

國立成功大學
工業與資訊管理研究所
碩士論文

應用網路單形法求解最小分配成本問題之研究

**A network simplex algorithm for the
minimum distribution cost problem**

指導教授：王逸琳 博士

研究生：林修杰

中華民國九十五年七月

國立成功大學
碩士論文

應用網路單形法求解最小分配成本問題之研究

研究生：林修杰

本論文業經審查及口試合格特此證明

論文考試委員：

陳 文 智

許 瑞 麟

李 宇 欣

王 逸 琳

陳 文 智
許 瑞 麟
李 宇 欣
王 逸 琳

指導教授：王 逸 琳

系(所)主管：李 賢 得

中華民國 九十五 年 五 月 二十七 日

博碩士論文授權書

(國科會科學技術資料中心版本, 93.2.6)

本授權書所授權之論文為本人在 國立成功 大學(學院) 工業與資訊管理學 系所
組 九十四 學年度第 二 學期取得 碩 士學位之論文。

論文名稱: 應用網路單形法求解最小分配成本問題之研究

☒ 同意 ☐ 不同意

本人具有著作財產權之論文全文資料, 授予行政院國家科學委員會科學技術資料中心(或其改制後之機構)、國家圖書館及本人畢業學校圖書館, 得不限地域、時間與次數以微縮、光碟或數位化等各種方式重製後散布發行或上載網路。

本論文為本人向經濟部智慧財產局申請專利(未申請者本條款請不予理會的)附件之一, 申請文號為: _____, 註明文號者請將全文資料延後半年後再公開。

☒ 同意 ☐ 不同意

本人具有著作財產權之論文全文資料, 授予教育部指定送繳之圖書館及本人畢業學校圖書館, 為學術研究之目的以各種方法重製, 或為上述目的再授權他人以各種方法重製, 不限地域與時間, 惟每人以一份為限。

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行權為非專屬性發行權利。依本授權所為之收錄、重製、發行及學術研發利用均為無償。上述同意與不同意之欄位若未鈎選, 本人同意視同授權。

指導教授姓名: 王逸琳 博士

研究生簽名:

(親筆正楷)

林俊豪

學號: R36931268

(務必填寫)

日期: 民國 九十五 年 七 月 十一 日

1. 本授權書 (得自 <http://sticnet.stic.gov.tw/sticweb/html/theses/authorize.html> 下載或至 <http://www.stic.gov.tw> 首頁右下方下載) 請以黑筆撰寫並影印裝訂於書名頁之次頁。
2. 授權第一項者, 請確認學校是否代收, 若無者, 請個別再寄論文一本至台北市(106)和平東路二段 106 號 1702 室 國科會科學技術資料中心 黃善平小姐。(電話: 02-27377606 傳真: 02-27377689)

摘要

網路最佳化為作業研究之一重要子領域，而最小成本流量問題 (Minimum Cost Flows; MCF) 即為一常見的網路最佳化問題。MCF 事實上為線性規劃中的特例問題，基於其特殊的網路架構，得以簡化原本在使用單形法 (simplex algorithm) 時繁瑣的矩陣運算，並由此發展出一套更有效率的網路單形法 (network simplex algorithm)。MCF 雖然能夠涵蓋許多網路問題，但仍有其應用的限制，Fang and Qi(2003) 基於產業中製造與分配之特性，將原本的 MCF 加入一種新型態的分配節點 (D 節點)，並提出一套新的最小成本流量模型，稱之為最小分配成本問題 (Minimum Distribution Cost Problem; MDCP)。MDCP 中，流入 D 節點的總流量必須依照給定的比例分配給所有的流出弧。在此架構下，以往的網路單形法將無法適用。雖然 Fang and Qi(2003) 提出一套求解 MDCP 的網路單形法，但該方法僅適用於無弧容量限制下的 MDCP，而且其求解方式仍偏向直接求解方程式，缺乏圖形化的求解方法；另外，他們對於網路單形法中諸如初始解之找尋、流量疊代與計算對偶變數等等的圖形化求解方式亦未深入探討。在本論文中，將進一步討論這些議題，並利用拆解基解圖使之成為數個基群集的方式，以基群集為基礎發展出一套可求解具弧容量限制 MDCP 的圖形化網路單形法。最後，本論文亦將進一步探討加入 D 節點的最大流量問題，以解釋 Lin(2005) 論文中關於最大流量問題達最佳解的對偶變數之分群情形。

關鍵字: 網路最佳化；最小分配成本問題；最小成本流量問題；網路單形法

Abstract

The minimum cost flow problem seeks an optimal flow assignment over a network satisfying the node flow balance constraints and arc flow bounds constraints. These constraints are too simplified to model some real cases. To model the distillation or decomposition of products in some manufacturing processes, a minimum distribution cost problem (MDCP) on a specialized manufacturing network flow model has been investigated. In an MDCP, a specialized node called D-node is used to model a distillation process which only connects with a single incoming arc and several outgoing arcs. The flows entering a D-node have to be distributed according to a pre-specified ratio associated with each of its outgoing arcs. Such a proportional relationship between the arc flows associated with each D-node complicates the problem and makes the MDCP harder than conventional min-cost network flow problem. A network simplex algorithm for uncapacitated MDCP has been outlined in literature, but its detailed graphical procedures such as initial basic feasible solution computation, dual variables updates, and flow pivoting operations have not yet been given. In this thesis, we resolve these issues by upper bound techniques as well as graphical operations which decompose each pivoting graph into several components for calculating both the arc flows and the dual variables. Other issues regarding efficient ways to obtain an initial primal basic feasible solution to start with our algorithm and mathematical insights for solving the MDCP on distribution networks will also be investigated and discussed.

keywords: network optimization, minimum distribution cost problem, minimum cost flow, network simplex algorithm

誌謝

本論文得以順利付梓，承蒙諸位師友的協助與鼓勵。首先非常感謝指導教授王逸琳老師，在研究所求學期間的悉心指導，讓學生在兩年的研究生生活獲益匪淺，謹此致上由衷的敬意與謝忱。

論文口試期間亦承蒙陳文智老師、許瑞麟老師與李宇欣老師提供寶貴的意見，亦感謝各位老師們給予本份論文的肯定以及珍貴且實質的建議，得以讓本論文更加完備，學生在此致上最誠摯的謝意。

在研究所求學生涯中，要感謝研究所同學們兩年來的照顧與互相幫忙，像是常跟我分享經驗的宗儒、每天可以跑20圈操場的宏傑、嗓門很大的雅靖、老大哥敦穎；師門中惠娥、群達、橙坤、姿君、筑軍、小嵐與正翰，還有宛琳、宜倫、子修、維鴻、泰瑋、品樺等研究所的同窗，以及雖然不在台南，但在我徬徨之時也給予我諸多鼓勵的大學死黨們，沒有你們的參與，我的研究所生活就不算多采多姿。

最後，謹將這份完成學士學業的榮耀與喜悅獻給我最敬愛的父母親與家人，感謝你們的細心栽培、照顧與鼓勵，讓我能順利地完成各階段的學業，你們無怨無悔的支持是我不斷前進的動力，謝謝你們。

目 錄

摘要	i
Abstract	ii
誌謝	iii
表目錄	vii
圖目錄	x
符號說明	xi
第一章 緒論	1
1.1 研究動機	2
1.2 研究目的	3
1.3 研究範圍與限制	4
1.4 論文架構	5
第二章 文獻探討	6
2.1 MCF 相關文獻探討	6
2.1.1 MCF 模式建構	6
2.1.2 MCF 求解方法	8
2.1.3 處理容量上界之方法	9
2.1.4 最短路徑問題求解方法	11
2.1.5 最大流量問題求解方法	11
2.2 MDCP 相關文獻探討	12
2.2.1 MDCP 模式建構	12

2.2.2	MDCP 求解方法	17
2.2.3	包含D節點之最短路徑問題求解方法	19
2.2.4	包含D節點之最大流量問題求解方法	20
2.2.5	MDCP 其他相關問題求解方法	20
2.3	小結	21
第三章	模式轉換、主問題與對偶問題	22
3.1	模式轉換	22
3.2	主問題基解架構	26
3.3	對偶問題與最佳解條件	29
3.4	小結	36
第四章	網路單形法	37
4.1	對偶變數計算程序	38
4.1.1	基群集	39
4.1.2	對偶變數求算方法	43
4.2	流量疊代—基變數的進入與退出	46
4.2.1	進入弧之兩端點位於同一群集	47
4.2.2	進入弧之兩端點位於不同群集	48
4.3	對偶變數更新—更新對偶變數與基群集	54
4.4	初始解	59
4.5	單形法、修正單形法與網路單形法的比較	62
4.6	小結	64
第五章	MDCP 相關議題	66
5.1	樞紐圖的錯誤判讀—不存在的Dead-cycle	66
5.2	以T節點之需求為基礎的模式轉換	69
5.3	應用 compatible component 求解樞紐圖流量比例	73
5.3.1	compatible component 求解概念	73

5.3.2	應用 compatible component 於流量疊代	75
5.3.3	compatible component 與基群集之比較	78
5.4	應用網路單形法探討法最大流量問題	81
5.4.1	傳統最大流量問題之對偶性質	81
5.4.2	包含 D 節點之最大流量問題其對偶性質	83
5.5	小結	85
第六章 結論與未來研究方向		86
6.1	結論	86
6.2	未來研究方向	87
參 考 文 獻		89

