合。
3. 宣告計算節點  $i$  繼承執行緒類別物件，其中  $i=0,1,\dots,I-1$ 。
4. 每個計算節點存在著 `indexTransfer ( fName,fByte[] )` 物件，作為傳遞進行運算時所需之參數，其中 `fName` 為文字檔檔案名稱；`fByte[]` 為文字檔轉換成位元陣列。伺服端透過呼叫各節點 `indexTransfer ( fName,fByte[] )` 物件，傳遞所需參數給計算節點  $i$ 。本研究將各個參數寫作成不同的文字檔格式進行傳遞。
5. 每個計算節點存在著 `modelConstruct ( nI,nJ )` 物件，作為建立數學模型功用，本研究以 Java 將數學模型寫作成 `lng` 檔案格式供 LINGO 讀取。伺服端呼叫各計算節點，將數學模型傳遞給計算節點並要求 LINGO 讀取。
6. 將計算節點  $i$  初始化，設定  $i=0$ ，作為找尋計算節點的起始依據。
7. 根據步驟 2 所運算出之排列組合，伺服端判斷是否存在尚未進行運算之組合，若結果為是，則進行步驟 8 動作，反之則進行步驟 14。
8. 判斷類別方法 `isAlive()` 回傳值是否為 `false`，若回傳值為 `false` 代表該計算節點目前處於閒置狀態則進行步驟 9，反之代表該節點正在進行計算，則進入步驟 11。
9. 呼叫計算節點  $i$  的 `combinationTransfer(COMBI_IJ)` 方法物件，分配尚未進行計算的組合給計算節點  $i$ 。其中 `COMBI_IJ` 為尚未計算的組合所轉換成的字串。
10. 呼叫計算節點  $i$  的 `modelSolve(cCount)` 方法物件，並針對步驟 9 所接收到的組合進行分散式平行計算。其中 `cCount` 為組合的編號，作為用來建立計算結果的檔案命名依據。
11. 對  $i$  進行累加， $i=i+1$ ，作為找尋下一個閒置計算節點之依據。
12. 判斷  $i=I$ ，若為是，表示計算節點  $I$  也處於計算狀態，則進行步驟 13；若為否，則回到步驟 7，找尋尚未計算的組合。

13. 因所有的計算節點都已經計算過任一組合或仍處於計算狀態，故檢查各節點是否都處於閒置狀態，若是，則進入步驟 14，反之。則回到步驟 6。
  14. 判斷所有組合皆已計算完畢，若是進入步驟 15，反之，則回到步驟 6。
- 根據各節點計算各組合後回傳至伺服器端之結果，找尋最佳的規劃結果。

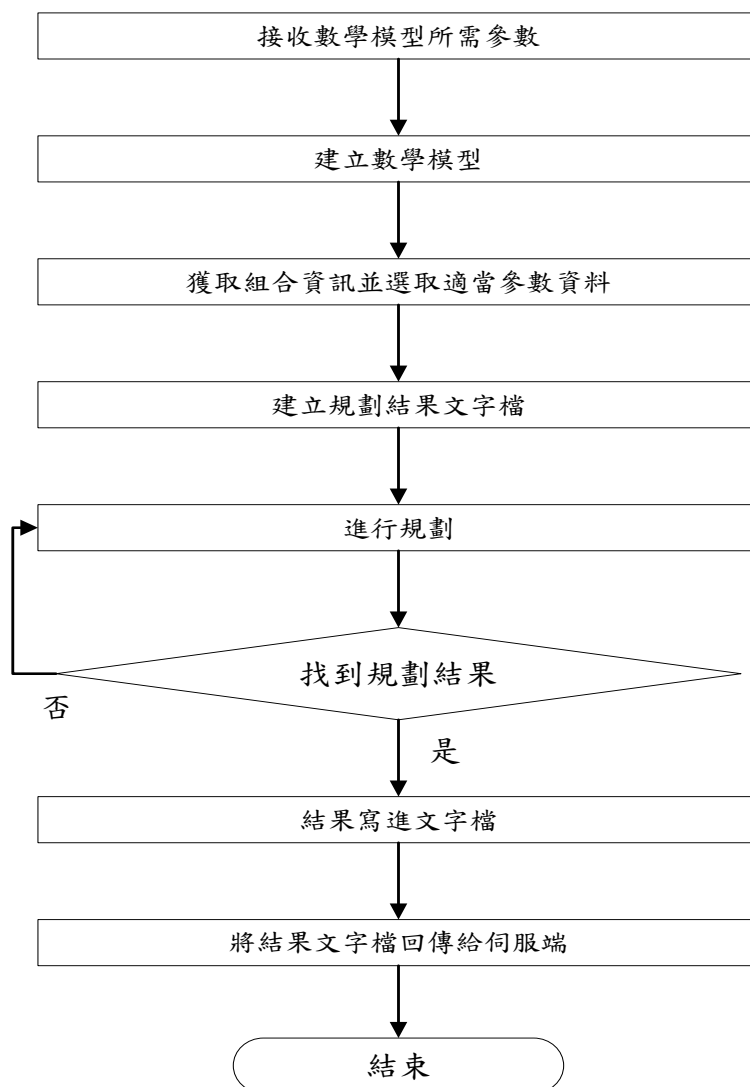


圖 3.11、計算節點計算流程

下列針對計算結點流程做說明：

1. 接收數學模型所需參數乃利用 *indexTransfer(String fName,byte[] fByte)* 方法物件接收自伺服器端傳送之參數文字檔。
2. 建立數學模型乃利用 *modelConstruct(int nI, int nJ)*方法物件接收自伺服器端所傳遞 lng 檔案格式之數學模型供計算結點端 LINGO 讀取。
3. 利用物件方法 *cominationTransfer(String COMBI\_IJ)*方法物件接收伺服

端所分配之組合資訊，並且依照分配之組合讀取對應之參數值。

4. 建立所求結果之文字檔檔案，其檔案名稱以 *cCount* 組合編號參數作為命名依據。
5. LINGO 增加疊代(Iterations)求解數學模型。
6. LINGO 是否計算完畢，是則進入步驟 7，反之回到步驟 5。
7. 利用 *modelSolve(int cCount)* 方法物件將計算結果寫入文字檔。
8. 利用 *modelSolve(int cCount)* 方法物件將文字檔結果以陣列方式回傳至伺服器端。
9. 計算結點計算完畢，等待伺服器端指令。

然而透過上述兩流程進行分散式平行計算，雖將原先較為複雜的數學模型分割成數個規模較小的區域供應鏈網絡進行規劃，但這些較小規模的區域供應鏈網絡在求解過程中，規劃時間不盡然會有所減少。以小規模供應鏈網絡分散式平行計算而言，實際總供應商家數為 3 家，製造廠家數也為 3 家，而規劃者於伺服器端輸入之供應商與製造廠選取家數分別為 2 家與 1 家，在此結構下，其組合數一共有 9 組 ( $C_2^3 \times C_1^3$ )，並利用 9 台處理器進行運算，總規劃時間為 224 秒，各組合規劃結果如下表

表 3.2、分散式平行計算結果

組合編號	規劃時間(秒)	目標函數值
1	102	46645
2	223	46630
3	93	46645
4	78	46625
5	132	46640
6	79	46645
7	183	46640
8	196	46635
9	109	46645

結果顯示組合編號 2 所花費的時間為最多，但其規劃結果並非最佳，而

最佳的規劃結果組合至多僅需要 109 秒的規劃時間，在規劃上多花了 114 秒，卻無找尋到較佳的規劃結果，顯然此規劃結果並不合理，故將採用 LINGO 在進行規劃時找尋最佳解上下界 (Upper & Lower Bound) 的概念進行規劃，更能有效解決最佳計算時間上的問題。

### 3.3.2. LINGO 上界與下界

在 LINGO DLL 中提供了 *LSgetCallbackInfoDoubleLng( int nLngEnv, int nObject, double dCBInfo[])* 的方法，此方法藉由觀察 LINGO 求解過程中上界(Upper Bound)與下界(Lower Bound)，當上下界交會時，即為所求之目標值，根據此種方法將這兩種資訊隨著時間改變而記錄下來，並且作一些資料上的整理，其結果如下圖(橫軸為時間(秒)、縱軸為值)：

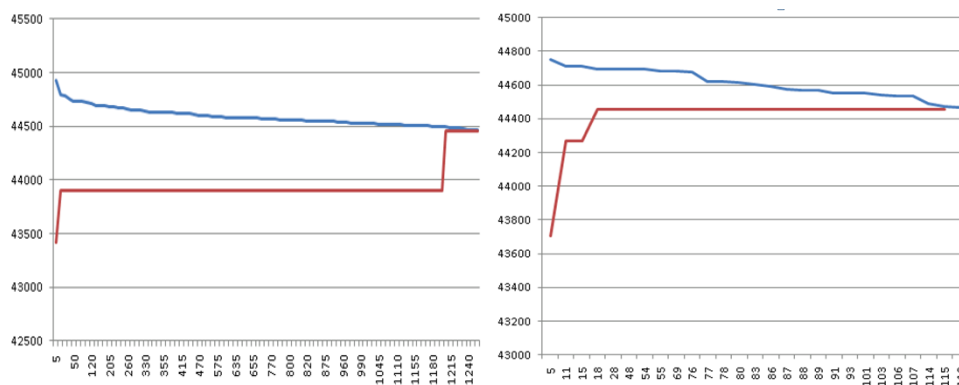


圖 3.12、LINGO 計算上界與下界變化

本研究提出兩種上下界變化模式，第一種為圖 3.12 左方，其下界於規劃開始時急速上升至某一特定值，其後相當長的一段時間內並無變化，直至上界收斂至某一數值，下界才急速上升與其交會求出結果。第二為圖 3.12 右方，下界於一開始急速上升至某一特定值後就無明顯改變，直至上界收斂與其交會，對於在何種狀況下應停止，利用以下概念使平行計算更具效率：在進行規劃時，所選取的供應商以及製造廠的家數越多，越接近全域最佳規劃，故其規劃結果必定大於等於供應商及製造廠家數選取較少者，舉例而言，小規模所選取供應商家數為 3 家、製造廠選取家數為 2 家，其規劃結果必定大於等於供應商選取家數為 2 家、製造廠選取家數為 2 家之結果。故當以給定目標函數值 (或先進行較小規模規劃，其中此論述之較小規模意指選取供應商、製造廠家數較少者，並非環境規模大小。)，若進行平行運算時，部分組合之上界已經小於給定之目標函數值，則該組合之

目標函數值也必定小於所給定的目標函數值，若與此種狀況，則終止該計算節點規劃，其示意圖如下：

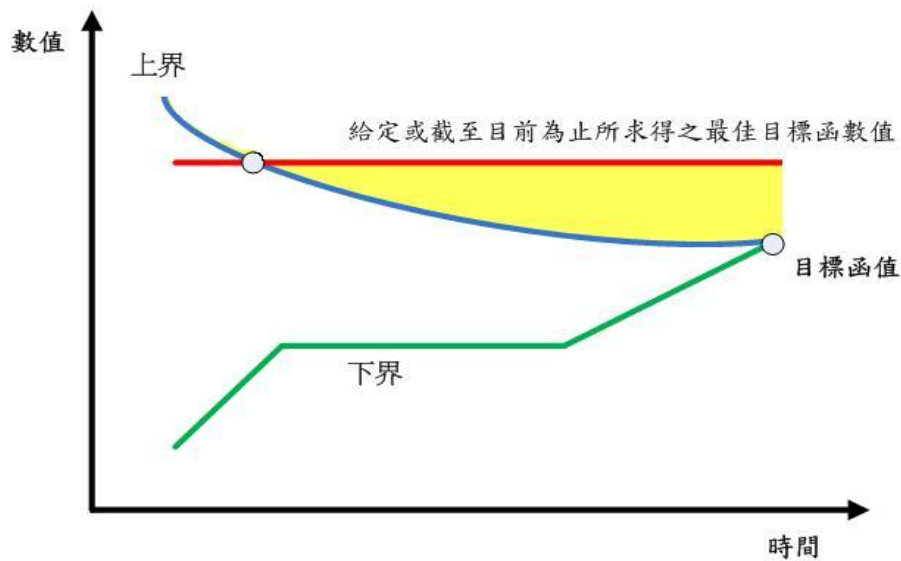


圖 3.13、以給定或截至目前為止所求得之最佳目標函數值控制上界

說明圖 3.13，當進行分散式平行計算前，由規劃者於伺服器端輸入一初始值作為當前所求得之最佳目標函數值，而進行平行計算後，若有較佳的規劃值出現時，則以該較佳規劃數值取代之，作為上界比較標準。然而並非每個組合皆可求得不小於上界標準之規劃結果，所以當任一組合之上界值低於該標準值，代表該組合之目標函數值必定小於當前之最佳目標函數值（因每組合目標函數值為上下界相交點）。當遇到如此情況，則終止該組合之運算進行下一組合。

根據上述之上下界觀念，在進行 JavaRMI 所執行的界面中，增加一物件負責溝通伺服器端以及計算節點間之組合最佳規劃結果。以下為界面所定義的物件說明：

1. *void indexTransfer(String fName, byte[] fByte)*：此物件主要功能為伺服器端傳送數學模型的參數給計算節點，其中 *fName* 為文字檔的檔案名稱，*fByte* 為文字檔轉換成的位元陣列。
2. *void modelConstruct(int nI, int nJ)*：此物件功能為計算節點根據伺服器端傳送的參數建立數學模型，其中 *nI* 為供應商的選取數量，*nJ* 為製造廠的選取數量。本研究所採用的方式為利用 JAVA 建立 LINGO 所需要讀取

的 `lng` 文字檔。

3. `void UBLimit(double PREV_Obj)`: 此物件主要功能為計算節點端根據伺服器端傳送的參數來限制上界。參數 `PREV_Obj` 為子集合的最佳規劃結果之目標值。
4. `void combinationTransfer(String COMBI_IJ)`: 此物件主要功能為計算節點端接收伺服器端所分配的組合，其中 `COMBI_IJ` 為依照組合所轉換成的字串。計算節點端將會依照此組合的組合方式進行計算。
5. `byte[] modelSolve(int cCount)`: 此物件的主要功能為伺服器端呼叫此物件並且傳送參數給計算節點端後，計算節點端針對所接收到的組合進行數學模型的求解，求解完得到的結果寫入文字檔，轉換成位元陣列並且回傳給伺服器端。其中 `cCount` 為組合的編號，作為用來建立文字檔的檔案名稱的依據。

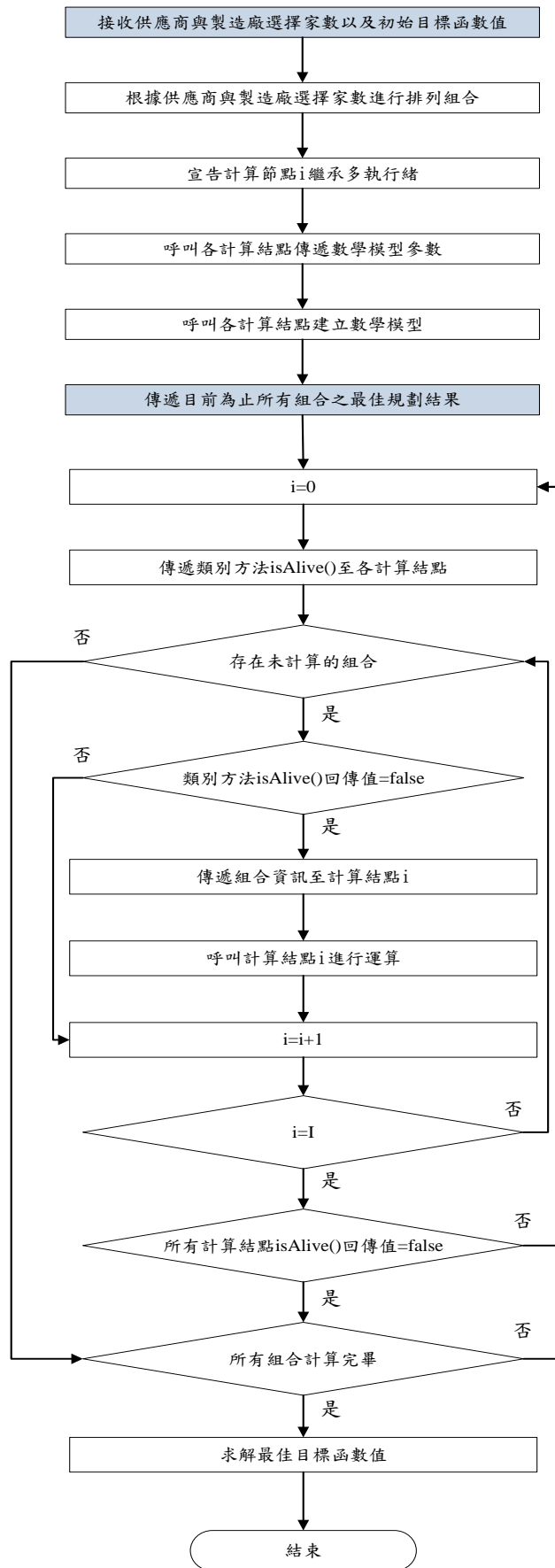


圖 3.14、伺服器端計算流程(使用最佳規劃結果控制上界)

圖 3.14 為說明伺服器端使用最佳規劃結果控制上界的計算流程，大致流程與未使用控制上界流程差異不大，僅有兩個部份以不同顏色作為區隔，以下為詳細流程說明，差異之步驟為步驟 1，規劃者除輸入供應商及製造廠選取家數外，尚需輸入初始目標函數值；另一差異為新增步驟 6，說明如下：

1. 規劃者輸入最小選取的供應商家數  $nI$  與製造廠選取家數  $nJ$ ，並且輸入初始規劃目標函數值，一般設定為 0。
2. 根據實際供應商(I)、製造廠家數(J)與規劃者輸入之  $nI$ 、 $nJ$  產生所有排列組合數，共有  $C_{nI}^I \times C_{nJ}^J$  種組合。
3. 宣告計算節點  $i$  繼承執行緒類別物件，其中  $i=0,1,\dots,I-1$ 。
4. 每個計算節點存在著 `indexTransfer ( fName,fByte[] )` 物件，作為傳遞進行運算時所需之參數，其中 `fName` 為文字檔檔案名稱；`fByte[]` 為文字檔轉換成位元陣列。伺服器端透過呼叫各節點 `indexTransfer ( fName,fByte[] )` 物件，傳遞所需參數給計算節點  $i$ 。本研究將各個參數寫作成不同的文字檔格式進行傳遞。
5. 每個計算節點存在著 `modelConstruct ( nI,nJ )` 物件，作為建立數學模型功用，本研究以 Java 將數學模型寫作成 `lng` 檔案格式供 LINGO 讀取。伺服器端呼叫各計算節點，將數學模型傳遞給計算節點並要求 LINGO 讀取。
6. 呼叫每個計算節點的 `UBLimit(PREV_Obj)`，傳送初始設定或截至當前所有組合最佳規劃結果給各個計算節點  $i$ ，以此 `PREV_Obj` 的值來限制住上界的值。其中 `PREV_Obj` 為初始設定值或截至當前所有組合最佳規劃結果。
7. 將計算節點  $i$  初始化，設定  $i=0$ ，作為找尋計算節點的起始依據。
8. 根據步驟 2 所運算出之排列組合，伺服器端判斷是否存在尚未進行運算之組合，若結果為是，則進行步驟 9 動作，反之則進行步驟 15。
9. 判斷類別方法 `isAlive()` 回傳值是否為 `false`，若回傳值為 `false` 代表該計算節點目前處於閒置狀態則進行步驟 10，反之代表該節點正在進行計算，則進入步驟 12。



10. 呼叫計算節點  $i$  的 *combinationTransfer*(*COMBI\_IJ*)方法物件，分配尚未進行計算的組合給計算節點  $i$ 。其中 *COMBI\_IJ* 為尚未計算的組合所轉換成的字串。
11. 呼叫計算節點  $i$  的 *modelSolve*(*cCount*)方法物件，並針對步驟 9 所接收到的組合進行分散式平行計算。其中 *cCount* 為組合的編號，作為用來建立計算結果的檔案命名依據。
12. 對  $i$  進行累加， $i=i+1$ ，作為找尋下一個閒置計算節點之依據。
13. 判斷  $i=I$ ，若為是，表示計算節點  $I$  也處於計算狀態，則進行步驟 14；若為否，則回到步驟 8，找尋尚未計算的組合。
14. 因所有的計算節點都已經計算過任一組合或仍處於計算狀態，故檢查各節點是否都處於閒置狀態，若是，則進入步驟 15，反之。則回到步驟 7。
15. 判斷所有組合皆已計算完畢，若是進入步驟 16，反之，則回到步驟 7。
16. 根據各節點計算各組合後回傳至伺服端之結果，找尋最佳的規劃結果。

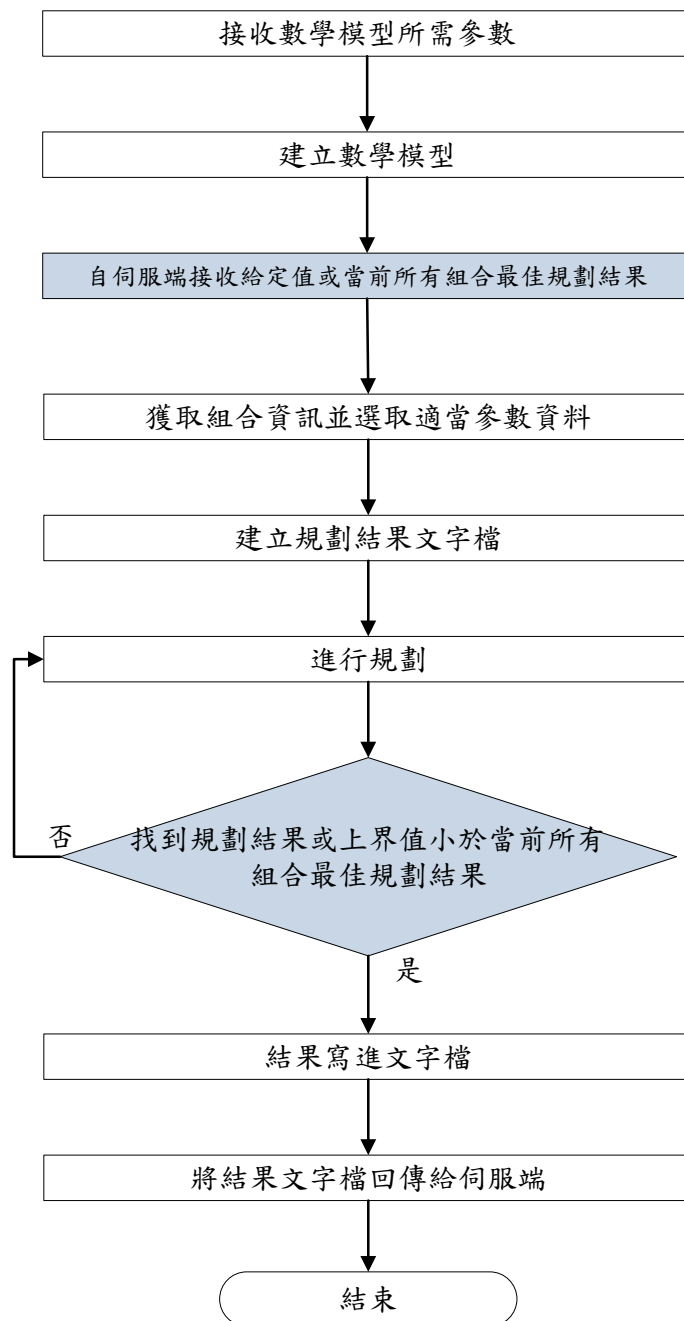


圖 3.15、計算節點計算流程(使用最佳規劃結果控制上界)

上圖為計算節點在使用最佳規劃結果控制上界的計算流程，與伺服器端流程一樣，流程圖亦將與未使用控制上界流程相異之處以不同顏色標出，以下為計算節點的計算流程：

1. 接收數學模型所需參數乃利用 *indexTransfer(String fName,byte[] fByte)* 方法物件接收自伺服器傳送之參數文字檔。
2. 建立數學模型乃利用 *modelConstruct(int nI, int nJ)*方法物件接收自伺服器所傳遞 lng 檔案格式之數學模型供計算結點端 LINGO 讀取。

3.  $UBLimit(double\ PREV\_Obj)$ 物件接收從伺服器端傳送的參數限制  $Upper\ Bound$ 。參數  $PREV\_Obj$  為初始設定值或截至當前所有組合最佳規劃結果。
4. 利用物件方法  $combinationTransfer(String\ COMBI\_IJ)$ 方法物件接收伺服器端所分配之組合資訊，並且依照分配之組合讀取對應之參數值。
5. 建立所求結果之文字檔檔案，其檔案名稱以  $cCount$  組合編號參數作為命名依據。
6. LINGO 增加疊代(Iterations)求解數學模型。
7. LINGO 是否計算完畢或者上界的值小於  $PREV\_Obj$  的值時，判別結果為是的話，進入步驟 8，反之則進入步驟 6。
8. 利用  $modelSolve(int\ cCount)$ 方法物件將計算結果寫入文字檔。
9. 利用  $modelSolve(int\ cCount)$ 方法物件將文字檔結果以陣列方式回傳至伺服器端。
10. 計算結點計算完畢，等待伺服器端指令。

在實驗時發現一個狀況，平行過程中伺服器端會陸陸續續接收到自計算節點所規劃的結果，然而在這些結果中可能存在著比當前所有組合最佳規劃結果（控制上界值）還要更好的目標函數值，如圖 3.16。面對此狀況，伺服器端就必須依據自計算節點回傳結果，不斷判別是否有更好的控制上界值，若有，則立即更新  $PREV\_Obj$  值，並且將新的控制上界值傳遞給各計算節點，也因為計算節點接收到新的控制上界值，也可能立即中止當下的運算（因接收到新的控制上界值已大於目前進行運算中的上界值）。

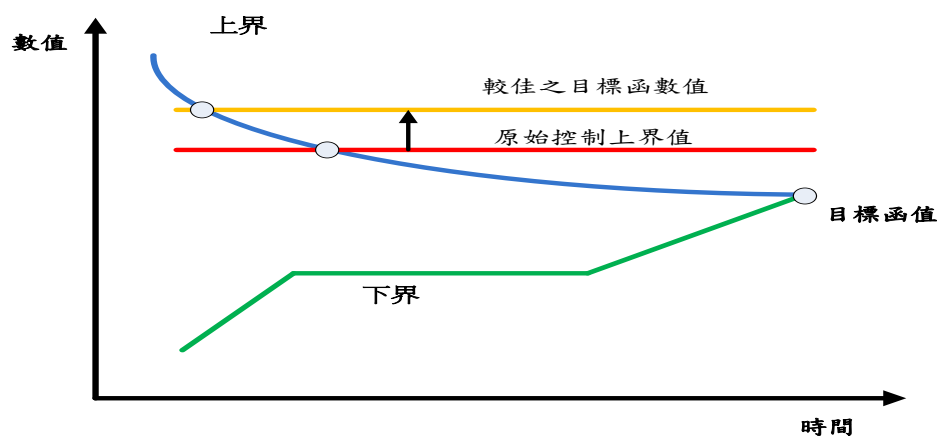


圖 3.16、平行計算後得到較佳之目標函數值

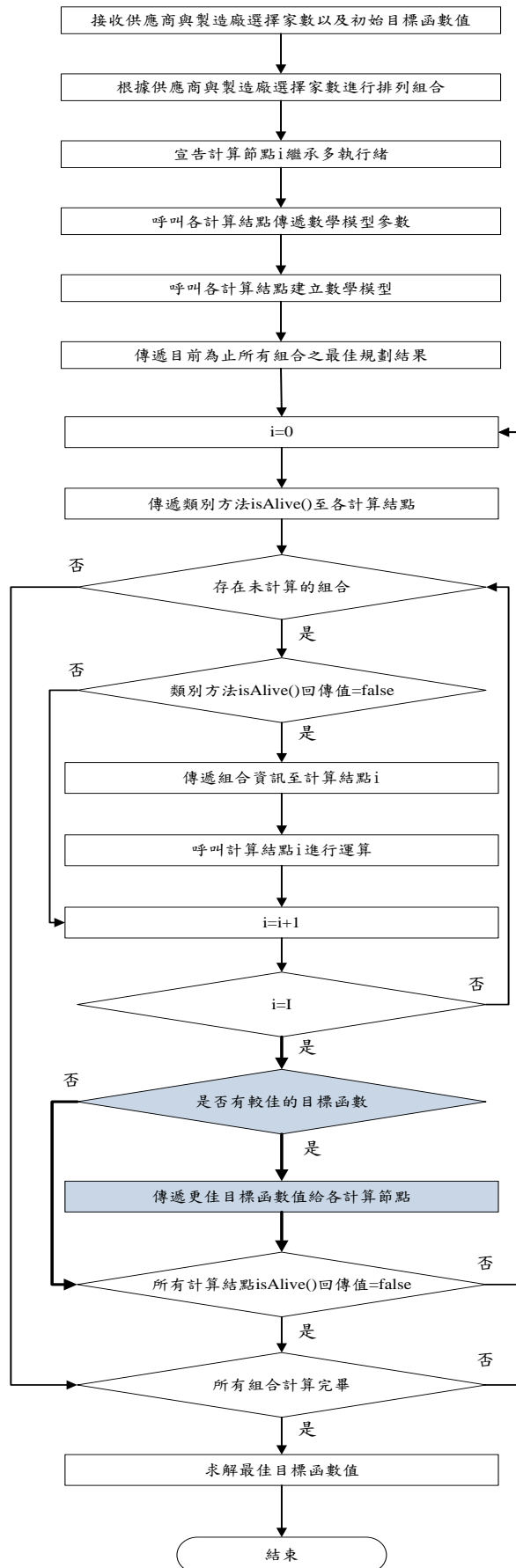


圖 3.17、伺服端計算流程(取得更佳目標函數值取代原始控制上界值)

若利用前述概念，將其控制機制交由伺服器端，判斷是否有較佳的目標函數值，並傳遞給各計算節點，其計算流程如圖 3.17，與使用最佳規劃結果控制上界的計算流程相較，取得更佳目標函數值取代原始控制上界值流程多了兩個步驟，一步驟為判斷是否有較佳目標函數值出現，接著將新目標函數值傳遞給計算結點作為新控制上界值，詳細說明如下：