表 目 錄

4.1 單形法、修正單形法與網路單形法之比較	62
----------------------------------	----

圖目錄

1.1	MDCP 範例	2
2.1	MCF 節點類型	7
2.2	以新增節點方式轉換弧容量限制式	10
2.3	應用上界技巧於網路問題	10
2.4	$E(i)$ 與 $L(i)$ 集合 (來源:Fang and Qi, 2003)	13
2.5	MNF : S 節點 (來源:Fang and Qi, 2003)	13
2.6	MNF : T 節點 (來源:Fang and Qi, 2003)	14
2.7	MNF : O 節點 (來源:Fang and Qi, 2003)	14
2.8	MNF : I 節點 (來源:Fang and Qi, 2003)	14
2.9	MNF : D 節點 (來源:Fang and Qi, 2003)	15
2.10	MNF : C 節點 (來源:Fang and Qi, 2003)	16
2.11	在 MDCP 中, T 節點可能接收多於其需求下限的流量值	17
2.12	MDCP 範例與其基解圖	18
3.1	原 MDCP 模式套用上界技巧造成圖形分離	23
3.2	MDCP-S 節點之轉換	25
3.3	MDCP-T 節點之轉換	25
3.4	新增節點 s 與 t 以轉換多個供給點、多個需求點之 MDCP 網路	25
3.5	轉換後 MDCP 範例與其基解圖	29
3.6	MDCP 轉換容量限制之網路-兩端為 \hat{O} 節點之弧	31
3.7	MDCP 轉換容量限制之網路-尾端為 \hat{O} 節點, 頭端為 D 節點之弧	32
3.8	MDCP 轉換容量限制之網路-尾端為 D 節點, 頭端為 \hat{O} 節點之弧	33
3.9	D 節點之對偶變數關聯示意圖	35
4.1	對偶變數之推算受限於 D 節點	38

4.2	新增 i_j 以對應 ρ_{ij}	39
4.3	基群集 (BC)	40
4.4	若 D 節點有一條相鄰弧不為基底，存在自成群集之節點 (虛線圈選處)	40
4.5	圖 4.3 求解後之對偶變數值	44
4.6	MDCP 樞紐圖的流量比例關係複雜	46
4.7	進入弧兩端點皆位於同一群集，其推送流量比例簡單	47
4.8	針對 D 節點的流入弧設置變數，並推算其流出弧流量改變量	50
4.9	從樹之葉端往根節點推算流量	51
4.10	計算樞紐圖流量比例範例	52
4.11	移除退出弧，群集分割為兩子群集	55
4.12	對偶變數更新範例	58
4.13	非基解圖其對偶變數無解	60
4.14	初始解示意圖 (適用於精簡過後之網路)	61
4.15	D 節點皆與所有相連弧之基解圖亦有可能不用求解聯立方程組	63
4.16	每一個群組均互相連接	64
5.1	樞紐圖中形成 dead-cycle (來源:Fang and Qi, 2003)	67
5.2	dead-cycle 並不存在，可依此圖推送流量	68
5.3	Fang and Qi 對樞紐圖判讀錯誤 (來源:修改自 Fang and Qi, 2003)	69
5.4	以需求為基礎轉換 T 節點	71
5.5	以需求為基礎轉換 S 節點	71
5.6	以需求為基礎轉換多個供給點、多個需求點之 MDCP 網路	72
5.7	Compatible component (來源:Ting, 2005)	73
5.8	不可通過流量之子圖 (來源:Ting, 2005)	74
5.9	求解後之變數違反非負限制 (來源:Ting, 2005)	75
5.10	應用 compatible component 求解樞紐圖流量比例	76
5.11	加入弧 (9, 8) 後，圖形可通過流量	77
5.12	compatible component 與基群集分群方式不同	79

5.13 適當的搜尋可減少樞紐圖中所設置的變數個數	80
5.14 視群組特性可減少計算對偶變數時所設置的變數個數	80
5.15 傳統最大流量問題可轉換為MCF	81
5.16 傳統最大流量問題其對偶特性	82
5.17 包含D節點之最大流量問題範例1	83
5.18 包含D節點之最大流量問題範例2	85
6.1 strongly feasible spanning tree	88

符號說明

- A ：網路中的弧集合。
- \tilde{A} ：刪除分支後，樞紐圖中的弧集合。
- B ：基弧所構成的弧集合。
- BC_i ：表示第 i 個基群集。
- b_i ：節點 i 的淨流量值，等於流出量減去流入量的差值。
- c_{ij} ：弧 (i, j) 上的單位流量成本。
- e_{ij} ：在流量疊代中進行最小值檢定時，弧 (i, j) 計算之值。
- f_{ij} ：樞紐圖中弧 (i, j) 相對於進入弧之流量的流量比例係數。
- G ：表示一個有向網路系統。
- G_B ：基解圖形。
- \tilde{G} ：刪除分支後的樞紐圖， $\tilde{G} = (\tilde{N}, \tilde{A})$ 。
- k_{ij} ：D節點 i 的分流係數，為流出弧 (i, j) 的流量對流入弧 (i^*, i) 之流量所佔的比例。
- L ：流量達下界之非基弧所構成的弧集合。
- M ：某一個代表沒有上限的數。
- N ：節點集合。

- \tilde{N} ：刪除分支後，樞紐圖中的節點集合。
- N_{Ts} ：MCF 中，淨流量值大於0的節點集合。
- N_{Tt} ：MCF 中，淨流量值小於0的節點集合。
- N_{To} ：MCF 中，淨流量值等於0的節點集合。
- N_S ：MNF 中，供給節點(S節點)構成之集合。
- N_T ：MNF 中，需求節點(T節點)構成之集合。
- N_O ：MNF 中，轉運節點(O節點)構成之集合。
- N_I ：MNF 中，庫存節點(I節點)構成之集合。
- N_C ：MNF 中，合流節點(C節點)構成之集合。
- N_D ：MNF 中，分流節點(D節點)構成之集合。
- $N_{\hat{O}}$ ：傳統節點之集合， $N_{\hat{O}} = N_S \cup N_O \cup N_T$ 。
- t_i ：計算對偶變數時，用來推導 BC_i 內所有對偶變數的單一變數。
- U ：網路中流量達上界之非基弧所構成的弧集合。
- u_{ij} ：弧 (i, j) 上的流量上限， $x_{ij} \leq u_{ij}$ 。
- x_{ij} ：弧 (i, j) 上之流量值。
- δ_i ：更新對偶變數時， BC_i 內每個對偶變數的增加量。
- π_i ：對應節點 i 之流量守恆限制式的對偶變數。
- ρ_{ij} ：對應 D 節點 i 流出弧 (i, j) 之分流限制式的對偶變數。

第一章

緒論

網路問題常在各種場合以各種不同形式出現，舉凡運輸、電力、通訊網路等等應用問題皆與日常生活息息相關。透過網路的表示方法，可為問題本身提供更直覺及清楚的認知，使人們容易了解系統中各項活動之間的相互關聯。因此，網路模式在工業、科學、經濟等領域的應用相當廣泛，亦使得探討各類網路問題並求取其最佳解的網路最佳化問題(network optimization)為作業研究之一重要子領域。

在眾多的網路最佳化問題中，最小成本流量問題(Minimum Cost Flows；MCF)探討以最小總成本之方式，在滿足節點的流量守恒限制與弧的容量限制下，使供給節點產生流量通過網路以滿足需求節點之需求。MCF事實上為線性規劃中的特例問題，其特殊的網路架構可以用來簡化使用單形法(simplex algorithm)時繁瑣的矩陣運算，並由此發展出一套更有效率的網路單形法(network simplex algorithm)。

由於MCF僅考慮流量守恒限制與弧容量限制，因此僅能被用來描述較簡化的網路最佳化問題；若要用MCF處理更實際而複雜的網路最佳化問題，其網路數學模式必須加以修改。例如Fang and Qi(2003)基於製造網路的特性而提出一種製造網路流量模型(Manufacturing Network Flow；MNF)，該模型將不同材料以特定的比例去組合成一個產品，或是一個產品可以依比例分解成數個材料的現象列入考慮，因此可視為在原本MCF的架構下，額外加入特殊節點所構成的網路問題。在此架構下，以往求解MCF的演算法將無法適用，而必須對模式經過適當的修改，並探討新增的限制式對問題所產生的改變，才能正確地解決MNF。

本研究之主旨即在探討MNF相關的數學模式—最小分配成本流量問題(Minimum Distribution Cost Problem; MDCP)之特性，冀能提出更有效率的求解方法。

1.1 研究動機

Fang and Qi(2003)依照製造網路的特性，將網路中的節點(node)依其類型分為傳統MCF所擁有的供給節點(S節點)、需求節點(T節點)與轉運節點(O節點)，以及因應製造網路而設計的存貨節點(I節點)、分流節點(D節點)與合流節點(C節點)等六大類。此外，他們亦針對包含D節點的網路問題定義出MDCP，其中流入D節點的總流量必須依照給定的比例分配給其所有的流出弧。圖1.1表