1. 規劃者輸入最小選取的供應商家數  $nI$  與製造廠選取家數  $nJ$ ，並且輸入初始規劃目標函數值，一般設定為 0。
2. 根據實際供應商(I)、製造廠家數(J)與規劃者輸入之  $nI$ 、 $nJ$  產生所有排列組合數，共有  $C_{nI}^I \times C_{nJ}^J$  種組合。
3. 宣告計算節點  $i$  繼承執行緒類別物件，其中  $i=0,1,\dots,I-1$ 。
4. 每個計算節點存在著 `indexTransfer (fName,fByte[])` 物件，作為傳遞進行運算時所需之參數，其中 `fName` 為文字檔檔案名稱；`fByte[]` 為文字檔轉換成位元陣列。伺服器端透過呼叫各節點 `indexTransfer (fName,fByte[])` 物件，傳遞所需參數給計算節點  $i$ 。本研究將各個參數寫作成不同的文字檔格式進行傳遞。
5. 每個計算節點存在著 `modelConstruct (nI,nJ)` 物件，作為建立數學模型功用，本研究以 Java 將數學模型寫作成 `lng` 檔案格式供 LINGO 讀取。伺服器端呼叫各計算節點，將數學模型傳遞給計算節點並要求 LINGO 讀取。
6. 呼叫每個計算節點的 `UBLimit(PREV_Obj)`，傳送初始設定或截至當前所有組合最佳規劃結果給各個計算節點  $i$ ，以此 `PREV_Obj` 的值來限制住上界的值。其中 `PREV_Obj` 為初始設定值或截至當前所有組合最佳規劃結果。
7. 將計算節點  $i$  初始化，設定  $i=0$ ，作為找尋計算節點的起始依據。
8. 根據步驟 2 所運算出之排列組合，伺服器端判斷是否存在尚未進行運算之組合，若結果為是，則進行步驟 9 動作，反之則進行步驟 15。
9. 判斷類別方法 `isAlive()` 回傳值是否為 `false`，若回傳值為 `false` 代表該計算節點目前處於閒置狀態則進行步驟 10，反之代表該節點正在進行計算，則進入步驟 12。

10. 呼叫計算節點  $i$  的  $combinationTransfer(COMBI\_IJ)$  方法物件，分配尚未進行計算的組合給計算節點  $i$ 。其中  $COMBI\_IJ$  為尚未計算的組合所轉換成的字串。
11. 呼叫計算節點  $i$  的  $modelSolve(cCount)$  方法物件，並針對步驟 9 所接收到的組合進行分散式平行計算。其中  $cCount$  為組合的編號，作為用來建立計算結果的檔案命名依據。
12. 對  $i$  進行累加， $i=i+1$ ，作為找尋下一個閒置計算節點之依據。
13. 判斷  $i=I$ ，若為是，表示計算節點  $I$  也處於計算狀態，則進行步驟 14；若為否，則回到步驟 8，找尋尚未計算的組合。
14. 伺服端對目前所接收到的結果進行判斷，判斷各目標函數值是否較原始控制上界值好，若是，進入步驟 15，反之則進入步驟 16。
15.  $PREV\_Obj$  由較佳的目標函數值所取代，伺服端呼叫每一個計算節點  $i$  的  $UBLimit(PREV\_Obj)$  物件，傳送更佳的目標值給各個計算節點  $i$ ，作為新的控制上界值。
16. 因所有的計算節點都已經計算過任一組合或仍處於計算狀態，故檢查各節點是否都處於閒置狀態，若是，則進入步驟 17，反之。則回到步驟 7。
17. 判斷所有組合皆已計算完畢，若是進入步驟 18，反之，則回到步驟 7。
18. 根據各節點計算各組合後回傳至伺服端之結果，找尋最佳的規劃結果。

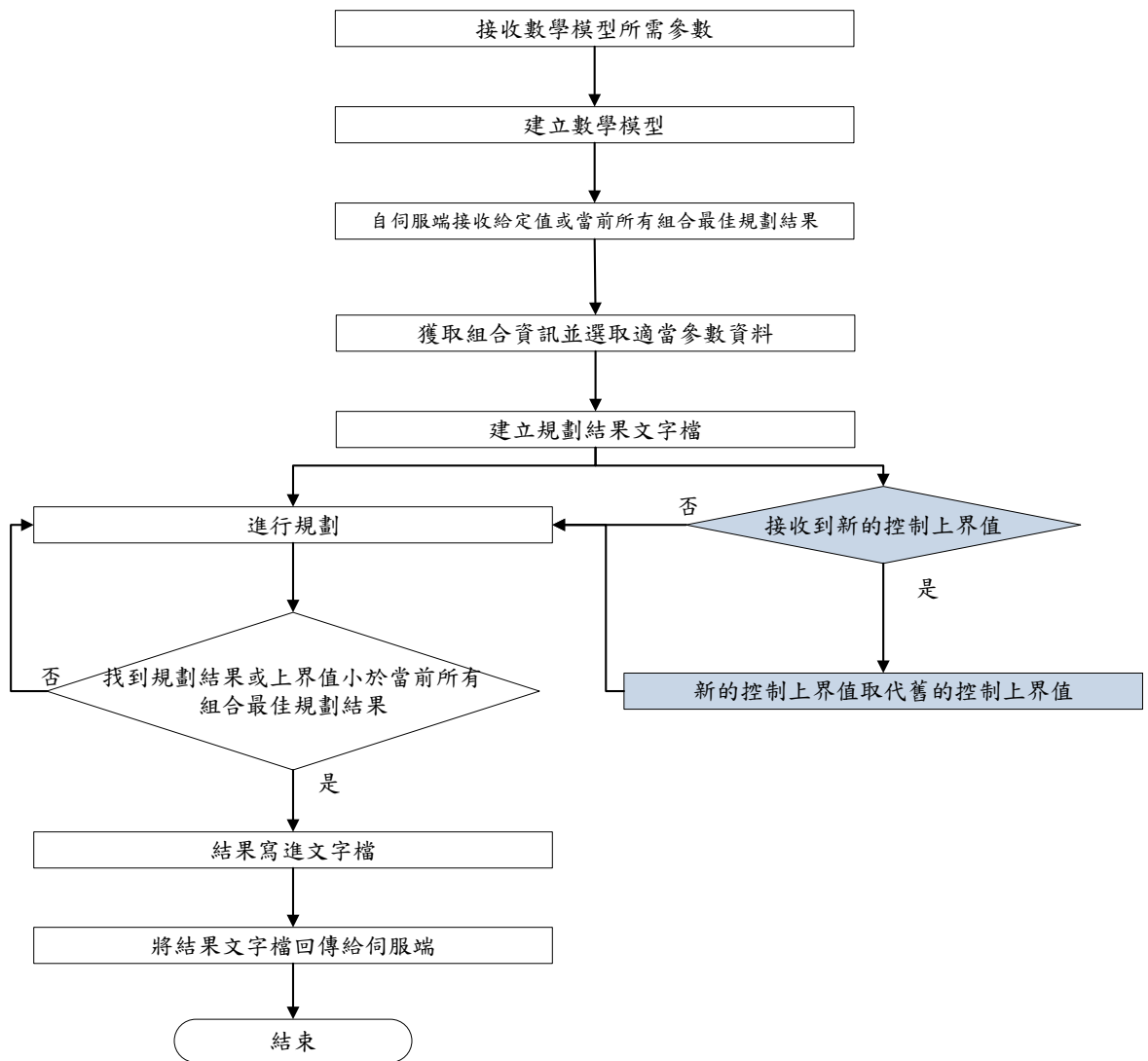


圖 3.18、計算結點計算流程(取得更佳目標函數值取代原始控制上界值)

上圖為計算節點自伺服器取得佳目標函數值後，取代原始控制上界值之計算流程，說明如下

1. 接收數學模型所需參數乃利用  $indexTransfer(String fName, byte[] fByte)$  方法物件接收自伺服器傳送之參數文字檔。
2. 建立數學模型乃利用  $modelConstruct(int nI, int nJ)$  方法物件接收自伺服器所傳遞 lng 檔案格式之數學模型供計算結點端 LINGO 讀取。
3.  $UBLimit(double PREV_Obj)$  物件接收從伺服器傳送的參數限制 *Upper Bound*。參數 *PREV\_Obj* 為初始設定值或截至當前所有組合最佳規劃結果。
4. 利用物件方法  $cominationTransfer(String COMBI_IJ)$  方法物件接收伺服

- 端所分配之組合資訊，並且依照分配之組合讀取對應之參數值。
5. 建立所求結果之文字檔檔案，其檔案名稱以 *cCount* 組合編號參數作為命名依據。而步驟 6 與步驟 7 為同步進行。
  6. LINGO 增加疊代(Iterations)求解數學模型，接著進入步驟 9。
  7. 計算結點 *UBLimit(double PREV\_Obj)*物件判別是否有接收伺服器端傳送更佳的目標值，若是，則進入步驟 8，若否，則進入步驟 6。
  8. *PREV\_Obj* 的值由原始之控制上界值改變為步驟 7 所接收的更佳目標值接著回到步驟 6。
  9. LINGO 是否計算完畢或者上界的值小於 *PREV\_Obj* 的值時，判別結果為是的話，進入步驟 10，反之則進入步驟 6。
  10. 利用 *modelSolve(int cCount)*方法物件將計算結果寫入文字檔。
  11. 利用 *modelSolve(int cCount)*方法物件將文字檔結果以陣列方式回傳至伺服器端。
  12. 計算結點計算完畢，等待伺服器端指令。



## 第四章 實驗與分析

本章將說明本研究進行之實驗，透過相同的實驗環境，使用不同的水準因子以及參數設定進行實驗。首先將對實驗環境、實驗方式進行說明，說明實驗中包含哪些因子以及參數設定，接著使用數理規劃軟體以及電腦進行實驗運算，最後針對實驗結果進行分析，在分析部分，除了針對因子及規模不同產生之不同結果進行說明外，另外會進行全域最佳規劃與分散式平行計算規劃之結果比較，並比較不同處理器個數對時間以及結果的影響。

### 4.1 實驗環境與參數設定

本節將分成三個部分，第一個部分將針對實驗將在何種環境下進行生產規劃進行說明，此部分包含供應鏈網絡結構以及處理器的規格。第二部分為實驗因子說明，本研究的實驗因子分成環境因子以及控制因子兩個部分，環境因子表示供應鏈網絡的規模大小，本研究分成大規模、中規模以及小規模；控制因子包含需求量、製造廠產能以及處理器個數三個因子。第三部分為參數設定

#### 4.1.1 實驗環境

1. 多階多廠區供應鏈網絡生產環境：本研究探討之供應鏈網絡生產環境如圖 4.1 所示，其中包含三個階層，而每個階層中又有「多廠」的狀況，包含多家供應商、多家製造廠、多家配銷中心。第一階層為供應商供應原物料給製造廠，供應商供給製造廠所需之原物料，本研究設定 BOM 表僅需印刷電路板以及記憶晶片兩個部分；第二階層為製造廠生產產品，此階層僅單純生產產品，並將成品運送至配銷中心，其生產產品的數量依據來自各配銷中心；第三階層為配銷中心進行分裝出貨給最終顧客。
2. 處理器之作業系統：進行本研究實驗之處理器作業系統為 Microsoft Windows XP Professional version 2002 SP3，CPU 為 Dual Core AMD Opteron (tm) Processor 1214, 2.19GHz, 2GB 的 RAM，並且使用數理規劃軟體 LINGO 10.0 的動態連結資料庫 (Dynamic Link Library, DLL) 套件搭配 JavaRMI 進行平行運算。

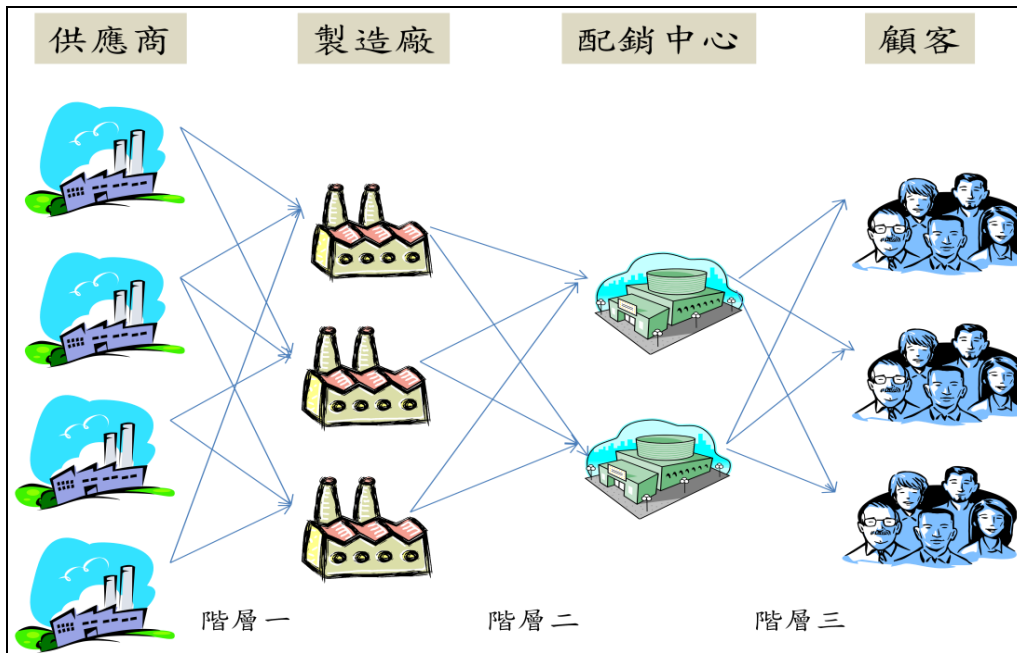


圖 4.1、多階多廠區供應鏈網絡示意圖

#### 4.1.2 實驗說明

本節分成環境規模以及實驗因子兩個部分，其中環境規模為供應鏈網絡的規模大小；而實驗因子包含需求量、製造廠產能以及處理器個數三個因子，以下將對於各項因子做說明。

1. 環境規模：本研究將供應鏈網絡規模分成大規模、中規模以及小規模三個水準，其中小規模供應鏈網絡包含 3 家供應商、3 家製造廠、3 家配銷中心以及 2 位顧客（訂單）；中規模供應商家數有 6 家、製造廠家數有 6 家、配銷中心家數有 4 家以及 6 位顧客（訂單）；而在進行小規模以及中規模實驗後，為證明本研究適用範圍可以擴大，故嘗試解決更大規模的供應鏈網絡，但因時間因素，此大規模實驗僅採取各因子部分水準實驗。此大規模供應鏈網絡生產環境中，供應商家數有 8 家、製造廠家數有 8 家、配銷中心家數有 6 家以及 8 位顧客（訂單）。

2. 實驗因子：實驗因子包含三個部分，分別如下

(1) 需求量因子：本研究之需求，設定為供應商產品種類最小總供給量的 60% 以及 80%，舉例而言，本研究供給原物料有印刷電路板以及記憶體晶片兩種，假設所有供應商能供給的印刷電路板以及記憶體晶片總產能分別為 200 及 100 單位，則需求量比較依據則取兩者較小者記憶體晶片 100 單位，故需求數量為 60 及 80 個。

- (2) 處理器個數因子：此因子分成三個水準，分別為高、中、低三水準。各水準對應到不同規模而有不同的處理器個數，本研究將小規模供應鏈網絡作為測試比較模型以及分散式平行計算結構正確與否的實驗，且考量其運算時間透過平行計算後時間已經降低許多，故小規模對應之處理器個數水準僅考量低水準 4 台。中規模處理器個數分成 4、8、16 台三種水準；大規模處理器個數則分成 8、16 及 32 台進行實驗。
- (3) 製造廠產能因子：此因子分成高水準因子以及低水準因子兩部分，低水準產能為製造廠原始產能，而高水準產能為製造廠原始產能之 2 倍。舉例而言，若製造廠之原始產能為 100 單位，則高水準製造廠產能則為 200 單位。

環境規模以及控制因子整理如表 4.1 及表 4.2

表 4.1、環境規模

環境規模	供應商家數	製造廠家數	配銷中心家數	顧客（訂單）數
小規模	3	3	3	2
中規模	6	6	4	6
大規模	8	8	6	8

表 4.2、實驗因子

控制因子	因子水準	說明
需求量	高	總需求量為最小原物料總供給量 80%
	低	總需求量為最小原物料總供給量 60%
處理器個數	高	中規模為 16 台、大規模為 32 台
	中	中規模為 8 台、大規模為 16 台
	低	小規模為 4 台、中規模為 4 台、大規模為 8 台
製造廠產能	高	2 倍低水準製造廠產能
	低	原始製造廠產能