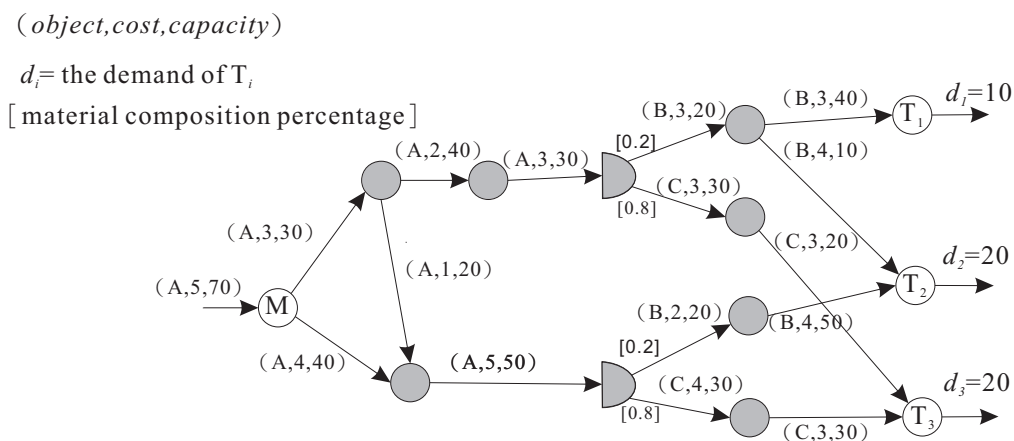


圖 1.1: MDCP 範例

示一個逆向物流問題中常見的回收網路圖，M表示產品A的回收點，負責產品A的回收；圖中以半圓形表示的節點即為回收網路中的拆解點，可以進一步將每一單位的A拆解分類為0.2單位的B與0.8單位的C； T_1 、 T_2 與 T_3 則為物料B與C的需求點， d_i 表示 T_i 的最小需求，另外各弧都有其轉運成本及容量上限。此網路可視為一個MDCP，M代表的是S節點， T_1 、 T_2 與 T_3 代表T節點，拆解點為D節點，網路中其他灰色的節點則可視為O節點，模式目標即希望以最小化的系統成本來使得供給可以滿足需求。

雖然 Fang and Qi(2003) 已在其論文中提出一套求解MDCP的網路單形法，然而其所提出的方法仍有不盡完備之處，茲列舉兩點如下：

1. 未將弧容量(arc capacity)列入考量：雖然 Fang and Qi(2003) 提出 MDCP 模型，但卻是在網路中之弧無容量限制下進行模式的探討與分析；因此他們所提出的網路單形法僅適用於無容量上限的特殊 MDCP，缺乏應用面的一般性。
2. 缺少求解過程的詳細探討：Fang and Qi(2003) 所提出的網路單形法在求解弧之流量與對偶變數的過程仍以求解聯立方程式為主，缺乏圖形化的求解方式。而且他們對於網路單形法中諸如初始解之找尋、流量疊代(主變數的進入與退出)與對偶變數之更新等程序亦未詳加探討。

此外，若將 MDCP 分別去除弧容量限制或成本限制也可以形成兩種特例問題—包含 D 節點的最短路徑問題與包含 D 節點的最大流量問題。Yang(2005) 引入 D 節點的特性定義了三種不同類型的最短路徑問題，稱之為 UMDCP(Uncapacitated Minimum Distribution Cost Flow Problem)，並試圖修改 Dijkstra 演算法(Dijkstra, 1959) 以求解 UMDCP；但在求解其第三類問題 UMDCP₃ 時，無法修改 Dijkstra 演算法而採用窮舉法求解；Ting(2005) 仿效 Ford and Fulkerson(1956) 求解最大流量問題的演算法，嘗試修改 DFS(Depth-First Search Algorithm) 尋找流量增益路徑(flow-augmenting path) 以求解包含 D 節點的最大流量問題，但其求解時間呈指數(exponential) 成長，理論上之求解效率不佳；Lin(2005) 則嘗試修改 Goldberg and Tarjan(1988) 所提出的預流推進法(preflow-push algorithm) 求解包含 D 節點的最大流量問題，但仍無法完全成功地求解該類問題。本研究嘗試使用網路單形法求解 MDCP，回到 MDCP 的網路特性，思考基解的架構，並探討加入 D 節點對問題所造成的影響，補充並修正 Fang and Qi(2003) 論文中尚未完備之處，進而對 MDCP 的模式特性與求解方式有更進一步的理解。

1.2 研究目的

本研究所欲探討的問題，主要承襲自 Fang and Qi(2003) 提出的 MDCP 模型；但為了保證基解圖形的相連性，本研究將其模式先做適當轉換，並討論具有容量

限制MDCP模式的基解特性及對偶性質。在求解MDCP的網路單形法方面，本研究將探討如何以「圖形化」且有系統之方式求解對偶變數以及流量疊代等程序；此外，本研究亦將針對Fang and Qi(2003)在其網路單形法中未詳盡介紹之程序，諸如對偶變數更新與初始解設立等重要步驟，提出詳盡且有效率的執行程序，以使整個網路單形法更為完備。

另外，對於包含D節點的最大流量問題，本研究亦將採用網路單形法分析其最佳解之對偶性質，以解釋Lin(2005)在其論文中所觀察到的對偶變數分群之現象。

1.3 研究範圍與限制

本研究對MDCP的網路圖做以下一般性的假設與限制：

1. 網路節點類型由S節點、T節點、O節點及D節點四種節點所構成。
2. 由於兩D節點相鄰之網路可再做簡化(Lin, 2005; Yang, 2005)，故本研究探討的網路類型為精簡過後之網路，其中最重要之性質即為任兩D節點不相鄰。
3. 假設每個節點不存在自我迴圈(亦即每一條弧的起點終點不為同一點)。
4. 節點與節點之間無平行弧的存在，連接相同節點的弧為唯一。
5. 各弧的總流量成本與其流量成正比。
6. 每條弧的容量下界為零，且允許有其各自的流量上限。
7. 從D節點流出之弧其分流比例為界於(0,1)之實數，且從該點流出弧其分流比例總和為100%。
8. 網路圖上所有的參數，包括成本、弧容量、節點淨流量值及D節點流出比例皆為固定的已知參數。

1.4 論文架構

本論文之架構如下：第二章為文獻探討，回顧與MCF及MDCP相關的文獻；第三章先將MDCP模式給予適當轉換，探討具弧容量限制的MDCP其基解特性與對偶問題特性，並依據對偶問題可行解條件做為本研究所提出之網路單形法的最佳解條件；在第四章中，本研究將探討如何以圖形化的操作方式求解對偶變數、流量疊代、對偶變數更新與初始解設立等等與網路單形法相關的計算程序；第五章探討其他與MDCP相關之議題，最後在第六章提出相關結論與建議。

第二章

文獻探討

本章主要回顧最小成本流量問題(Minimum Cost Flows; MCF)與最小分配成本問題(Minimum Distribution Cost Problem; MDCP)相關之文獻。因為MCF與MNF對於節點之定義稍有不同，故本研究以 N_{Ts} 、 N_{Tt} 、 N_{To} 等符號來表示MCF中的節點集合；而以 N_S 、 N_T 、 N_O 、 N_I 、 N_D 、 N_C 等符號表示MNF中的節點集合。

2.1 MCF 相關文獻探討

2.1.1 MCF 模式建構

一般常以 $G = (N, A)$ 表示一有向網路系統， N 為節點所構成之集合， A 為有向弧所構成之集合，其中 $|N| = n$ ，而 $|A| = m$ 。系統中常以 x_{ij} 表示弧 (i, j) 上的流量， c_{ij} 表示通過弧 (i, j) 的單位流量成本， u_{ij} 表示弧 (i, j) 之流量上限， b_i 則是節點 i 之流出量與流入量的差值(即淨流量值)。利用 b_i 值的不同，可將節點集合 N 區分為 N_{Ts} 、 N_{Tt} 、 N_{To} 三種類型：

1. 供給節點：屬於該節點的集合以 N_{Ts} 表示，對節點 $i \in N_{Ts}$ ，其流出量大於流入量，使得 $b_i > 0$ 。
2. 需求節點：屬於該節點的集合以 N_{Tt} 表示，對節點 $i \in N_{Tt}$ ，其流入量大於流出量，使得 $b_i < 0$ 。
3. 轉運節點：屬於該節點的集合以 N_{To} 表示，對節點 $i \in N_{To}$ ，其流入量等於流出量，使得 $b_i = 0$ 。

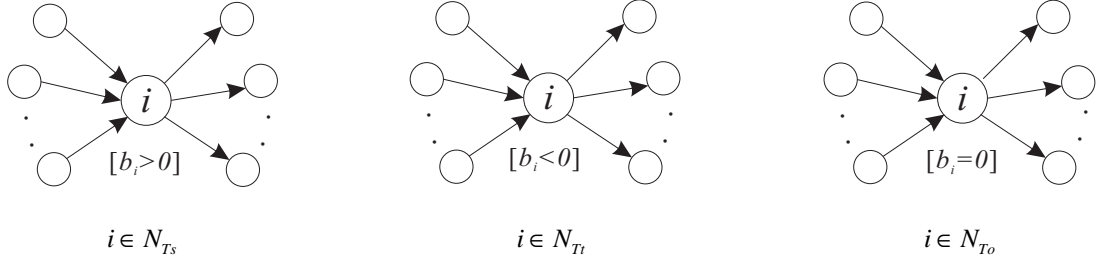


圖 2.1: MCF 節點類型

MCF 的模式目標即希望在不違反弧容量限制與流量守恆限制下，以最小總成本之方式使供給節點之流量通過網路滿足需求節點之需求，其模式可列式如下：

$$\min \sum_{(ij) \in A} c_{ij} x_{ij} \quad (2.1)$$

$$s.t. \quad \sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} = \begin{cases} b_i > 0 & \forall i \in N_{Ts} \\ b_i = 0 & \forall i \in N_{To} \\ b_i < 0 & \forall i \in N_{Tt} \end{cases} \quad (2.2)$$

$$0 \leq x_{ij} \leq u_{ij} \quad \forall (i, j) \in A \quad (2.3)$$

其中 (2.1) 為此網路問題之目標式，(2.2) 為節點之流量守恆限制式，而 (2.3) 則是弧容量限制式。MCF 在某些應用題型中，模式要求弧 (i, j) 至少通過流量 $l_{ij} \geq 0$ ，若遇到此情形，則可利用代數變換令 $x'_{ij} = x_{ij} - l_{ij}$ ，代換 $x_{ij} = x'_{ij} + l_{ij}$ ，使問題轉換成弧容量下限為 0 之格式。