### 4.1.3 參數設定

表 4.3 為輸入參數之整理，參數包含各節點數量及時間資訊以及訂單訊息等，而本研究假設各節點間各項參數設定均相同，雖模式容易改變，唯有因原物料種類不同而產生部分差異，且各參數值並未因規模大小而有所改變。

表 4.3、參數值及其代表意義

符號	參數說明	數值	
$c_{rij}^{ST}$	原物料 r 自供應商 i 運送至製造廠 k 之運輸成本	r=1	r=2
		2	2
$cap_{ri}^S$	供應商 i 供給原物料 r 之產能限制	r=1	r=2
		106	230
$t_{ri}^{SP}$	供應商 i 對原物料 r 的生產前置時間	r=1	r=2
		3	2
$t_{ij}^{RT}$	供應商 i 至製造廠 j 之運輸時間	3	
$c_{rij}^B$	製造廠 j 向供應商 i 購買原物料 r 之單位成本	r=1	r=2
		2	5
$c_{sjk}^P$	製造廠 j 生產半成品 s 至配銷中心 k 之生產成本	10	
$c_{sjk}^{FT}$	製造廠 j 運送至配銷中心 k 之運輸成本	2	
$c_r^{RI}$	製造廠原物料 r 之存貨成本	20	
$c_s^{SI}$	製造廠半成品 s 之存貨成本	36	
$t_{jk}^{FD}$	製造廠 j 至配銷中心 k 之運輸時間	2	

表 4.3、參數值及其代表意義（續）

符號	參數說明	數值	
$cap_{sj}^F$	製造廠 j 生產半成品 s 之產能限制	141	
$t_j^P$	製造廠 j 之生產時間	3	
$t_k^{SHIP}$	配銷中心 k 之出貨時間	36	
$c_{pkl}^{DT}$	配銷中心 k 運送至顧客 l 之運輸成本	5	
$c_{pkl}^A$	配銷中心 k 組裝成品 p 給顧客 l 之組裝成本	20	
$c_k^{DI}$	配銷中心 k 之存貨成本	45	
$cap_{pk}^D$	配銷中心 k 裝配成品 p 之產能限制	320	
$q_{pl}^R$	顧客 l 購買產品 p 之數量	*註一	
$e_{pl}^R$	顧客 l 購買產品 p 之單位售價	300	
$c_{pl}^{SH}$	顧客 l 成品 p 之缺貨懲罰成本	494	
$B_{rs}^{RB}$	組成半成品 s 所需要原物料 r 之個數	r=1	r=2
		1	2

\*註一：因需求量隨水準高低而變動，將於 4.2 節進行分項說明。

## 4.2 實驗設計及分析

本結將針對小規模以及中規模實驗分別說明全域規劃以及平行計算之結果，並且對於結果間之差異進行比較。首先從小規模實驗開始，利用小規模運算全域最佳解，接著進行分散式平行計算，將兩者結果進行比對驗證，證明本研究之數學模型無誤。小規模且在考量分散式平行計算後，其運算時間所減少的幅度有限，故於小規模分散式平行運算處理器個數僅考量 4 台。最後為證明本研究之數學模型以及分散式平行計算方法可解決更大規模之問題，故嘗試使用更多處理器平行計算之，而在考量時間因素下，其分散式平行計算僅計算不同處理器數目對控制因子中需求量以及製造廠產能兩因子中的低水準因子。

### 4.2.1 小規模全域最佳解

本研究利用小規模對數學模型以及平行運算進行驗證，並且小規模的各組實驗皆進行三次實驗，以觀察在規模及數據間皆相同的狀況下，其規劃時間是否有差異。而本小節進行在不同製造廠產能以及需求量總共四種因子組合，組合編號及其實際數量整理如表 4.4

表 4.4、小規模組合製造廠及顧客節點對應數量表

編號	製造廠產能倍率	製造廠節點 j 之數量			需求量百分比	顧客節點 l 需求量	
		j=1	j=2	j=3		l=1	l=2
1	2	281	281	281	60	95	96
2	1	141	141	141	60	95	96
3	2	281	281	281	80	127	128
4	1	141	141	141	80	127	128

#### 1. 小規模 2 倍製造廠產能，60% 需求量（如表 4.4，以編號一實驗代表）

以 LINGO 進行小規模 2 倍製造廠產能，60% 需求量第一次全域最佳規劃，其結果整理如圖 4.2，說明如下：

供應商節點一運送印刷電路板（以下稱之為原物料一）85 單位以及記憶體晶片（以下稱之為原物料二）170 單位至製造廠節點二；供應商節點二運送原物料一 106 單位以及原物料二 212 單位至製造廠節點一；供應商

節點三無運輸；製造廠節點一總共生產記憶體模組（以下稱之為產品）106單位，其中96單位運送至配銷中心節點二，10單位運送至配銷中心節點三；而製造廠節點二生產85單位產品並送至配銷中心節點一；配銷節點一與配銷節點三總共運送95單位產品至顧客節點一，配銷中心節點二運送96單位至顧客節點二。

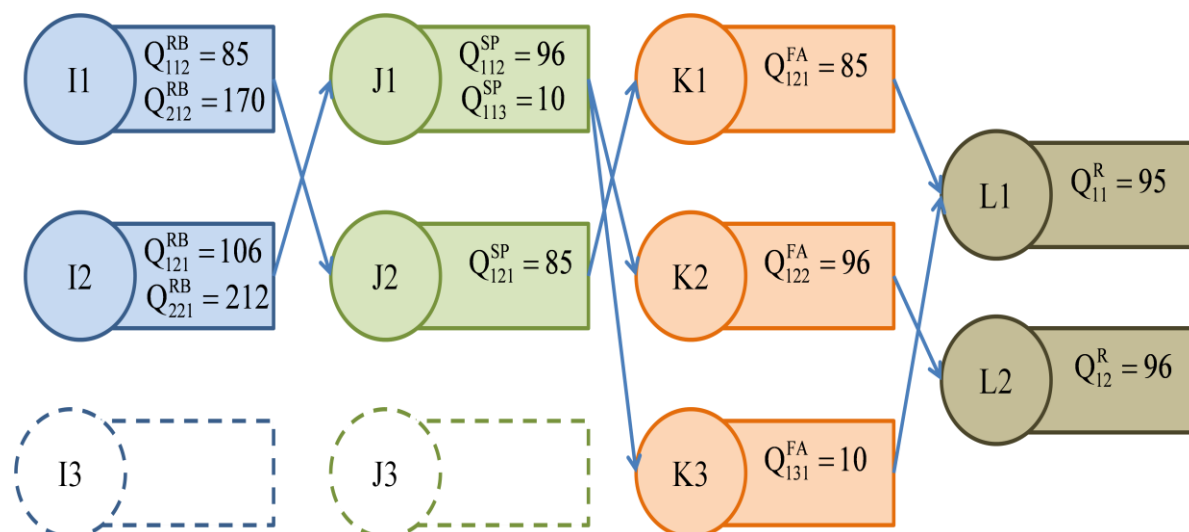


圖 4.2、小規模編號一實驗第一次、第二次實驗全域最佳解規劃運輸分配

以 LINGO 進行小規模 2 倍製造廠產能，60% 需求量第二次全域最佳規劃，運輸量分配結果與第一次實驗相同，唯有全域最佳解規劃時間有所差異。其結果整理如同圖 4.2 與規劃結果說明如下：

供應商節點一運送原物料一 85 單位以及原物料二 170 單位至製造廠節點二；供應商節點二運送原物料一 106 單位以及原物料二 212 單位至製造廠節點一；供應商節點三無運輸；製造廠節點一總共生產產品 106 單位，其中 96 單位運送至配銷中心節點二，10 單位運送至配銷中心節點三；而製造廠節點二生產 85 單位產品並送至配銷中心節點一；配銷節點一與配銷節點三總共運送 95 單位產品至顧客節點一，配銷中心節點二運送 96 單位至顧客節點二。

以 LINGO 進行小規模 2 倍製造廠產能，60% 需求量第三次全域最佳規劃，其結果整理如圖 4.3，本次實驗結果運輸數量與前兩次實驗相同，但其運輸路徑有所改變，產生多重最佳解的狀況，其說明如下：

供應商節點一運送原物料一 85 單位以及原物料二 170 單位至製造廠節點三；供應商節點二無運輸；供應商節點三運送原物料一 106 單位以及原

物料二 212 單位至製造廠節點一；製造廠節點一總共生產產品 106 單位，其中 96 單位運送至配銷中心節點二，10 單位運送至配銷中心節點三；而製造廠節點二生產 85 單位產品並送至配銷中心節點一；配銷節點一與配銷節點三總共運送 95 單位產品至顧客節點一，配銷中心節點二運送 96 單位至顧客節點二。

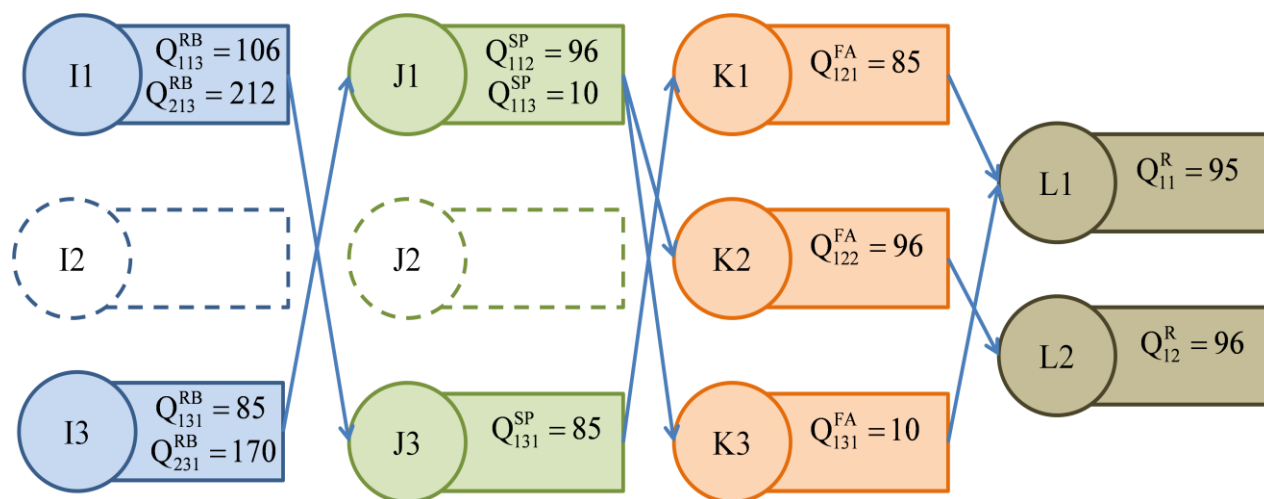


圖 4.3、小規模編號一實驗第三次實驗全域最佳解規劃運輸分配

本次實驗之運輸數量與前兩次之結果相同，而運輸路徑不盡相同，產生多重最佳解的狀況，推測是因為起始各廠區間的輸入參數皆相同所致，然而隨其參數模式改變，其結果將隨之改變，但其運輸數量、淨利等結果皆無明顯改變，且本研究將著重於平行計算前後之運輸量、淨利及各項成本結果比較，故對於因參數模式改變而產生多重運輸路徑之狀況不予贅述。統整實驗結果發現，包含目標函數總營收、各項成本與淨利，三次實驗結果皆相同，整理如表 4.5，唯獨在規劃時間上有所不同，規劃時間整理於表 4.6

表 4.5、小規模編號一實驗三次實驗目標函數規劃結果

編號	說明	結果	編號	說明	結果
1	總營收	57300	8	製造廠→配銷中心運輸成本	382
2	採購成本	2292	9	配銷中心→顧客運輸成本	955
3	製造成本	1910	10	供應商運輸批量成本	60
4	分裝成本	3820	11	製造廠運輸批量成本	45



表 4.5、小規模編號一實驗三次實驗目標函數規劃結果（續）

編號	說明	結果	編號	說明	結果
5	製造廠存貨持有成本	0	12	配銷中心運輸批量成本	45
6	配銷中心存貨持有成本	0	13	總淨利	46645
7	供應商→製造廠運輸成本	1146			

表 4.6、小規模編號一實驗三次實驗規劃時間

規劃實驗序次	第一次	第二次	第三次
規劃時間（秒）	3805	3399	3567

## 2. 小規模原始製造廠產能，60%需求量（如表 4.4，以編號二實驗代表）

小規模原始製造廠產能，60%需求量依然進行三次實驗，實驗結果分別如下說明：

以 LINGO 進行小規模原始製造廠產能，60%需求量第一次全域最佳規劃，其結果整理如圖 4.4：

供應商節點一運送原物料一 106 單位以及原物料二 212 單位至製造廠節點二；供應商節點二運送原物料二 170 單位至製造廠節點三；供應商節點三運送原物料一 85 單位至製造廠節點三；製造廠節點二總共生產產品 106 單位，其中 96 單位運送至配銷中心節點一，10 單位運送至配銷中心節點三；而製造廠節點三生產 85 單位產品至配銷中心節點二；配銷中心節點一運送 96 單位產品至顧客節點二；配銷節點二與配銷節點三總共運送 95 單位產品至顧客節點一。

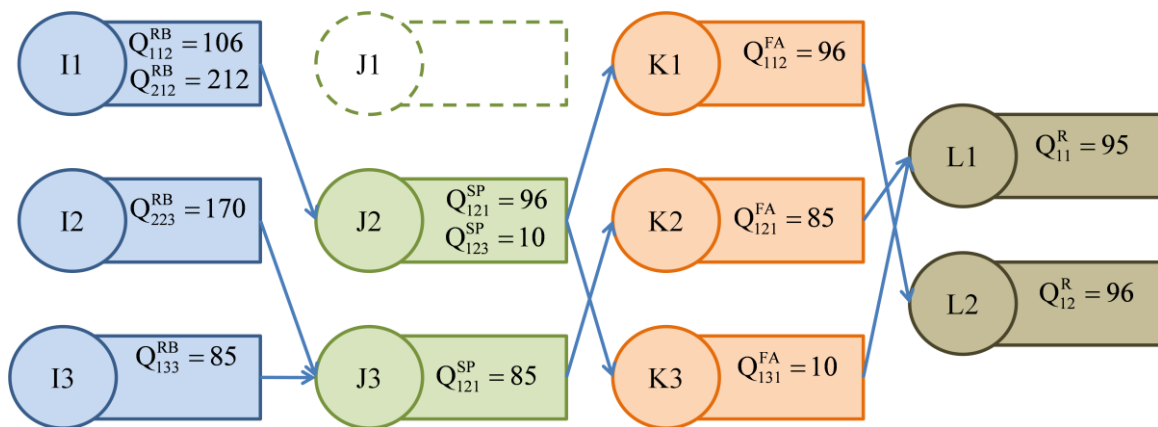


圖 4.4、小規模編號二實驗第一次實驗全域最佳解規劃運輸分配。

以 LINGO 進行小規模原始製造廠產能，60% 需求量第二次全域最佳規劃，其結果整理如圖 4.5：

供應商節點一運送原物料二 170 單位至製造廠節點三；供應商節點二運送原物料一 106 單位及原物料二 212 單位至製造廠節點一；供應商節點三運送原物料一 85 單位至製造廠節點三；製造廠節點一總共生產產品 106 單位，其中 96 單位運送至配銷中心節點一，10 單位運送至配銷中心節點二；而製造廠節點三生產 85 單位產品至配銷中心節點三；配銷中心節點一運送 96 單位產品至顧客節點二；配銷節點二與配銷節點三總共運送 95 單位產品至顧客節點一。