MCF 為確保問題的可行性，模式常使 $\sum_{i \in N} b_i = 0$ 代表供需均衡。假使 $\sum_{i \in N} b_i < 0$ ，表示供給不足需求；反之，若 $\sum_{i \in N} b_i > 0$ ，則代表供給大於需求。面對供需不均衡，數學模式上常以超額的供給(需求)當作其上限，並另設虛擬需求(供給)節點以吸收超額供給(需求)量，以此做為供需均衡的手段。另外，如果所有的 b_i 及 u_{ij} 皆為整數，根據限制式的整數解特性(Hoffman and Kruskal, 1956)，則 MCF 所有的可行基解(basic feasible solution)亦為整數。

許多基本的網路最佳化問題，只要經過適當的定義，就能夠轉換成 MCF，例如最短路徑問題可視為無容量上限的 MCF，並可依其起訖點類型區分為一對一、一對多與多對多的最短路徑問題。同樣地，最大流量問題亦為 MCF 之特例

問題，如果令所有弧之單位流量成本皆為0，並新增一條從終點至起點之弧，且令其單位流量成本為-1，就可將問題轉換為MCF以求解。

2.1.2 MCF 求解方法

MCF屬於線性規劃中的特例問題，Dantzig(1951)首先探討MCF使用單形法時的特性，仿照主單形法(primal simplex algorithm)發展出網路單形法(network simplex algorithm)以求解不具弧容量限制的MCF，並確立MCF的基解圖為一展開樹(spanning tree)。之後，Dantzig(1962)亦採用線性規劃中上界技巧之概念，發展出可求解具容量限制問題的網路單形法。網路單形法與單形法相仿，以主網路單形法為例，其操作步驟亦可分為幾個步驟來描述。首先，尋找一個初始的可行基解，並計算該基解所對應的對偶變數；之後，檢查網路中的每條弧是否滿足最佳解條件(對偶解可行為其最佳解條件)，選擇一條違反對偶解可行之弧做為進入基底之變數；最後，依照樞紐圖(pivoting graph，表示基解圖加入進入弧之後所形成之圖形)推送流量，選擇流量最先到達上界或下界的弧為退出變數，並更新對偶變數，就完成一次iteration；接著重複以上步驟直至最佳解，即求解出最小成本流量。

此外，Ford and Fulkerson(1962)提出主-對偶演算法(primal-dual algorithm)將MCF拆解成數個最短路徑問題與最大流量問題以求解；主-對偶演算法乃反覆尋找從供給節點至需求節點的最短路徑，並繪出所有滿足 $c_{ij} - \pi_i + \pi_j = 0$ 之弧所構成之子圖(π_i 為節點 i 之對偶變數)，再根據此子圖運送最大流量；而當所有的供給節點流量均滿足需求節點時，即解得最佳解。Ali et al.(1989)則嘗試發展對偶網路單形法(dual network simplex algorithm)以求解MCF，並規範了一些應用上的規則。另外，也有學者將MCF分解成數個最短路徑問題以求解，並以此發展出連續最短路徑演算法(successive shortest path algorithm)，詳見Ahuja et al.(1993)之9.7節。

網路單形法基於網路的特性，得以簡化原本在單形法中繁瑣的矩陣運算，並形成一套清楚易懂且兼具效率的圖形式演算法。雖然網路單形法的操作簡單，其求解時間仍有可能呈指數成長。而在經過一些設計安排之後，目前已發展出具

多項式求解時間 (polynomial time) 的網路單形法，舉凡主網路單形法 (Ahuja et al., 1992; Orlin, 1997) 與對偶網路單形法 (Orlin et al., 1993; Armstrong and Jin, 1997)，均有學者致力於其理論效率的改進。這些演算法均顯示了網路單形法無論在實務或是理論上均有卓越表現。

2.1.3 處理容量上界之方法

在線性規劃範疇裡，常以目標式 $\min cx$ 、限制式 $Ax = b$ 、變數非負限制式 $x \geq 0$ (要求下界為 0 而無上界限制) 做為標準形式 (standard form)，一般單形法也常以標準形式做為其基本的數學模式。但在某些模式中，問題的決策變數可能具有上界限制，不符合標準形式之格式。一般面對此種情形時往往有兩種做法：一是將上界限制式視為一般的函數限制式；二是將上界限制式視為退出基變數的另一項條件，並採取代數變換以轉換問題。同樣地，在求解具容量限制的網路問題時，也有類似的兩種作法：第一種方法採用新增節點之方式，將原網路轉換模式為無容量限制的網路；第二種作法則是採取線性規劃中的上界技巧套用於網路問題。茲將此兩種方式分述如下：

1. 轉換模式為無容量限制之網路：此方法將容量限制式視為函數限制式的一部份，對每一條弧 $(i, j) \in A$ 之容量限制式 $x_{ij} \leq u_{ij}$ 加入超額變數 (slack variable) s_{ij} ，使容量限制式變為 $x_{ij} + s_{ij} = u_{ij}$ ，兩邊同乘 -1 之後，可得：

$$-x_{ij} - s_{ij} = -u_{ij} \quad (2.4)$$

因此，(2.4) 可視為新增節點 k 的流量守恒限制式，其中 x_{ij} 與 s_{ij} 為節點 k 之流入量，節點 k 淨流量為 $-u_{ij}$ ，表示需求為 u_{ij} 單位。另外對於弧 (i, j) 的流入節點 j ，亦可利用 (2.4)，代換原本節點 j 之流量守恒限制式，可得 (2.5) 如下：

$$\begin{aligned} & \sum_{(j,k) \in A} x_{jk} - \sum_{(k,j) \in A, k \neq i} x_{kj} - x_{ij} = b_j \\ \Rightarrow & \sum_{(j,k) \in A} x_{jk} - \sum_{(k,j) \in A, k \neq i} x_{kj} - (u_{ij} - s_{ij}) = b_j \\ \Rightarrow & \sum_{(j,k) \in A} x_{jk} - \sum_{(k,j) \in A, k \neq i} x_{kj} + s_{ij} = b_j + u_{ij} \end{aligned} \quad (2.5)$$

所以對節點 j 而言，原本從節點 i 流入節點 j 之流量 x_{ij} ，就可替代成從節點 j 流出至節點 k 之流量 s_{ij} ，且節點淨流量增加為 $b_i + u_{ij}$ ，圖 2.2 為轉換前後圖形之改變。

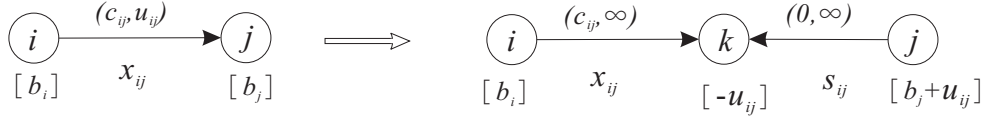


圖 2.2: 以新增節點方式轉換弧容量限制式

2. 採用上界技巧處理容量限制：雖然採取新增節點之方式可以移除問題中的流量限制，但在轉換的同時也增加了 $|A| = m$ 個節點，擴大了原問題的大小。為了能夠有效的處理弧容量限制式，另外一種做法便是不將此限制式視為函數限制式，而僅在選定退出基變數時才將之處理。原先單形法在求解標準形式的問題時，僅需注意何項變數 x_{ij} 到達下界退出基底；但在具有上界 u_{ij} 的模式裡，還需要注意何項基變數先到達上界，並以此做為退出基底的另一項條件。因此，若原先基變數 x_{ij} 到達上界 u_{ij} 退出基底，則將 x_{ij} 以 $u_{ij} - x'_{ij}$ 替代原問題，使 $x'_{ij} = 0$ 成為非基變數。同樣的步驟在網路問題中，可視為弧變數 x_{ij} 代換成反向弧變數 x_{ji} 之方式，並令 $c_{ji} = -c_{ij}$ 、 $b_i = b_i - u_{ij}$ 、 $b_j = b_j + u_{ij}$ ，且 $u_{ji} = u_{ij}$ 。另外，除了代換變數的方法外，亦可視為到達上界與到達下界的非基變數各有其不同的最佳解條件，對於流量達下界的非基弧 (i, j) ，其對偶變數 π_i 與 π_j 須滿足 $\pi_i - \pi_j \leq c_{ij}$ ；而若弧 (i, j) 為流量達上界之非基弧時，則其對偶變數則須滿足 $\pi_i - \pi_j \geq c_{ij}$ ，此為上界技巧的另一種表示方法。

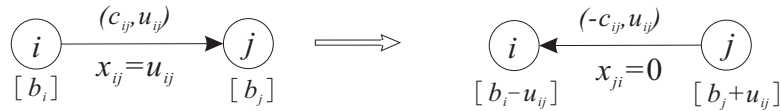


圖 2.3: 應用上界技巧於網路問題

2.1.4 最短路徑問題求解方法

最短路徑問題為MCF問題中的特例問題，一般求解最短路徑問題有兩種廣泛應用的演算法：標籤設定法(label-setting algorithm)和標籤修正法(label-correcting algorithm)。

最具代表性的標籤設定法為Dijkstra演算法(Dijkstra, 1959)，適用於成本非負的一對多最短路徑問題。Dijkstra演算法為一種貪婪式(greedy)的演算法，節點標籤 d_i 記錄的是起點至節點 i 的距離值，此方法藉由不斷地選擇最小標籤之節點，並針對所有弧更新節點 j 的標籤 $d_j = \min\{d_j, d_i + c_{ij}\}$ ，即可求解出一對多的最短路徑問題。標籤選擇的資料結構會影響Dijkstra演算法的時間複雜度，如果沒有經過特殊設計，所需的理論時間為 $O(n^2)$ ；但是如果採取特殊的資料結構，則可以衍生出許多應用版本。Dial(1969)將節點標籤設為 $nC + 1$ 個「提桶(bucket)」(C 為弧最大成本值)，分別代表儲存標籤值為0到 nC 的節點，應用此方法可以在 $O(m + nC)$ 的時間內解決最短路徑問題，其在實務應用上擁有不錯的效率；Johnson(1972)使用binary-heap儲存標籤資料，時間複雜度為 $O(m \log n)$ ；Fredman and Tarjan (1987)則使用Fibonacci-heap儲存標籤資料，時間複雜度為 $O(m + n \log n)$ 。