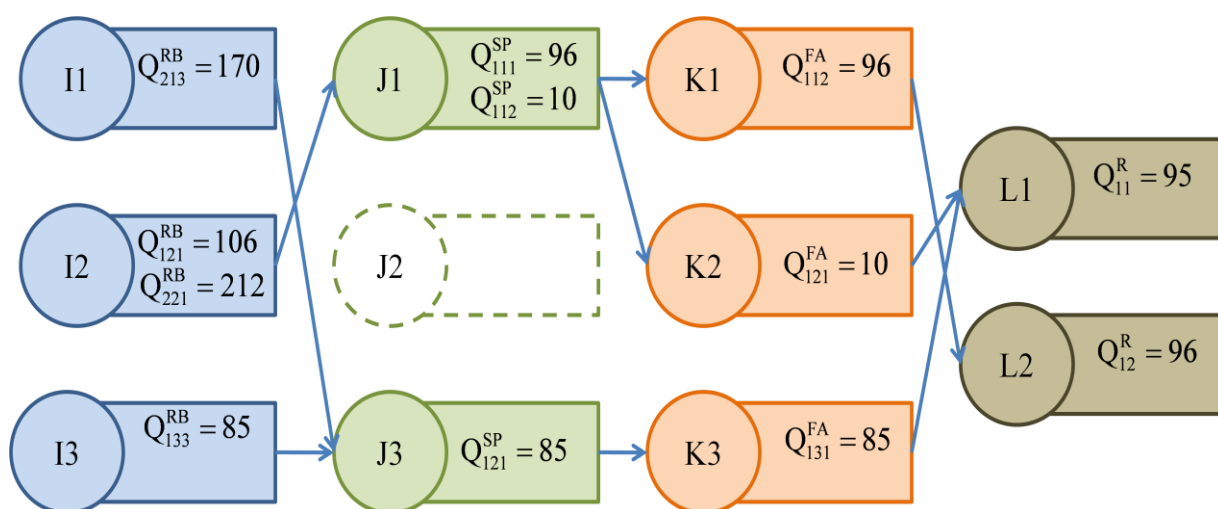


圖 4.5、小規模編號二實驗第二次實驗全域最佳解規劃運輸分配。

以 LINGO 進行小規模原始製造廠產能，60% 需求量第三次全域最佳規劃，規劃運輸數量及路徑如圖 4.6：

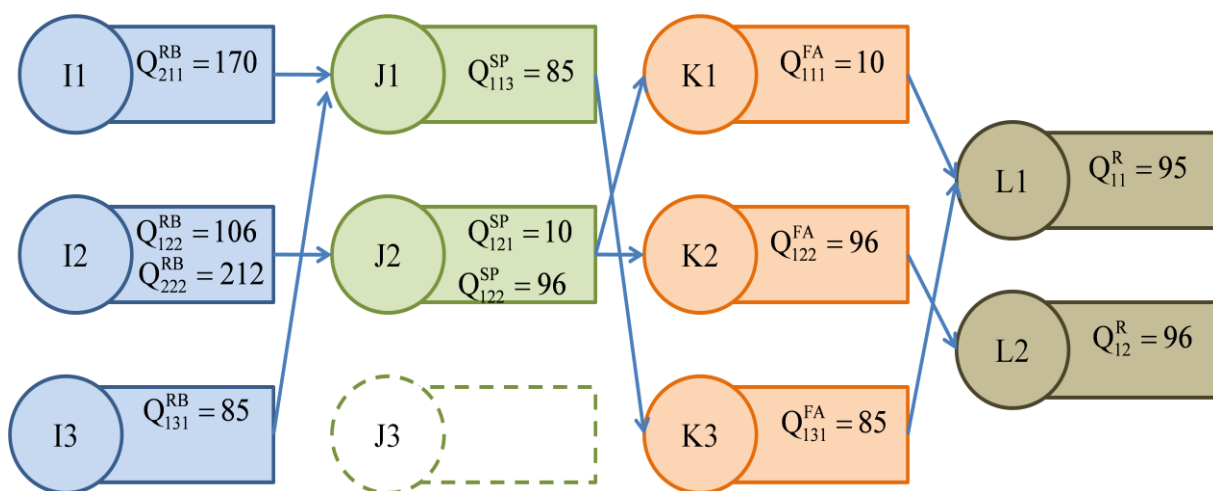


圖 4.6、小規模編號二實驗第三次實驗全域最佳解規劃運輸分配

供應商節點一運送原物料二 170 單位至製造廠節點一；供應商節點二運送原物料一 106 單位及原物料二 212 單位至製造廠節點二；供應商節點三運送原物料一 85 單位至製造廠節點一；製造廠節點二生產產品 85 單位，送至配銷中心節點三；製造廠節點二總共生產產品 106 單位，其中 10 單位運送至配銷中心節點一，96 單位運送至配銷中心節點二；而製造廠節點三在本次實驗沒有進行生產；配銷中心節點一與配銷節點三總共運送 95 單位產品至顧客節點一；而配銷節點二運送 96 單位產品至顧客節點二。

小規模原始製造廠產能，60% 需求量三次實驗目標式各項結果皆相同，並且與小規模 2 倍製造廠產能，60% 需求量相同，整理如表 4.7。而在規劃時間上三次實驗有所不同，三次實驗規劃時間整理如表 4.8

表 4.7、小規模編號二實驗三次目標函數規劃結果

編號	說明	結果	編號	說明	結果
1	總營收	57300	8	製造廠→配銷中心運輸成本	382
2	採購成本	2292	9	配銷中心→顧客運輸成本	955
3	製造成本	1910	10	供應商運輸批量成本	60
4	分裝成本	3820	11	製造廠運輸批量成本	45
5	製造廠存貨持有成本	0	12	配銷中心運輸批量成本	45
6	配銷中心存貨持有成本	0	13	總淨利	46645
7	供應商→製造廠運輸成本	1146			

表 4.8、小規模編號二實驗三次實驗規劃時間

規劃實驗序次	第一次	第二次	第三次
規劃時間（秒）	4137	3985	4539

### 3. 小規模 2 倍製造廠產能，80% 需求量（如表 4.4，以編號三實驗代表）

小規模 2 倍製造廠產能，80% 需求量如同前述兩小規模進行三次全域規劃實驗，經過實際實驗三次以及前述實驗發現，三次實驗結果目標式各項數值皆無改變，僅有在運輸路徑有所差異，產生多重最佳解，故以下對於各階層間的運輸路徑以及運輸數量不再一一以圖示化方式呈現，僅呈現最終規劃結果以及各次實驗所花費的時間。本實驗之需求水準增加，需求為供應節點產能的 80%，一共 255 單位，將顧客節點的需求量分成 127 及

128 單位兩部份，分別由三家供應商、製造廠與配銷中心供應，總營收 76500，總成本 15060，整理如表 4.9；規劃時間如表 4.10

表 4.9、小規模編號三實驗三次目標函數規劃結果

編號	說明	結果	編號	說明	結果
1	總營收	76500	8	製造廠→配銷中心運輸成本	510
2	採購成本	3825	9	配銷中心→顧客運輸成本	1275
3	製造成本	2550	10	供應商運輸批量成本	120
4	分裝成本	5100	11	製造廠運輸批量成本	75
5	製造廠存貨持有成本	0	12	配銷中心運輸批量成本	75
6	配銷中心存貨持有成本	0	13	總淨利	61440
7	供應商→製造廠運輸成本	1530			

表 4.10、小規模編號三實驗三次實驗規劃時間

規劃實驗序次	第一次	第二次	第三次
規劃時間（秒）	5965	6218	6532

#### 4. 小規模 1 倍製造廠產能，80% 需求量（如表 4.4，以編號四實驗代表）

小規模 1 倍製造廠產能，80% 需求量三次實驗之目標式總營收與各項成本皆相同，僅有各運輸路徑以及運輸數量的差異，並且其結果與前述小規模編號三實驗相同，如表 4.9 所示，唯獨規劃時間有所差異，整理如下表。

表 4.11、小規模編號四實驗三次實驗規劃時間

規劃實驗序次	第一次	第二次	第三次
規劃時間（秒）	12941	9964	10252

透過小規模四種不同組合，每種組合進行三次實驗發現，各種組合間實驗的目標函數值不會有任何改變，僅有各節點的運輸數量以及路徑有所差異，造成多重最佳解的原因推測乃因各節點間的各项輸入參數皆相同所致。而在規劃時間上，表 4.12 為各組合實驗的計算時間

表 4.12、小規模各水準單一處理器全域最佳規劃時間

實驗 編號	製造廠產 能倍率	需求量百分 比	規劃實驗序次與規劃時間（秒）			平均 時間
			第一次	第二次	第三次	
一	2	60	3805	3399	3567	3590
二	1	60	4137	3985	4539	4220
三	2	80	5965	6218	6532	6238
四	1	80	12941	9964	10252	11052

綜合小規模所有組合實驗規劃時間發現，當需求量百分比固定為 60% 時，若製造廠產能倍率為 1，其規劃平均時間將會增加 17.55%；當需求量百分比固定為 80%，製造廠產能限縮造成規劃時間增加幅度達 77.17%；又若當製造廠產能倍率固定為 2 時，需求量自 60% 增加至 80%，其規劃平均時間增加 73.76%；當製造廠產能倍率固定為 1 時，因需求量增加而平均規劃時間增加幅度達 161.90%。由上列數據顯示，本研究之規劃時間會隨需求百分比增加與製造廠產能限縮而增加，且增加幅度皆相當明顯。且經過實驗結果顯示，各項運輸量、運輸路徑、時間皆在合理的範圍內，也證明本研究之數學模型之正確性，此後再進行分散式平行計算以及擴大規模實驗，比對各實驗總淨利以及規劃時間之消長。

#### 4.2.2 小規模平行計算規劃結果

小規模全域最佳解四種實驗組合皆分別進行三次，本節亦針對四種不同組合進行三次實驗，並且考量到處理器個數，而處理器個數以 4 台為基準，若有需要則以倍數調整之。表 4.13 為小規模組合製造廠、顧客節點及處理器個數對應表，其中在進行分散式平行計算前，先進行供應商家數以及製造廠家數的選擇，而在此選擇的條件為最小滿足需求量之供給家數，舉小規模 2 倍製造廠產能，需求量 60% 而言，供應商可供給兩種產品為原物料一及原物料二，其單一供應商產能分別為 106 單位與 230 單位，以小規模而言供應商家數有三家，則供應商之種類最小供給總數目為 318 單位（ $106 \times 3 = 318$ ），而需求為其 60%，總需求為 191 單位。因總需求為 191 單位，故於製造廠以及供應商的供給產能必須大於等於需求，則至少需

要製造廠家數為 1 家（單一製造廠產能為 281 單位）、供應商家數為 2 家（單一供應商種類最小產能為 106 單位）。

表 4.13、小規模各水準對應選取家數表\*註二

編號	製造廠產能倍率	單一製造廠節點數量	需求量百分比	顧客節點總需求量	供應商選取家數	製造廠選取家數
1	2	281	60	191	2	1
2	1	141	60	191	2	2
3	2	281	80	255	3	1
4	1	141	80	255	3	2

\*註二：參與規劃之供應商與製造廠各三家

以下對於各組合進行三次實驗，比較三次實驗單一處理器與分散式平行計算在最終淨利以及整體規劃時間之差異：

### 1. 小規模 2 倍製造廠產能，60% 需求量

表 4.14、小規模 2 倍製造廠產能、60% 需求量單一處理器與多處理器比較

處理器個數	規劃實驗序次與規劃時間（秒）			平均規劃時間	淨利
	第一次	第二次	第三次		
1	3805	3399	3567	3590	46645
4	335	327	329	330	46645
差異百分比	91.20	90.38	90.78	90.81	0

小規模 2 倍製造廠產能、60% 需求量平行計算結果如表 4.14，經過三次實驗結果顯示，透過四台處理器進行平行計算，所求得之淨利與使用單一處理器進行規劃相同，並無差異，但每次實驗的規劃時間皆有顯著的減少，而使用的規劃時間平均節省 90.81%。

實驗設計原先預計在小規模中處理器個數提升至八個處理器進行實驗，但經過四台處理器之平行計算後發現，使用四台處理器已將規劃計算時間壓縮在六分鐘以內，對於處理器水準提升，認為無其必要行，待規模擴大

後，再提升處理器水準進行比較。

## 2. 小規模 1 倍製造廠產能，60% 需求量

表 4.15、小規模 1 倍製造廠產能、60% 需求量單一處理器與多處理器比較

處理器個數	規劃實驗序次與規劃時間（秒）			平均規劃時間	淨利
	第一次	第二次	第三次		
1	4137	3985	4539	4220	46645
4	439	451	449	446	46645
差異百分比	89.39	88.68	90.11	89.43	0

小規模 1 倍製造廠產能、60% 需求量平行計算結果如表 4.15，經過四台處理器進行平行計算，選取供應商及製造廠各兩家，其淨利與使用單一處理器進行規劃相同，但每次實驗的規劃時間皆有顯著的減少，而使用的規劃時間平均節省 89.43%。

## 3. 小規模 2 倍製造廠產能，80% 需求量

表 4.16、小規模 2 倍製造廠產能、80% 需求量單一處理器與多處理器比較

處理器個數	規劃實驗序次與規劃時間（秒）			平均規劃時間	淨利
	第一次	第二次	第三次		
1	5965	6218	6532	6238	61440
4	538	539	533	536	61440
差異百分比	90.98	91.33	91.84	91.41	0

小規模 2 倍製造廠產能、80% 需求量平行計算結果如表 4.16，選取供應商三家、製造廠一家，經過四台處理器進行平行計算，其淨利與使用單一處理器進行規劃相同，但每次實驗的規劃時間皆有顯著的減少，而使用的規劃時間平均節省 91.41%。

## 4. 小規模 1 倍製造廠產能，80% 需求量

小規模 1 倍製造廠產能、80% 需求量平行計算結果如表 4.17，經過選取三家供應商兩家製造廠之平行計算後，淨利與使用單一處理器進行規劃相同，但每次實驗的規劃時間皆有顯著的減少，而使用的規劃時間平均節省

87.24%。

表 4.17、小規模 1 倍製造廠產能、80% 需求量單一處理器與多處理器比較

處理器個數	規劃實驗序次與規劃時間（秒）			平均規劃時間	淨利
	第一次	第二次	第三次		
1	12941	9964	10252	11052	61440
4	1317	1270	1643	1410	61440
差異百分比	89.82	87.25	83.97	87.24	0

上述四種小規模不同組合之實驗，進行分散式平行計算對於規劃時間有明顯的改善，時間的減少量皆有將近 90% 的減少，且其規劃結果（淨利）與全域最佳解完全相同，由此顯示，若使用分散式平行計算既可達到規劃出與全域最佳解相同結果，亦達到預期節省規劃時間的目的。

#### 4.2.3 中規模全域最佳解

本節介紹將環境因子水準提高至中規模之實驗，因小規模各組中各次實驗，無極端的規劃時間值出現，故考量時間因素，各組實驗僅進行一次實驗。以下為四種不同水準的實驗結果。下表為中規模各水準所對應的單一製造廠產能以及總需求量。

表 4.18、中規模各水準對應產能與總需求量表

編號	製造廠產能倍率	單一製造廠節點數量	需求量百分比	顧客節點總需求量
1	2	281	60	382
2	1	141	60	382
3	2	281	80	509
4	1	141	80	509

##### 1. 中規模 2 倍製造廠產能，60% 需求量

本實驗透過 LINGO 軟體進行規劃，圖 4.7 為 LINGO 的規劃畫面，圖中顯示之 BsetObj 為規劃下界，而 Obj Bound 為規劃上界，當規劃上下界數值相同時，即為最佳規劃，然而實驗經過約 28 小時的規劃後，仍無最佳解的