求解最短路徑問題另一種常用的方法為標籤修正法。標籤修正法是由Ford(1956)所提出的演算法，可適用於弧成本為負但無負迴圈(negative cycle)的網路，每一次iteration均檢查 $d_j > d_i + c_{ij} : \forall (i, j) \in A$ ，條件成立則更新 d_j 。採取FIFO(First-In-First-Out)(Bellman, 1958)的標籤修正法可以在 $O(nm)$ 時間內求解此問題。Goldfarb and Jin(1999)提出使用單形法的操作概念亦可在 $O(nm)$ 時間內求解最短路徑問題，並將標籤設定法與標籤修正法視為單形法的特殊應用。

2.1.5 最大流量問題求解方法

最大流量問題亦為MCF之特例問題，也有許多學者針對最大流量問題發展特殊的求解方法。一般求解最大流量問題的演算法主要有流量增益路徑法(flow-augmenting path algorithm)和預流推進法(preflow-push algorithm)。

流量增益路徑法是由 Ford and Fulkerson(1956) 所提出的演算法，採取的是一種直覺式的解題方式，藉由不斷地尋找起點到終點的路徑並推送流量，直到已無流量可推送至終點時即得最佳解。此種解法假如沒有訂立特殊的篩選規則，其最佳解未必是基解（亦即其解不一定為展開樹結構）；因此流量增益路徑法可視為在可行區域內尋找改善向量求解的演算法，其時間複雜度為 $O(mnU)$ (U 為所有弧的最大容量上限)。

流量增益路徑法並非強多項式時間 (strongly polynomial time) 的演算法，Ahuja and Orlin(1991) 利用節點標籤記錄節點到終點所經最少弧之個數，選擇經過弧個數最少之路徑優先推送流量，而發展出一套可在 $O(n^2m)$ 時間內求解的演算法；Goldfarb and Hao(1990) 使用主單形法的操作概念，在 $O(n^2m)$ 時間內解出最大流量問題，將最短流量增益路徑法和主單形法建立起關聯。

預流推進法為另外一種解決最大流量問題的演算法，預流的概念一開始是由 Karzanov(1974) 提出，之後由 Goldberg and Tarjan(1988) 發展出較為一般性的預流推進法，其時間複雜度為 $O(n^2m)$ 。Goldfarb and Chen(1997) 則利用預流的概念，說明對偶單形法可在 $O(n^2m)$ 時間內求解最大流量問題；之後 Armstrong et al.(1998) 亦採用預流之概念發展可在 $O(n^3)$ 時間內求解最大流量問題的類對偶 (dual-like) 單形法，此兩篇皆說明預流推進法可視為對偶單形法的一項應用。

2.2 MDCP 相關文獻探討

2.2.1 MDCP 模式建構

Fang and Qi(2003) 根據製造產業之特性提出製造網路模型 (Manufacturing Network Flow, MNF)。在該篇論文中，首先對任一節點 $i \in N$ 定義出兩類集合——流入節點 i 的節點集合 $E(i)$ 及流出節點 i 的節點集合 $L(i)$ ，如下所示：

$$E(i) := \{j \in N : (j, i) \in A\}$$

$$L(i) := \{j \in N : (i, j) \in A\}$$

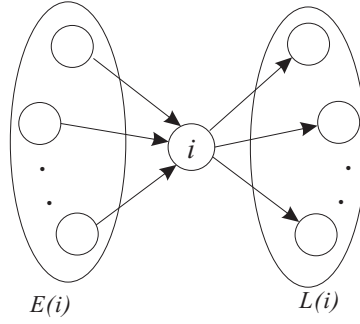


圖 2.4: $E(i)$ 與 $L(i)$ 集合 (來源:Fang and Qi, 2003)

而 MNF 中的節點類型可依其功能分為六大類：

1. 供給節點(S 節點)：屬於 S 節點的集合以 N_S 表示，為 MNF 的原料供給節點。MNF 中的供給節點可能為一個或多個，此類型的節點只有流出弧，而無來自於其他節點的流入弧。另外，為了維持流量守恆限制式，對於節點 $i \in N_S$ 給予 x_i 代表流出量總和。

$$x_i = \sum_{j \in L(i)} x_{ij} \quad \forall i \in N_S \quad (2.6)$$

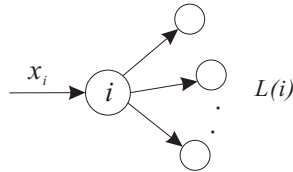


圖 2.5: MNF : S 節點 (來源:Fang and Qi, 2003)

2. 需求節點(T 節點)：屬於 T 節點的集合以 N_T 表示，為 MNF 的最終需求節點。MNF 中的需求節點可能為一個或多個，此類型的節點只有流入弧，同樣為了保持流量守恆限制，對於節點 $i \in N_T$ 給予 x_i 代表流入量總和。

$$x_i = \sum_{j \in E(i)} x_{ji} \quad \forall i \in N_T \quad (2.7)$$

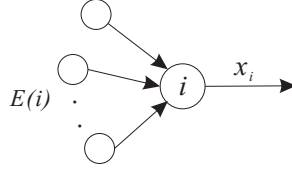


圖 2.6: MNF : T 節點 (來源 :Fang and Qi, 2003)

3. 轉運節點(O 節點)：屬於O 節點的集合以 N_O 表示，為 MNF 中的中介節點。其定義基本上同 MCF 中的轉運節點，其流出量等於流入量，且淨流量為 0。

$$\sum_{j \in E(i)} x_{ji} = \sum_{j \in L(i)} x_{ij} \quad \forall i \in N_O \quad (2.8)$$

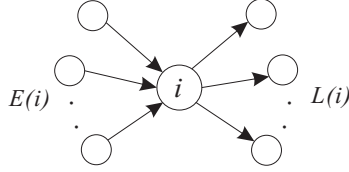


圖 2.7: MNF : O 節點 (來源 :Fang and Qi, 2003)

4. 庫存節點(I 節點)：屬於I 節點的集合以 N_I 表示，為 MNF 中儲存成本 / 半成品的節點。其基本定義類似 O 節點，但節點的流入量大於流出量，存在部分流量囤積於此節點。對於節點 $i \in N_I$ ，給予 x_i 表示囤積在此節點的數量。

$$\sum_{j \in E(i)} x_{ji} = x_i + \sum_{j \in L(i)} x_{ij} \quad \forall i \in N_I \quad (2.9)$$

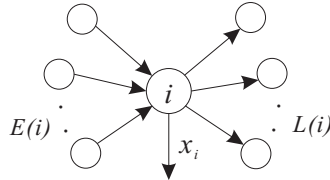


圖 2.8: MNF : I 節點 (來源 :Fang and Qi, 2003)

5. 分流節點(D節點)：屬於D節點的集合以 N_D 表示，為MNF中負責分散物料的節點，如產品的分解，或是原料的分餾等等。對於節點 $i \in N_D$ 而言， $E(i) = \{i^*\}$ ，只有一條流入弧 (i^*, i) ，而有多條流出弧，且流出弧的流量必須根據流入弧的流量做固定比例的分配。 k_{ij} 為一個介於(0,1)之間的實數，表示弧 (i^*, i) 對弧 (i, j) 所分配之比例；另外，為了不與流量守恆限制式相衝突， k_{ij} 的總和必須為1。

$$\begin{aligned} x_{i^*i} &= \sum_{j \in L(i)} x_{ij} \quad \forall i \in N_D \\ x_{ij} &= k_{ij} x_{i^*i} \quad \forall j \in L(i) \quad \forall i \in N_D \end{aligned} \quad (2.10)$$

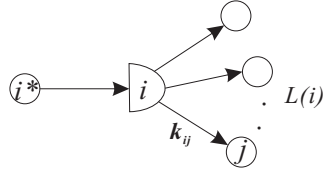


圖 2.9: MNF : D 節點 (來源:Fang and Qi, 2003)

6. 合流節點(C節點)：屬於C節點的集合以 N_C 表示，為MNF中負責集合物料的節點，如產品的裝配等等。對於節點 $i \in N_C$ 而言， $L(i) = \{i^*\}$ ，只有一條流出弧 (i, i^*) ，而有多條流入弧，流入弧的流量必須依照一固定比例流至節點 i 。 h_{ji} 為一個介於(0,1)之間的實數，為弧 (j, i) 與弧 (i, i^*) 之比例；同樣為了不與流量守恆限制式相衝突， h_{ji} 的總和必須為1。

$$\begin{aligned} \sum_{j \in E(i)} x_{ji} &= x_{ii^*} \quad \forall i \in N_C \\ x_{ji} &= h_{ji} x_{ii^*} \quad \forall j \in E(i) \quad \forall i \in N_C \end{aligned} \quad (2.11)$$

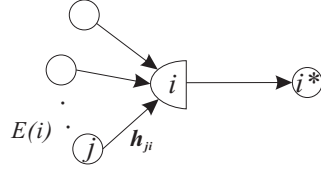


圖 2.10: MNF : C 節點 (來源 :Fang and Qi, 2003)

定義出 MNF 節點類型之後，Fang and Qi(2003) 基於分配網路的特性，考慮單一供應商 (節點 s)，多個需求節點，利用 (2.6)、(2.7)、(2.8) 與 (2.10) 四種節點的定義式定義 MDCP。MDCP 建構在 $G = (N, A)$ 網路中， N 表示節點集合， A 表示弧集合，其中 $|N| = n$ 且 $|A| = m$ 。其模式可列式如下：

$$\min \quad c_s x_s + \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (2.12)$$

$$\text{s.t.} \quad x \in F_D \quad (2.13)$$

$$0 \leq x_{ij} \leq u_{ij} \quad \forall (i, j) \in A \quad (2.14)$$

$$x_i \geq d_i \quad \forall i \in N_T \quad (2.15)$$

$$x_s \leq u_s \quad \text{for } s \quad (2.16)$$

(2.12) 代表 MDCP 之目標式， c_s 為物料單位成本， c_{ij} 為單位轉運成本；(2.13) 中的 F_D 代表滿足 (2.6)、(2.7)、(2.8) 及 (2.10) 四條限制式之流量變數 x ，此式代表網路中的所有流量須滿足此四種節點之限制；(2.14) 則代表弧容量限制式。另外，此模式由於 D 節點之影響，造成某些 T 節點可能會接收到比原來的需求量下限還多的流量，所以分別利用 (2.15) 與 (2.16) 代表需求節點的最小需求與產能上限，而用來表示 $x_i : i \in N_T$ 與 x_s 之弧均只有一端與節點相連，這些均和以往 MCF 的表示方式有所不同。圖 2.11 為一 MDCP 範例其最佳解分佈情形，假設此範例中每條弧無容量上限， u_1 代表節點 1 供給上限， d_5 、 d_6 與 d_7 代表節點 5、節點 6 與節點 7 之需求下限。在此範例中，由於流量必須滿足 D 節點的比例限制，造成節點 5 與節點 7 所接收之流量均比其最小需求量大為多，系統總需求其實只需要 2.8 單位，但是

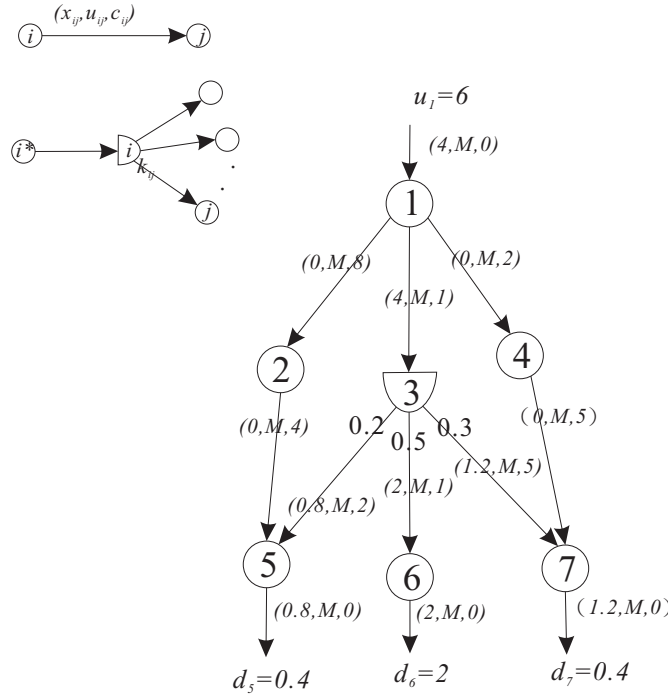


圖 2.11: 在 MDCP 中，T 節點可能接收多於其需求下限的流量值

卻從供給節點送出 4 單位；因此在 Fang and Qi(2003) 所提出的 MDCP 中，其流入與流出的總量不似以往 MCF 般可直接由節點的流量守恆限制式所判斷。

另外，若 MDCP 中存在兩個相鄰的 D 節點，可對網路圖做進一步簡化 (Lin, 2005; Yang, 2005)，使之成為任兩 D 節點不相鄰，且規模較小的網路問題。本研究在不失一般性的條件下，假設所討論的 MDCP 皆已經過簡化。Yang(2005) 求解三類無弧容量限制之 MDCP 的特例問題 UMDCP (Uncapacitated Minimum Distribution Cost Problem)，該問題求解如何在需求點需求下限為 1 單位的情況下，從供應點運送流量使其總流通成本最小；Lin(2005) 與 Ting(2005) 則探討加入 D 節點之後的最大流量問題。這些問題皆可視為 MDCP 之特例問題。

2.2.2 MDCP 求解方法

Fang and Qi(2003) 定義出 MDCP，並嘗試探討 MDCP 的基解架構。為了簡化分析，他們假設模式無弧容量限制 ($u_{ij} = \infty$)、網路中每條弧之單位流量成本值均為非負 ($c_{ij} \geq 0$)，且無產能限制 ($u_s = \infty$)，並令每一個 T 節點 i 的需求量必須為正 ($d_i > 0$)；除此之外，亦將 T 節點的流量守恆限制式 (2.7) 與 T 節點的需求下限

限制式 (2.15) 做進一步的簡化，加入超額變數 y_i ，使 $\sum_{j \in E(i)} x_{ji} - y_i = d_i$ ，以 y_i 做為超出節點需求的流量，藉此探討出無容量限制 MDCP 之基解特性。因此，如果網路中有 n 個節點，其中包含 p 個 D 節點，每個 D 節點 i 有 q_i 個流出弧， $q = \sum_{i \in N_D} q_i$ ， G_B 為其基解圖，則：

1. G_B 之弧個數為 $n + q - p$ 。
2. G_B 為一節點相連之圖形。
3. G_B 中的任何一個迴圈經過的節點至少包括一個 D 節點。
4. 每個 D 節點 i 必與 q_i 或 $q_i + 1$ 條基弧相連。
5. G_B 在適當移除 $q - y$ 條弧之後可形成一展開樹 (y 表示 $y_i : i \in N_T$ 為基底之個數)。

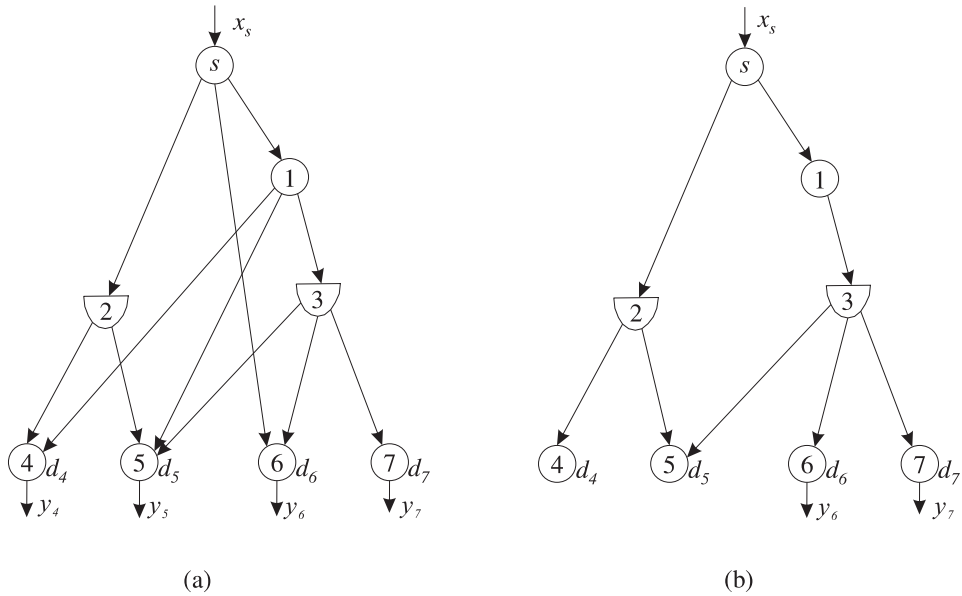


圖 2.12: MDCP 範例與其基解圖

圖 2.12(a) 為 Fang and Qi(2003) 所舉之 MDCP 圖例，圖中改以 y_i 來表示超出節點需求的流量，而圖 2.12(b) 為此範例的基解圖；由於 $n + \sum_{i \in N_D} q_i - p = 8 + 5 - 2 = 11$ ，故其基弧數目為 11，且迴圈由 D 節點所引起。

在分析基解架構之後，Fang and Qi(2003)即根據相關之假設提出一套網路單形法，其概念原則上如同單形法般，亦可分為幾個步驟描述：

步驟1: 計算對偶變數。

步驟2: 檢查其最佳解條件(對偶解可行為其最佳解條件)，選擇違反對偶解可行之弧為進入弧，若對偶解均可行時則為最佳解。

步驟3: 依據樞紐圖選擇流量最先到達下界之弧為退出弧，回步驟1。

Fang and Qi(2003)建構出求解MDCP的網路單形法的基本架構，提出求解主問題基變數與對偶變數相關的演算法，但此兩程序皆缺乏有系統的圖形化求解步驟。另外，由於其模型主要建構在無容量限制之假設下所做的分析，使其所提出的求解方法與模型特性不完全適用於具弧容量限制的網路。因此，本研究決定重新探討並修正其不足之處，除了將探討具容量限制的MDCP模式外，亦將以更多圖形化操作之概念探討諸如對偶變數計算、流量疊代、對偶變數更新等等與網路單形法中相關之程序，以使整個網路單形法更為完備。

Cohen and Megiddo(1994)亦曾探討是否能在強多項式求解時間內求解固定比率流量問題(fixed-ratios flow problem)，該問題擁有與MDCP類似的分流比例限制式；然而，他們的研究結果對於求解MDCP並無太大的幫助。首先，該論文僅證明在分流限制式數目為一固定常數時，此類含有分流限制式的最小成本流量問題或是最大流量問題才可在強多項式時間內求解；然而，本研究所探討的MDCP其分流限制式的數目並不固定，因此比該論文所探討的問題更為困難。另外，該論文僅以數學方式證明出強多項式演算法之存在性，並未實際地發展出一套專門求解該類問題的演算法以供探討或應用。