結果出現，僅有區域最佳解（Local Optimal），而 local optimal 的值即為 BestObj 之值。

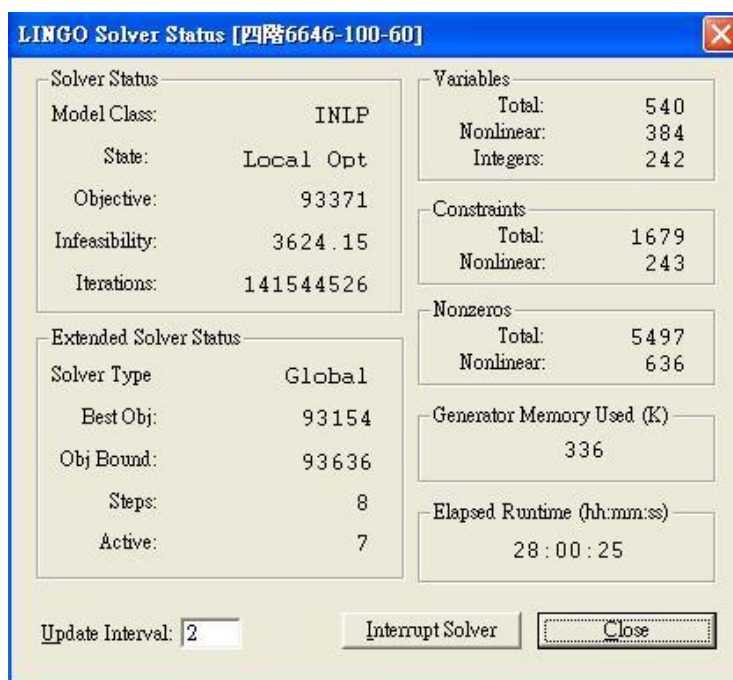


圖 4.7、中規模 2 倍製造廠產能，60% 需求量之 LINGO 全域規劃結果

因無法求得最佳規劃結果，且於實驗過程中觀察 BestObj 收斂情況發生如圖 4.8 之狀況，下界實際上會很快速上升到某一數值，但在此之後以緩慢的速度向上收斂，而上界的部分也是以緩慢的速度向下收斂。

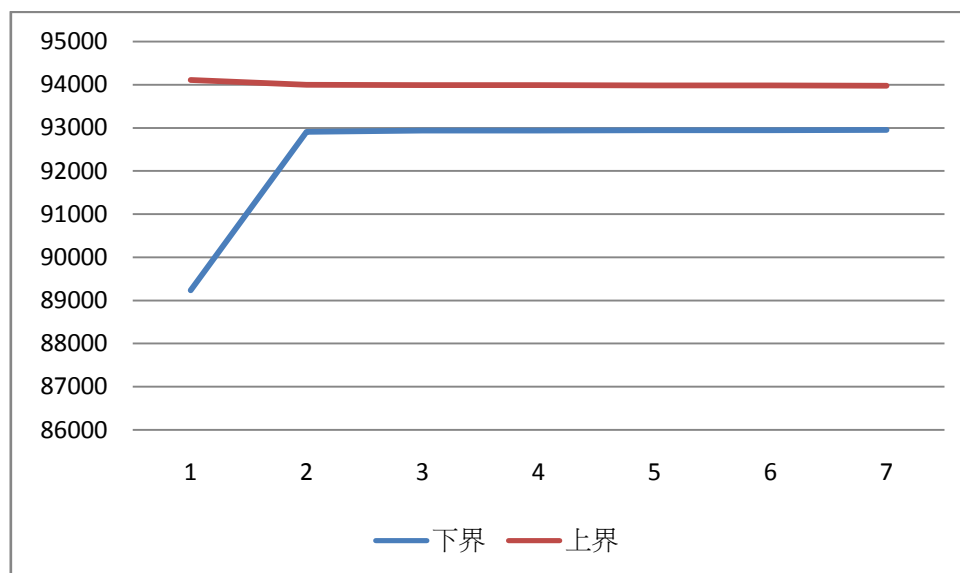


圖 4.8、中規模 LINGO 全域規劃上界與下界收斂狀況

產生如圖 4.8 之狀況時，對於目標函數值與規劃時間之間，就有必要做取捨。因為倘若花費在計算的時間上非常的長，但目標函數值增加幅度卻

是有限的，可能就得考慮以時間換取目標函數值。以下將分別比較規劃時間長短其目標函數值改變差異。我們將相同實驗環境、輸入參數分別進行規劃 8 小時、10 小時以及約 28 小時（即前述實驗，為隨機進行）的規劃，而之所以採用 8 小時與 10 小時做比較乃是以約一日之工作時間做為考量，但實際上要採用多少小時並無特定設定規則可循，故亦可比較在其他時間下，規劃結果與時間上的變化差異。本研究比較分別之目標函數值整理如下表：

表 4.19、中規模 2 倍製造廠產能，60% 需求量規劃時間與淨利增加變化

	規劃時間（秒）		
	28800 （8 小時）	36000 （10 小時）	100800 （28 小時）
規劃淨利*註三	90150	90154	93154
規劃時間增加百分比（與 8 小時相比）	—	25%	250%
規劃淨利增加百分比（與 8 小時相比）	—	0.004%	3.332%
規劃時間增加百分比（與 10 小時相比）	—	—	180%
規劃淨利增加百分比（與 10 小時相比）	—	—	3.333%

註三：三次實驗結果皆非全域最佳解，皆為規劃時間內之下界值

從以上結果來看，以 8 小時為比較基準，時間增加 25%，規劃淨利僅僅增加 0.004%；若時間增加為 2.5 倍，淨利也僅增加 3.332%。若比較 10 小時與 28 小時，時間增加 1.8 倍，規劃淨利也僅增加 3.333%。由數據來看發現花費 8 小時與花費 10 小時，淨利結果差異不大，建議採用 8 小時規劃結果會更具效率，其中 8 小時是採用一般正常工作時數。

## 2. 中規模 1 倍製造廠產能，60% 需求量

如圖 4.9，中規模 1 倍製造廠產能，60% 需求量進行約 30 小時得全域最佳規劃後，亦無最佳解出現。表 4.20 也將針對 8 小時、10 小時以及 30 小時的規劃結果進行時間與淨利比較，結果顯示，無論是規劃時間自 8 小時增加至 10 小時，或是由 10 小時增加至 30 小時，規劃總淨利增加幅度皆不到 0.2 個百分比。由此推論，與其花費冗長的規劃時間，建議選擇較短的

規劃時間-8 小時（考量以一天正常工時作為計算依據），因其與花費長時間進行規劃的結果相去不遠。



圖 4.9、中規模 1 倍製造廠產能，60% 需求量之 LINGO 全域規劃結果

表 4.20、中規模 1 倍製造廠產能，60% 需求量規劃時間與淨利增加變化

	規劃時間（秒）		
	28800 (8 小時)	36000 (10 小時)	108000 (30 小時)
規劃淨利*註三	93023	93076	93152
規劃時間增加百分比（與 8 小時相比）	—	25%	275%
規劃淨利增加百分比（與 8 小時相比）	—	0.057%	0.139%
規劃時間增加百分比（與 10 小時相比）	—	—	200%
規劃淨利增加百分比（與 10 小時相比）	—	—	0.082%

### 3. 中規模 2 倍製造廠產能，80% 需求量

接著提高需求量水準，進行中規模 2 倍製造廠產能，80% 需求量實驗，圖 4.10 為 LINGO 進行規劃之畫面，結果也說明經過約 27 小時的規劃，並無最佳解產生。其中下界 BestObj，也就是 local optimal 值為 124367，而上界 Obj Bound 為 124447。因為無最佳解，進而討論縮短規劃時間其規劃結

果之差異情形。

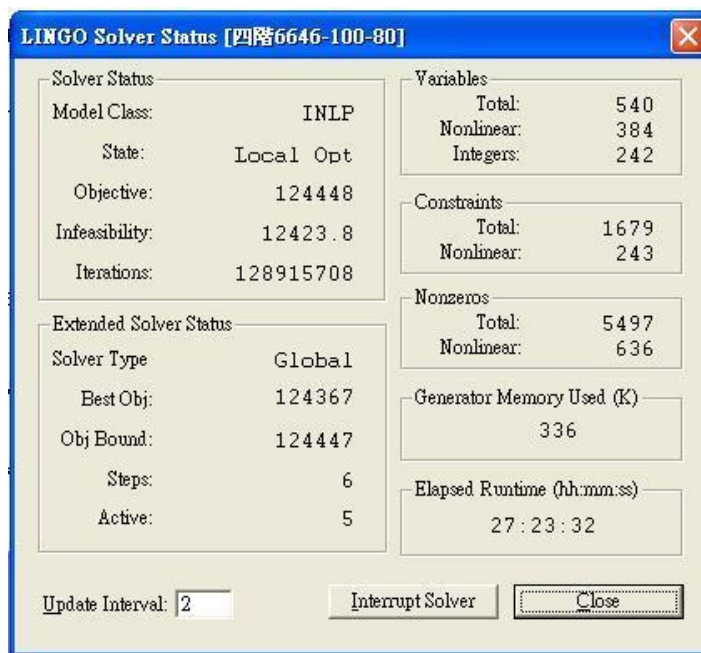


圖 4.10、中規模 2 倍製造廠產能，80% 需求量之 LINGO 全域規劃結果

表 4.21 為三種不同規劃時間長度之規劃結果與比較，三次皆無求出最佳規劃結果，經過 8 小時與 10 小時的規劃後，其淨利無任何變化，皆為 120498；而進行約 27 小時後，淨利增加百分比為 2.827，與 8 小時比較狀況下，時間增加百分比為 237.5。

表 4.21、中規模 2 倍製造廠產能，80% 需求量規劃時間與淨利增加變化

	規劃時間 (秒)		
	28800 (8 小時)	36000 (10 小時)	97200 (27 小時)
規劃淨利* <sup>註三</sup>	120948	120948	124367
規劃時間增加百分比 (與 8 小時相比)	—	25%	237.5%
規劃淨利增加百分比 (與 8 小時相比)	—	0%	2.827%
規劃時間增加百分比(與 10 小時相比)	—	—	170%
規劃淨利增加百分比(與 10 小時相比)	—	—	2.827%

#### 4. 中規模 1 倍製造廠產能，80% 需求量

最後一組實驗為中規模 1 倍製造廠產能，80% 需求量，本組全域最佳規

劃亦無求出最佳解，經過約 28 小時的規劃後，下界值為 124330，上界值為 124628。表 4.22 為此實驗考量不同時間的實驗結果，進行 8 小時規劃後得到的結果為 120935，10 小時規劃結果為 120951，兩者間的增加幅度與前述三實驗類似，幅度不大，僅 0.0132%；若 8 小時與 28 小時結果比較，淨利增加幅度提升至 2.782%，但增加的時間百分比卻是 250%。

表 4.22、中規模 1 倍製造廠產能，80% 需求量規劃時間與淨利增加變化

	規劃時間 (秒)		
	28800 (8 小時)	36000 (10 小時)	100800 (28 小時)
規劃淨利* <sup>註三</sup>	120935	120951	124300
規劃時間增加百分比 (與 8 小時相比)	—	25%	250%
規劃淨利增加百分比 (與 8 小時相比)	—	0.0132%	2.782%
規劃時間增加百分比(與 10 小時相比)	—	—	180%
規劃淨利增加百分比(與 10 小時相比)	—	—	2.769%

綜合上列四組實驗，若以 8 小時作為比較基準，時間增加比例對應到淨利成長幅度都有如圖 4.11 之趨勢：

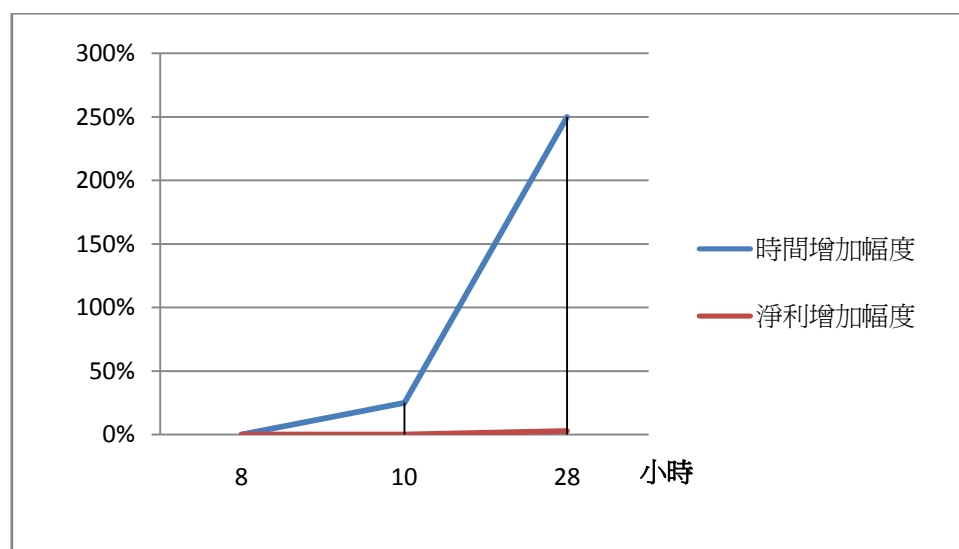


圖 4.11、時間-淨利成長幅度圖

圖 4.11 顯示，中規模之淨利成長幅度並未與規劃時間呈正向比例，甚至變化幅度甚小，故建議在規劃達一定時間，即可停止規劃結果，因花費

更長時間於本研究中規模全域最佳解規劃上，可能較不符合效益。故本研究僅以約一日工作時間作為規劃時間，對於無法於 8 小時內完成全域最佳規劃者，接著進行 10 小時的規劃實驗，比較 8 小時與 10 小時規劃總淨利的增加幅度，就中規模實驗結果顯示，無論規劃時間設定為 10 小時，抑或是約 28 小時，其淨利結果與規劃 8 小時者差異不顯著，故本研究建議採用規劃 8 小時之單一處理器規劃結果即可。

#### 4.2.4 中規模平行計算規劃結果

平行計算將對於處理器個數做不同的調整，中規模的處理器個數水準有 4 台、8 台與 16 台三種水準，如同前一小節，若規劃時間超過 8 小時者，僅討論 8 小時以及 10 小時兩者結果之差異。又如同小規模分散式平行計算，在進行平行計算前，須先選取供應商家數以及製造廠家數，表 4.23 為中規模個水準對應之最小選取家數

表 4.23、中規模各水準對應選取家數表<sup>\*註四</sup>

編號	製造廠產能倍率	單一製造廠節點數量	需求量百分比	顧客節點總需求量	供應商選取家數	製造廠選取家數
1	2	281	60	382	4	2
2	1	141	60	382	4	3
3	2	281	80	509	4	2
4	1	141	80	509	4	4

\*註四：參與規劃之供應商與製造廠各六家

透過選取供應商家數及製造廠家數後，將本研究數學模型切割成 90 至 300 不等的組合數，以表 4.23 編號一實驗為例，參與規劃之供應商以及製造廠分別有六家，編號一實驗選取供應商家數為四家，製造廠選取家數為兩家，則其排列組合數為  $225 (C_4^6 \times C_2^6)$ ，分散式平行計算此 225 個組合數，再從此 225 個組合結果中找尋最佳解。以下為各實驗對應不同處理器個數之規劃結果：

## 1. 中規模 2 倍製造廠產能，60% 需求量

表 4.24、中規模 2 倍製造廠產能，60% 需求量平行計算結果

處理器個數	1		4		8	16
計算完畢	否		否		是	是
計算時間 (秒)	28800	36000	28800	36000	17866	5726
時間差異百分比*註五	—	—	0%	0%	37.96%	80.11%
總淨利	90150	90154	92936	92936	93445	93445
淨利差異百分比*註五	—	—	3.090%	3.086%	3.655%	3.655%