2.2.3 包含D節點之最短路徑問題求解方法

Yang(2005)引入D節點之特性探討UMDCP，並依D節點特性的不同定義出三種包含D節點的數學問題，稱之為UMDCP₁、UMDCP₂與UMDCP₃。在UMDCP₁與UMDCP₂中，由於D節點只選擇一條分流弧繼續分流，其他分流弧

的流量則囤積於網路中的節點，因此能夠修改 Dijkstra 演算法求解。但在求解 UMDCP₃ 時，以圖 2.12(a) 為例，在 $d_5 = 1$ ，而 $d_4 = d_6 = d_7 = 0$ 之條件下求解此問題的最小成本流量時，則無法成功修改 Dijkstra 演算法 (Dijkstra, 1959) 求解；該論文最後採取窮舉法來求解，並未有效地解決此類問題。

2.2.4 包含 D 節點之最大流量問題求解方法

如果將傳統最大流量問題引入 D 節點之特性，也可定義其最大流量問題。Ting(2005) 採取類似 Ford and Fulkerson(1956) 所提出之流量增益路徑之概念，修改 DFS(Depth-First Search Algorithm) 求解此類問題。首先採取 DFS，先尋找一條由起點至終點之路徑，而後針對 D 節點其他分支做處理直至搜尋出一組可推送流量的子圖，並計算此子圖可推送的流量；而當無法尋找到可推送流量之子圖時，即已求得最佳解。然而，此種搜尋方法的運算時間會隨著 D 節點分流個數呈指數成長，每當遇到一 D 節點 i 有 q_i 條分流，則需要做 $q_i!$ 次之搜尋才能確保該節點無法通過流量。Lin(2005) 則嘗試修改預流推進法求解含 D 節點的最大流量問題，其雖然無法成功地解決此問題，但亦提出了一些此類問題的對偶性質之觀察。說明在最大流量問題到達最佳解時，如果依照對偶變數的數值做分群，可能會有兩組以上的分群，與傳統最大流量問題僅有兩组分群有顯著的不同。

2.2.5 MDCP 其他相關問題求解方法

Mo et al. (2005) 嘗試引用多元商品 (multicommodity) 的概念，將 MNF 中不同的物料視為不同層之物品，D 節點及 C 節點則視為多元商品之間的轉換節點，運送至 D 節點或是 C 節點之流量會轉換至其他層商品的來源。該問題的目標函式同樣為最小化總系統成本，但是在做多元商品的轉換時，做了一些簡化問題的假設；假設 C 節點的來源點均由同一節點而來，D 節點的流出點亦為同一節點，而且同一類 D 節點或是 C 節點其分流比例皆相同。因此他們所求解的問題其實為 MNF 中的特殊問題，實質上仍未真正求解一般化的 MNF。

2.3 小結

MDCP 與 MCF 最大不同之處在於 D 節點之引入，因此可將 MDCP 視為 MCF 的延伸；如欲修改原先求解 MCF 的演算法，亦相當於擴大原本演算法的適用範圍，而必須一併考慮 D 節點對基解架構所帶來的改變。以 Dijkstra 演算法為例，其選擇節點與更新節點標籤之步驟皆建構在前一步驟為最佳解之情況下而進行；所以若將 Dijkstra 演算法求解之概念套用在求解 UMDCP 時，將因為 D 節點產生之迴圈，使得演算法無法確定其前一步驟是否為最佳解，因而無法直接適用。另外，預流推進法中預流的概念可視為對偶單形法之應用，因此在修改此法求解包含 D 節點的最大流量問題時，也需考慮其在對偶問題的意義。Fang and Qi(2003) 雖然提出求解 MDCP 的網路單形法，但僅適用於無容量限制之模式，亦缺乏一套有系統的圖形化求解方式；所以本研究嘗試重新檢視 MDCP 的基解特性，探討具容量限制 MDCP 基解性質，以發展一套可行且有系統的圖形化網路單形法來求解此類的網路問題。

第三章

模式轉換、主問題與對偶問題

MDCP 由於 D 節點之影響，使得模式無法以確切值表示 S 節點與 T 節點的淨流量值，並視表示供給量與需求量之弧流量為決策變數的一部份，此種方式亦造成這些弧均只有一端與節點相連，而另一端並無連通至其他節點，與傳統網路系統的表示方法有所差異。同時，若採用原分析方式套用於具弧容量限制的網路問題，則不能確保 MDCP 基解圖的相連性。故本研究在探討具容量限制的網路架構之前，先以單一 S 節點、多個 T 節點之 MDCP 為基礎，對模式做適當的轉換，使得 MDCP 模式不但能以確切值來表示每一個節點的淨流量值，亦能使得模式中所有弧的兩端均與節點相連，以符合傳統最小成本網路的表示方法。之後再探討加入容量限制後 MDCP 的基解架構以及該問題的對偶可行解特性，並依此做為網路單形法的最佳解條件。

3.1 模式轉換

Fang and Qi(2003) 雖然建構 MDCP 的基本模式，但其問題卻是在無容量上限、網路成本值非負及 T 節點之需求嚴格限制為正之假設條件下進行分析，並依此提出模式的基解特性。與 MCF 的基解圖形相同，無容量上限 MDCP 的基解圖基本上為一個連結圖 (connected graph)。然而，由於原模式中並非所有弧之兩端皆與節點相連，所以若單純採用上界技巧於模式中，有可能會因為某條弧之流量為到達上界的非基變數，而造成基解圖的分群，破壞他們所提出的基解圖之連結架構。圖 3.1(a) 為一個具流量上限的 MDCP 其流量分布情形，其中弧 (1,2) 之容量上限為 4，其他弧則不設限 (以 M 表示)；圖 3.1(b) 為其相對應的基解圖。在此圖

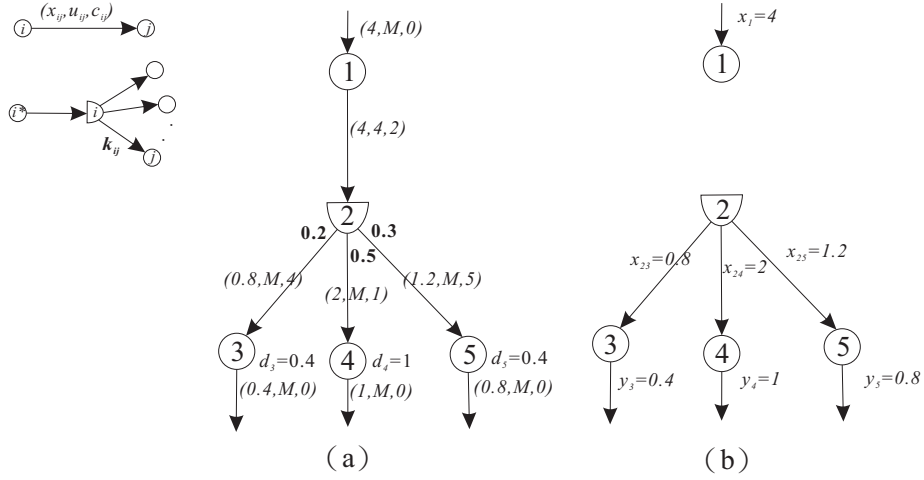


圖 3.1: 原 MDCP 模式套用上界技巧造成圖形分離

例中，由於弧(1,2)上的流量 x_{12} 為到達上界的非基變數，使得此圖形變成一個非連結圖。另外，雖然採用新增節點之方式可轉換MDCP的流量上限限制，但是此種方式不但會使得網路新增的節點個數依照其弧個數呈比例增加，亦會破壞原本節點的定義；例如原本的O節點在轉換過後會使得其淨流量值大於0且僅有流出弧，而T節點亦會因此而出現用來表示需求量之外的流出弧，連帶使得轉換過後的網路圖亦無法保證其圖形的相連性。因此本研究計畫在最小化系統成本的目標下，以單一S節點 s 的供給量通過網路以滿足T節點需求，並在確保問題有可行解的條件下($u_s \geq \sum_{i \in N_T} d_i$)，進行模式轉換，以便能確保基解圖為一個連結圖。

首先，新增一個虛擬節點 t 以做為轉換的基礎，並將節點 s 的產能上限限制式(2.16)加入超額變數 x_{st} ，亦即新增無容量上限之弧 (s, t) ，使其變為：

$$\begin{aligned} x_s &\leq u_s && \text{for } s \\ \implies x_s + x_{st} &= u_s && \text{for } s \end{aligned} \quad (3.1)$$

再利用(3.1)代換節點 s 的流量守恒限制式(2.6)，並令 $L(s) = L(s) \cup \{t\}$ ，可得：

$$\begin{aligned} \sum_{i \in L(s)} x_{si} - x_s &= 0 && \text{for } s \\ \implies \sum_{i \in L(s)} x_{si} - (u_s - x_{st}) &= 0 && \text{for } s \\ \implies \sum_{i \in L(s)} x_{si} + x_{st} &= u_s && \text{for } s \end{aligned} \quad (3.2)$$

之後引用類似的程序，對所有節點 $i \in N_T$ 之需求下限限制式 (2.15) 加入超額變數 x_{it} ，亦即新增弧 (i, t) (u_{it} 可設為 T 節點的超收量上限)，可得：

$$\begin{aligned} x_i &\geq d_i && \text{for } i \in N_T \\ \implies x_i &= d_i + x_{it} && \text{for } i \in N_T \end{aligned} \quad (3.3)$$