\*註五：若計算完畢，差異百分比與單一處理器規劃 8 小時比較，未計算完畢者，與相同規劃時間結果比較。

上表處理器為 1 之實驗即前小節所敘述之全域最佳解實驗。本編號實驗供應商選取家數為 4 家，製造廠選取家數為 2 家，全部的組合數有 225 個。若將處理器數目增加至 4 台，依然無法於 8 小時之內規劃出結果，故於 4 台處理器實驗進行 8 小時與 10 小時規劃比較，探討是否仍有繼續規劃之必要性。結果發現，四台處理器 8 小時以及 10 小時所規劃出來的總淨利相同。而與全域最佳規劃結果相比較，所花費的時間相同，無論是 8 小時或 10 小時，四台處理器所找尋之淨利結果都較單一處理器佳，儘管增加幅度有限，但仍證實利用四台處理器進行平行運算有其貢獻性。

將處理器個數提升一倍至 8 台進行實驗，此實驗花費 17866 秒將規劃完成，此次實驗不但求得最佳解，時間也較單一處理器 8 小時實驗少了 37.96%；進一步試驗 16 台處理器發現，依然可以求得最佳解，且時間僅花費 5726 秒，較 8 小時節省 80.11%。

## 2. 中規模 1 倍製造廠產能，60% 需求量

中規模 1 倍製造廠產能，60% 需求量實驗供應商選取家數為 4 家，製造廠選取家數為 3 家，組合數總共為 300 個 ( $C_3^6 \times C_2^6$ )。表 4.25 為平行算實驗結果，實驗與上段一樣，透過增加處理器的個數進行平行計算，研究在不同個數處理器水準下規劃總淨利變化以及規劃時間的差異。若實驗無法於 8 小時內規劃出最佳解，接著進一步進行 10 小時的規劃實驗，探討 8 小



時與 10 小時的規劃結果差異，實驗結果如下：

表 4.25、中規模 1 倍製造廠產能，60% 需求量平行計算結果

處理器個數	1		4		8	16
計算完畢	否		否		是	是
計算時間 (秒)	28800	36000	28800	36000	20627	6636
時間差異百分比*註五	—	—	0%	0%	28.38%	76.86%
總淨利	93023	93076	93130	93136	93352	93352
淨利差異百分比*註五	—	—	0.115%	0.064%	0.354%	0.354%

\*註五：若計算完畢，差異百分比與單一處理器規劃 8 小時比較，未計算完畢者，與相同規劃時間結果比較。

四台處理器無法於 8 小時內完成規劃，所以進行 8 小時與 10 小時實驗比較，結果顯示，在相同規劃時間下，4 台處理器所求之解較單台處理器全域最佳規劃好，其淨利增加幅度在 0.06%~0.12% 之間；而 8 台處理器規劃淨利為 93352，為最規劃結果，且規劃時間節省 28.38%。16 台處理器也是最佳規劃結果，而時間方面節省了 76.86%。

### 3. 中規模 2 倍製造廠產能，80% 需求量

表 4.26、中規模 2 倍製造廠產能，80% 需求量平行計算結果

處理器個數	1		4		8	16
計算完畢	否		否		是	是
計算時間 (秒)	28800	36000	28800	36000	21444	7277
時間差異百分比*註五	—	—	0%	0%	25.54%	74.73%
總淨利	120948	120948	120957	120965	121025	121025
淨利差異百分比*註五	—	—	0.007%	0.014%	0.064%	0.064%

\*註五：若計算完畢，差異百分比與單一處理器規劃 8 小時比較，未計算完畢者，與相同規劃時間結果比較。

中規模 2 倍製造廠產能，80% 需求量總共實驗組數為 225 組（供應商選



取家數為 4 家，製造廠選取家數為 2 家)。四台處理器平行計算結果無法在 8 小時內規劃完成，而進行四台處理器 10 小時實驗，淨利經過 10 小時運算僅有些許提升，但分別相較於全域最佳規劃都有些許成長；8 台與 16 台處理器規劃則有最佳解產生，其節省之規劃時間百分比分別為 25.52% 與 74.73%，也證明透過多台處理器進行平行計算有不錯的結果。

#### 4. 中規模 1 倍製造廠產能，80% 需求量

表 4.27、中規模 1 倍製造廠產能，80% 需求量平行計算結果

處理器個數	1		4		8	16
計算完畢	否		否		是	是
計算時間 (秒)	28800	36000	28800	36000	24879	9169
時間差異百分比*註五	—	—	0%	0%	13.61%	68.16%
總淨利	120935	120951	120955	120968	121014	121014
淨利差異百分比*註五	—	—	0.017%	0.014%	0.065%	0.065%

\*註五：若計算完畢，差異百分比與單一處理器規劃 8 小時比較，未計算完畢者，與相同規劃時間結果比較。

本編號實驗與前三組實驗結果相似，中規模 1 倍製造廠產能，80% 需求量四台處理器依舊無法於 8 小時以及 10 小時內規劃出結果，但所規劃出之結果與相同規劃時間比較皆有進步，再次證實透過分散式平行計算，在相同時間內規劃結果較單一處理器好。而在 8 台與 16 台處理器的狀況下，求得最佳規劃，且在運算時間上亦有明顯的節省，節省百分比分別為 13.61% 與 68.16%。

圖 4.12 至圖 4.15 為四組實驗依據不同處理器個數，以單一處理器規劃時間為基準，比較處理器因子各水準規劃時間百分比。並分別對照在相同製造廠產能水準下，不同需求量造成規劃時間的差異；以及比較相同需求水準，不同製造廠產能造成計算時間的差異。實驗編號參照表 4.19 之編號，依序為：

編號一：中規模 2 倍製造廠產能，60% 需求量。

編號二：中規模 1 倍製造廠產能，60% 需求量。

編號三：中規模 2 倍製造廠產能，80% 需求量。

編號四：中規模 1 倍製造廠產能，80% 需求量。

圖 4.12 與圖 4.13 為在相同需求水準下(分成 60% 需求量與 80% 需求量)，不同製造廠水準，多種處理器之運算時間比較圖，由圖可知，透過分散式平行計算可將計算時間大幅縮短，若採用 16 台處理器計算，計算時間僅需單一處理器的五分之一（單一處理器採 8 小時規劃時間計算）。在兩組實驗比較下，若製造廠產能限縮至 1 倍，無論需求量为 60% 或是 80%，在 8 台處理器以及 16 台處理器規劃時間上皆有小幅度增加。

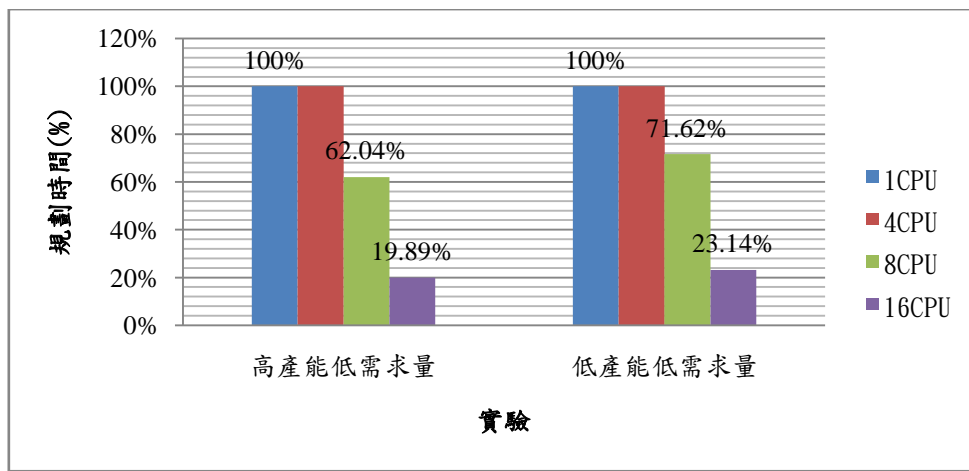


圖 4.12、中規模 60% 需求量實驗規劃時間比較圖

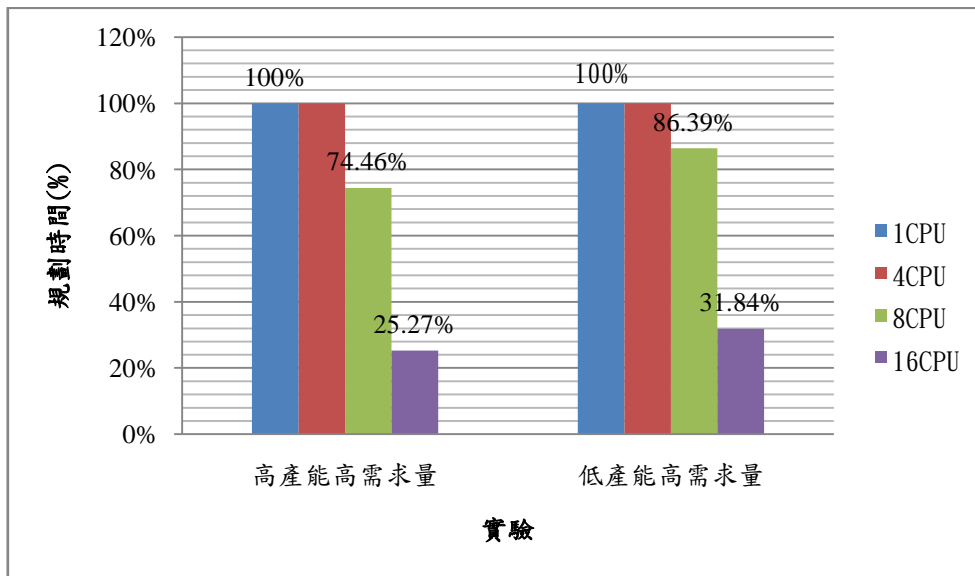


圖 4.13、中規模 80% 需求量實驗規劃時間比較圖

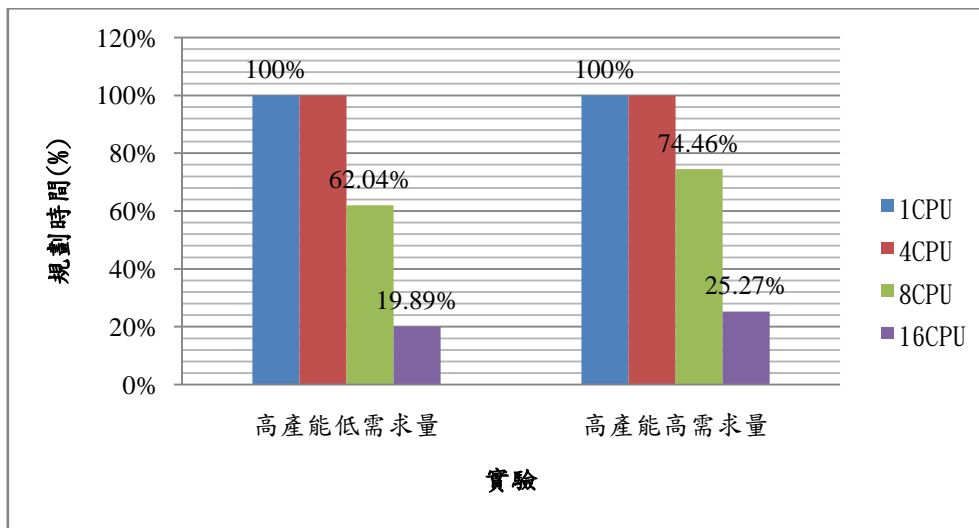


圖 4.14、中規模 2 倍製造廠產能規劃時間比較圖

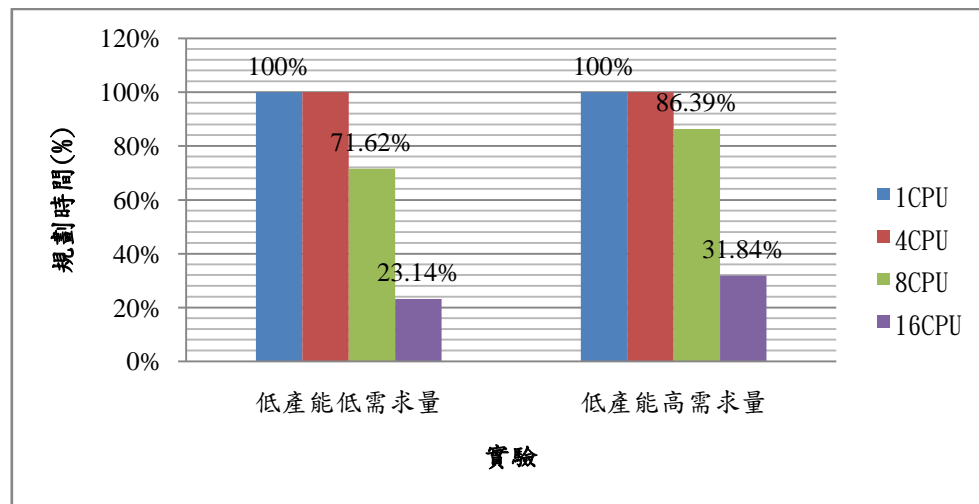


圖 4.15、中規模 1 倍製造廠產能規劃時間比較圖

圖 4.14 與圖 4.15 分別代表在 2 倍與 1 倍製造廠水準下，不同需求水準以及不同處理器個數狀況下對規劃時間的變化。兩張圖除了表示運用多台處理器進行規劃，規劃時間將大幅縮短外，由圖 4.14 顯示，單以 8 台處理器作比較，當需求量自 60% 增加至 80%，規劃時間增加了約 12%；若以圖 4.15 來看，更是增加了約 15%；16 台處理器的部分則是增加了約 6%。各種水準的實際計算時間整理如表 4.27。

中規模實驗結果顯示，以處理器個數水準來看，水準越高，規劃的時間有明顯減少的趨勢；以製造廠產能觀點來看，製造廠產能限縮，會使規劃時間有小幅上升之趨勢；若比較需求量水準因子，需求水準越高，規劃的時間也會越長。

表 4.28、中規模 8 與 16 台處理器實際規劃時間

製造廠產能 倍率	需求量百分 比	規劃時間（秒）	
		8 台處理器	16 台處理器
2	60	17866	5726
1	60	20627	6636
2	80	21444	7277
1	80	24879	9169

#### 4.2.5 大規模全域最佳解與平行計算規劃結果

為了證實本研究模型可以套用到更大的模型，接著擴大環境因子，將規模提升至 8 家供應商節點、8 家製造廠節點、6 家配銷中心節點、8 個顧客節點。因考量到時間因素，大規模實驗僅進行一組編號實驗，隨機挑選 1 倍製造廠產能，需求量 60% 這組進行實驗，而在處理器個數水準上，為了測試本研究在處理器個數上可擴展到更多台電腦，故調整處理器個數水準，進行單一處理器全域最佳規劃，8 台處理器、16 台處理器、32 台處理器分散式平行計算四種實驗。四種實驗結果整理如表 4.28

表 4.29、大規模 1 倍製造廠產能，60% 需求量規劃結果

處理器個數	1		8		16		32
計算完畢	否		否		否		是
計算時間 (秒)	28800	36000	28800	36000	28800	36000	23223
時間差異百 分比	—	—	—	—	0%	0%	19.36%
總淨利	無可行解	無可行解	121938	121947	122123	122134	122176
淨利差異百 分比	—	—	—	—	0.152%	0.161%	0.195%

進行單一處理器全域最佳解規劃時發現，無論是 8 小時規劃或是 10 小時規劃，都無可行解產生，LINGO 規劃截圖如圖 4.16 與圖 4.17，圖中規劃狀態顯示「Unknown」，透過查閱 LINGO 軟體說明，當規劃狀態顯示 unknown

代表在規劃時間內，尚未找尋到可行解，所以在大規模的實驗中無法找到參考淨利值。

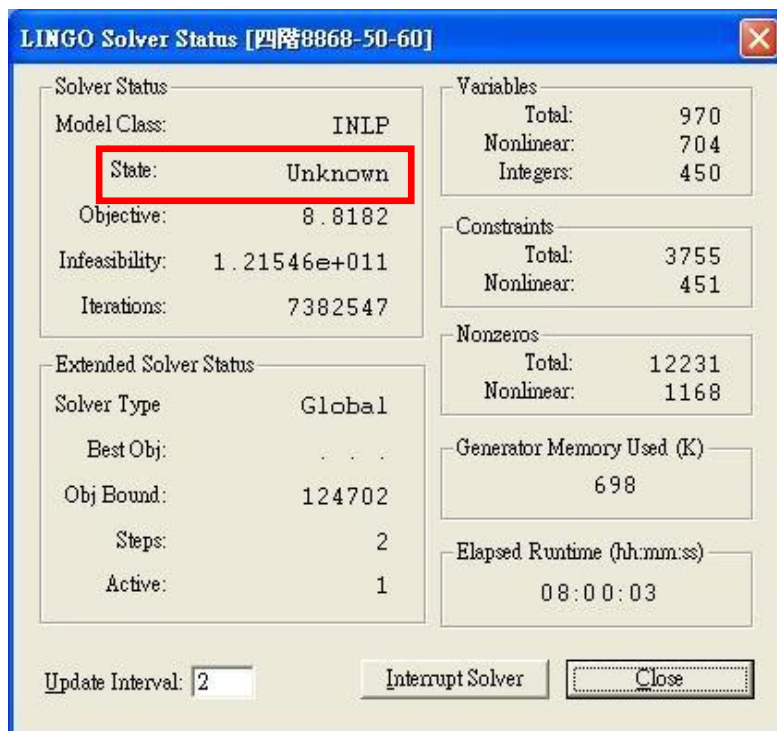


圖 4.16、大規模單一處理器 8 小時處理結果

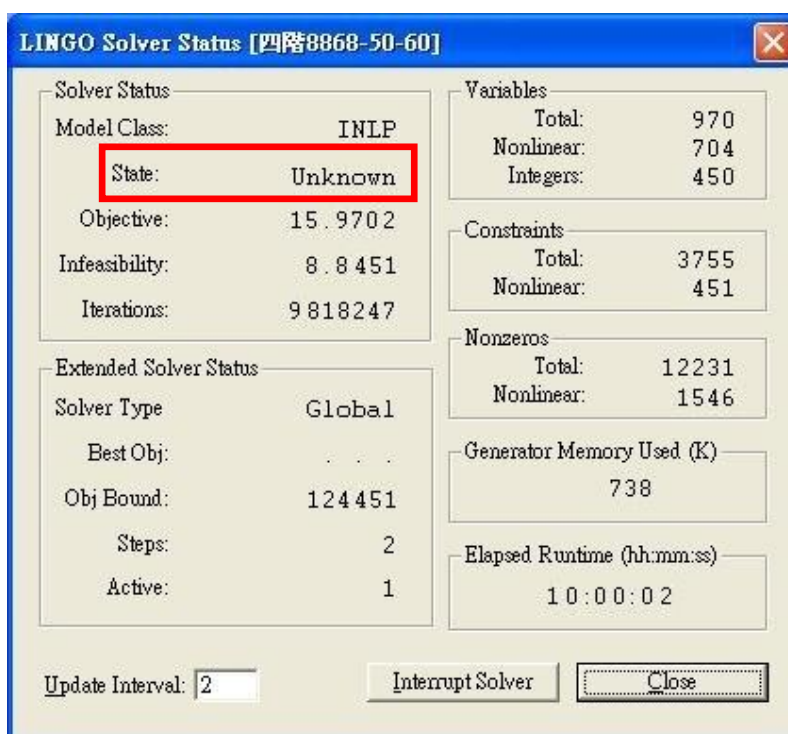


圖 4.17、大規模單一處理器 10 小時處理結果

找尋不到可行解後，嘗試使用多台處理器進行平行運算找出最佳解，在

分割組合數時，選擇的供應商家數為 5 家，製造廠家數為 4 家，總共組合數為 3920 組 ( $C_5^8 \times C_4^8$ )。透過這些分割後，利用 8 台處理器進行運算，在 8 小時規劃時間內無法求出最佳解，僅找到可行解 121938，而 10 小時 8 台處理器規劃亦無最佳解，但求出之可行解與 8 小時相較，10 小時規劃結果較佳。接著嘗試使用 16 台處理器進行規劃，所求結果皆有較佳，但無最佳解產生。最後使用 32 台處理器進行規劃，本次實驗花費 23223 秒的時間，完成 3920 組組合數的計算求出最佳解 122176。

### 4.3 結果討論

本研究與過去類似文獻最大差異在於過去文獻僅探討以三階的供應鏈生產規劃型態，本研究以四階的方式呈現之，並且以製造廠為中心，除在製造廠方面考量開始生產時間外，亦考量製造廠進行生產時所需要之 BOM 表結構，能夠更加精確掌握製造廠開始生產時間以及原物料對於製造廠所產生的影響。最後在各階層皆有考量因運輸批量所產生的批量成本，更能控制運輸成本。透過上述三種環境規模的實驗進行平行運算，結果之說明如下：

首先就小規模實驗進行討論，小規模各組實驗採用重複性實驗，每組實驗皆進行三次，從全域最佳規劃結果來看，產生多重最佳解的狀況，即結果之目標函數值皆相同，僅有運輸路徑、運輸數量在各次實驗有些許不同，判斷為各參數設定皆相同所致，故若參數模式改變，其結果亦有所差異。透過平行計算，計算時間可以有效降低，僅需單一處理器十分之一的時間。

中規模因考量小規模實驗結果產生多重最佳解，故僅進行一次性實驗。在中規模各組實驗中，利用 8 台與 16 台處理器進行規畫可有效降低規劃時間；而採用單一處理器以及 4 台處理器在八小時與十小時的規劃時間內，皆無計算出最佳之規劃結果。接著進行期規劃淨利之比較發現，利用 4 台處理器進行規劃，所求之規劃淨利較單一處理器好，證明平行計算除了可以大幅降低運算時間外，若在有限的規劃時間內，可求解出較佳的規劃結果。從中規模實驗中也發現，若在相同需求量下，低製造廠產能。

大規模僅進行一組實驗，結果顯示擴大環境規模，求解狀況將更加困難，單一處理器於規劃時間內無任何可行解出現，利用平行計算在有限規劃時間，可求得更佳的規劃結果，若採用更多的處理器可求解出最佳規劃結果。

## 第五章 結論與未來發展方向

### 5.1 結論

供應鏈網絡生產規劃是現今許多企業面臨到的重要課題，往往在規劃時間長短與規劃結果是否為最佳化間進行取捨。為解決如此兩難問題，本研究以製造廠為中心，考量製造廠的生產特性，建立適當之數學模型，並且利用數理規劃軟體 LINGO 進行最佳生產規劃。然而從過去文獻證實供應鏈多階多廠的規劃環境是一個 NP-Complete 的問題，經過實驗證實，本研究所建立的供應鏈網絡數學模型其規劃時間會隨著其規劃環境規模擴大，規劃時間呈現指數成長的現象。

對於上述 NP-Complete 之現象，過去研究多以啟發式演算法與模擬方法進行求解，這類方法雖節省了許多的規劃時間，但其規劃結果並非最佳解。故本研究提出以分散式平行計算的方式進行規劃，透過將原本較為複雜的數學模型切割成許多較小的模型，利用 JavaRMI 套件結合 LINGO DLL 功能，將較小的模型分散至各計算節點進行規劃。透過實驗，首先利用小規模實驗驗證本研究數學模型與分散式平行計算之正確性，確認結果無誤後，接著進行中規模實驗。中規模實驗證實當處理器個數增加，規劃時間將有明顯的減少，並且發現當需求水準增加以及製造廠產能限縮，規劃時間皆有增加趨勢。最後為了測試本研究可將規模更加擴大，並且可以利用更多處理器進行運算，結果顯示在更大規模的規劃環境下，求解的狀況會更加困難，在有限時間內僅能以 32 台處理器完成規劃。由上述實驗證實，使用分散式平行計算，除了規劃時間可以大幅縮短外，在有限的規劃時間內，因部分實驗無法求得規劃結果，儘管所求仍為 Local optimal，但若使用更多處理器進行規劃，可以求得更好的規劃結果，也證實了分散式平行計算對供應鏈生產規劃的貢獻。

### 5.2 未來發展方向

本研究雖證實利用分散式平行計算應用於以製造廠為中心之供應鏈生產規劃有其貢獻性，但建構本研究之數學模型時有其假設前提，故未來可更進一步解決本研究模型不盡周全的的假設，增加問題的複雜性。且可考慮站在不同的角度來考量進行供應鏈生產規劃時的特性，例如以供應商的

角度作為出發點，採用不同的供應補貨政策等情況。

在未來，亦可將分散式平行計算之概念應用於某些特定產業，但是將對該產業特性做詳細的了解，其建構出之數學模型更具適合性。且在一些參數設定上，因部分資料取得困難，故部份採用歷史資料，部分採用假設，若有適當機會取得適當資料，其研究將更趨於完整。然而無論針對何種問題，有何種的假設前提，在進行分散式平行計算前，必須了解到該問題的複雜程度，是否有必要利用平行計算解決，因為平行計算的珍貴貢獻在於可顯著降低運算時間抑或可以得到更佳的结果。另外，在進行分散時，會面臨到模型切割的問題，不同問題會有不同的切割方式，但要掌握切割時的合理性以及可行性。



## 參考文獻

### 中文部分：

- [1] 凌繼遠. (2006). 半導體產業多階多廠產能分配機制之構建. 碩士, 國立交通大學.
- [2] 桂思強. (2008). 網路程式設計 *Java Network programming*: 碁峰出版社.
- [3] 郭瑞祥, 蔣明晃, 陳亞男, & 劉基全. (2004). 多廠生產規劃之供應鏈決策支援模式. *臺大管理論叢*, 15(1), 49-74.
- [4] 陳思齊. (2004). 推式多廠區生產規劃與排程-以TFT-LCD面板產業為例. 碩士, 國立清華大學.
- [5] 陳盟翔. (2005). 分散式平行計算模型之設計與建立. 碩士, 義守大學.
- [6] 陳權. (2008). 平行分子動力學應用於個人電腦叢集系統分析與效能研究. 工程科學系, 博士.
- [7] 楊漢鵬. (2006). 對等式平行計算系統與其平行程式設計樣式之研究. 碩士, 長庚大學.
- [8] 葉斯暢. (2010). 分散式平行系統應用於供應鏈網絡規畫-以記憶體模組產業為例. 碩士, 東海大學.

### 英文部分：

- [9] Altiparmak, F., Gen, M., Lin, L., & Karaoglan, I. (2009). A steady-state genetic algorithm for multi-product supply chain network design *Computers & Industrial Engineering*, 56(2), 521-537.
- [10] Altiparmak, F., Gen, M., Lin, L., & Paksoy, T. (2006). A genetic algorithm approach for multi-objective optimization of supply chain networks *Computers and Industrial Engineering* 51(1), 196-215.
- [11] Arntzen, B. C., Brown, G. G., Harrison, T. P., & Trafton, L. L. (1995). Global supply chain management at digital equipment corporation. *Interfaces*, 25(1), 69-93.
- [12] Bhatnagar, R., & Chandra, P. (1993). Models for multi-plant coordination *European Journal of Operational Research*, 67(2), 141-160.
- [13] Byrne, M. D., & Bakir, M. A. (1999). Production planning using a hybrid simulation - analytical approach. *International Journal of Production Economics*, 59, 305-311.
- [14] Chan, T. S., & Chung, S. H. (2004). A multi-criterion genetic algorithm for

- order distribution in demand driven supply chain. *International Journal of Computer Integrated Manufacturing*, 17(4), 339-351.
- [15]Chan, T. S., Chung, S. H., & Wadhwa, S. (2005). A hybrid genetic algorithm for production and distribution *Omega*, 33(4), 345-355.
- [16]Chern, C.-C., & Hsieh, J.-S. (2007). A heuristic algorithm for master planning that satisfies multiple objectives *Computers & Operations Research*, 34(11), 3491-3513.
- [17]Guinet, A. (2001). Multi-site planning: A transshipment problem *International Journal of Production Economics*, 74(1-3), 21-32.
- [18]Hwang, K., & Faye, A. (1984). *Computer architecture and parallel processing*. New York: McGraw-Hill.
- [19]Hwang, K., & Xu, Z. (1998). *Scalable Parallel Computing: Technology, Architecture, Programming*. New York: McGraw-Hill, Inc.
- [20]Ian, F. (1995). *Designing and building parallel programs concepts and tools for parallel software engineering*: Addison-Wesley.
- [21]Jayaraman, V., & Pirkul, H. (2001). Planning and coordination of production and distribution facilities for multiple commodities. *European Journal of Operational Research*, 133(2), 394-408.
- [22]Lambert, D. M., Cooper, M. C., & pagh, J. D. (1998). Supply Chain Management: Implementation Issues and Research Opportunities. *International Journal of Logistics Management*, 9(2), 1-20.
- [23]Lendermann, P., Gan, B. P., & McGinnis, L. F. (2001). *Distributed Simulation with Incorporated APS Procedures for High-Fidelity Supply Chain Optimization*. Paper presented at the Proceedings of the Winter Simulation Conference.
- [24]Leung, S. C. H., Wu, Y., & Lai, K. K. (2003). Multi-site aggregate production planning with multiple objectives: A goal programming approach *Production Planning & Control: The Management of Operations* 14(5), 425-436.
- [25]Sambasivan, M., & Schmidt, C. P. (2002). A heuristic procedure for solving multi-plant, multi-item, multi-period capacitated lot-sizing problems. *Asia - Pacific Journal of Operational Research*, 19(1), 87-105.
- [26]Sauer, J., & Appelrath, H. J. (2000). *Integrating Transportation in a*

- Multi-Site Scheduling Environment*. Paper presented at the Proceedings of the 33rd Hawaii International Conference on System Sciences.
- [27] Sauer, J., Suelmann, G., & Appelrath, H. J. (1998). Multi-site scheduling with fuzzy concepts. *International Journal of Approximate Reasoning*, 19(1-2), 145-160.
- [28] Shapiro, J. F. (2001). *Modeling for the Supply Chain* (First ed.): Thomson Learning.
- [29] Shen, W., Kremer, R., Ulieru, M., & Norrie, D. (2003). A collaborative agent-based infrastructure for Internet-enabled collaborative enterprises *International Journal of Production Research* 41(8), 1621-1638.
- [30] Simchi-Levi, D., Kaminsky, P., & Simchi-levi, E. (2001). *Designing and Managing the Supply Chain : Concept, Strategies, and Case Studies*: McGraw-Hill Companies, Inc.
- [31] Taylor, D. (2003). *Supply Chains A Manager's Guide*: Addison-Wesley Professional
- [32] Thierry, C., Besnard, P., D.Ghettas, & G, B. (1995). Multi-Site Planning: Non Flexible Production Units and Set-Up Time Treatment. *Emerging Technologies and Factory Automation, 1995. ETFA '95, Proceedings., 1995 INRIA/IEEE Symposium on 3*, 261-269.
- [33] Wilkinson, B., & Allen, M. (1998). *Parallel Programming: Techniques and Applications using Networked Workstations and Parallel Computers*: Prentice-Hall, Inc.
- [34] Wu, D. (2004). Multi-Item, Multi-Facility Supply Chain Planning : Models, Complexities, and Algorithms. *Computational Optimization and Applications*, 28(3), 325-356.
- [35] Yeh, W. C. (2005). The International Journal of Advanced Manufacturing Technology *The International Journal of Advanced Manufacturing Technology* 26(5-6), 675-685.
- [36] Yeh, W. C. (2006). An efficient memetic algorithm for the multi-stage supply chain network problem *The International Journal of Advanced Manufacturing Technology* 29(7-8), 803-813.