再利用 (3.3) 代換 (2.7)，並令 $L(i) = \{t\}$ ，可得：

$$\begin{aligned} x_i - \sum_{j \in E(i)} x_{ji} &= 0 && \text{for } i \in N_T \\ \implies x'_i + d_i - \sum_{j \in E(i)} x_{ji} &= 0 && \text{for } i \in N_T \\ \implies x_{it} - \sum_{j \in E(i)} x_{ji} &= -d_i && \text{for } i \in N_T \end{aligned} \quad (3.4)$$

由於 MDCP 模式除了 S 節點與 T 節點外，其他節點之淨流量皆為 0，代表送至需求節點之流量必定來自供給節點，可得出 $x_s = \sum_{i \in N_T} x_i$ 。而新增的節點 t 為一個僅有流入弧的節點，其流入量為 $\sum_{i \in N_T} x_{it} + x_{st}$ ，且 $E(t) = N_T \cup \{s\}$ 。因此，利用 (3.1) 與 (3.3) 可推得節點 t 的淨流量值如下：

$$\begin{aligned} - \sum_{i \in N_T} x_{it} - x_{st} &= \left(\sum_{i \in N_T} d_i - \sum_{i \in N_T} x_i \right) - (-x_s + u_s) \\ &= \sum_{i \in N_T} d_i - \sum_{i \in N_T} x_i + x_s - u_s \\ &= \sum_{i \in N_T} d_i - u_s \end{aligned} \quad (3.5)$$

經由模式轉換後，所有的節點均能夠以一確切值來表示其淨流量值，除原先 O 節點與 D 節點的淨流量值為 0 之外，我們可得節點 s 的淨流量值 $b_s = u_s$ ，T 節點 i 的淨流量 $b_i = -d_i$ ，新增的虛擬節點 t 其淨流量值 $b_t = \sum_{i \in N_T} d_i - u_s \leq 0$ ，可視為 T 節點的一部份，使得 $N_T = N_T \cup \{t\}$ 。圖形上之轉變如圖 3.2 與圖 3.3 所示。

引用相同的概念，亦可將多個 S 節點、多個 T 節點的 MDCP 轉換為一對多的 MDCP。首先，新增一個節點 s ，並新增弧 (s, i) 連通到所有節點 $i \in N_S$ ，而且將 u_{si} 設為原先節點 $i \in N_S$ 的產能上限 u_i ，再令節點 s 的淨流量值 $u_s = \sum_{i \in N_S} u_i$ ，而原

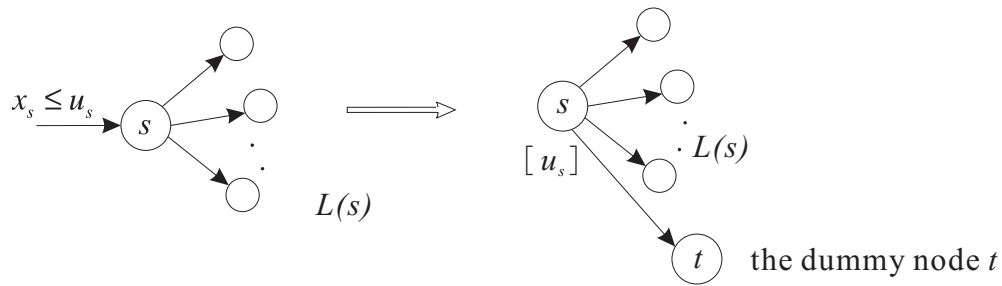


圖 3.2: MDCP-S 節點之轉換

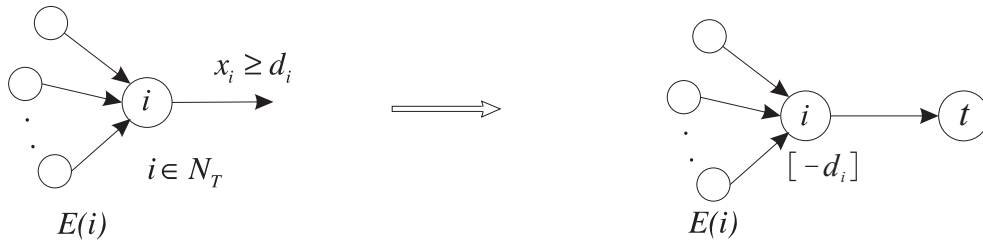


圖 3.3: MDCP-T 節點之轉換

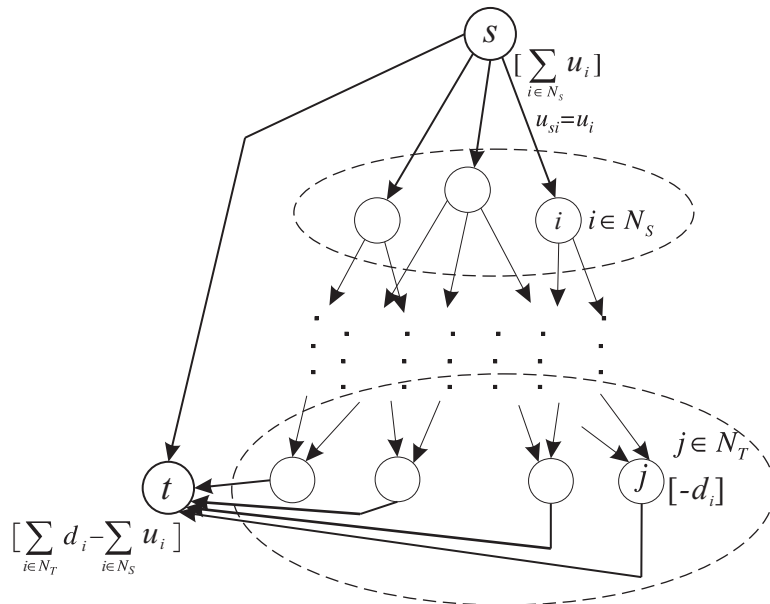


圖 3.4: 新增節點 s 與 t 以轉換多個供給點、多個需求點之 MDCP 網路

先 S 節點經轉換後則轉變為淨流量為 0 的 O 節點；之後同樣以上述新增虛擬節點 t 的轉換方式即可轉換原問題為新的一對多 MDCP，轉換後之圖形如圖 3.4 所示

不論是一對多或是多對多的 MDCP，在轉換所有相關節點之後，皆可轉換為新的一對多的 MDCP 模式。因此如同樣以 $G = (N, A)$ 表示轉換後的網路系統，

N 代表節點集合， A 代表弧集合，且 $|N| = n$ ， $|A| = m$ ，可列出一對多的 MDCP 模式如下：

$$\begin{aligned}
& \min \sum_{(i,j) \in A} c_{ij} x_{ij} && (\text{MDCP}_{new}) \\
& \text{s.t.} \quad \sum_{i \in L(s)} x_{si} = u_s && \text{for } s \\
& \quad \sum_{j \in L(i)} x_{ij} - \sum_{j \in E(i)} x_{ji} = 0 && \forall i \in N_O \cup N_D \\
& \quad x_{it} - \sum_{j \in E(i)} x_{ji} = -d_i && \forall i \in N_T, i \neq t \\
& \quad -x_{st} - \sum_{j \in N_T, j \neq t} x_{jt} = \sum_{i \in N_T} d_i - u_s && \text{for } t \text{ (dummy node)} \\
& \quad x_{ij} - k_{ij} x_{li} = 0 && \forall i \in N_D, (l, i) \in A \\
& \quad 0 \leq x_{ij} \leq u_{ij} && \forall (i, j) \in A
\end{aligned}$$

在新的 MDCP 模式中，所有節點的流量守恒限制式與 MCF 相同，其淨流量值皆能以一確切值表示，且所有弧的兩端皆與節點相接。相較於原 MDCP 模式，轉換後之網路顯得較簡單且有彈性，除了所有物料製造成本皆能以弧之單位流量成本表示外，對於 T 節點 i 而言，多於 d_i 之流量 x_{it} 亦可加入 c_{it} 做為超收需求量所需付出的成本值。另外，由於此時 S 節點、O 節點與 T 節點之淨流量值皆能以一確切值來表示，且圖形上皆以圓圈型態表示，與傳統 MCF 的表示方法相同；為了方便之後的討論分析，在此先將此三類節點合稱為 \hat{O} 節點，並以 $N_{\hat{O}}$ 表示此類節點集合，使得 $N_{\hat{O}} = N_S \cup N_O \cup N_T$ ，而 $N = N_{\hat{O}} \cup N_D$ 。

3.2 主問題基解架構

上一節將原 MDCP 模式轉換如 (MDCP_{new}) 所示，其特性也將隨著模式轉換而有所修正。首先，我們可以將 (MDCP_{new}) 以矩陣形式表示如下：

$$\left. \begin{aligned}
& \min \quad cx \\
& \text{s.t.} \quad Fx = b \\
& \quad \quad Dx = 0 \\
& \quad \quad 0 \leq x \leq u
\end{aligned} \right\} \quad (3.6)$$

x 代表的是弧向量， c 為成本向量， F 為流量守恒限制式所構成的node-arc incidence矩陣， b 代表淨流量值構成之向量，而 D 則為D節點其分流限制式矩陣。

在使用單形法求解時，我們通常假設函數限制式矩陣為全秩(full rank)矩陣，如果函數限制式矩陣不為全秩矩陣時，則代表該問題包含多餘的限制式。一般遇到不為全秩矩陣時有兩種做法：一為去除多餘限制式；二為加入適當的人工變數(artificial variable)使函數限制式矩陣變為全秩矩陣。而此處我們採取第一項做法，探討函數限制式的秩數為何，方能了解MDCP的基解個數。

(3.6)中 $Fx = b$ 乃表示節點的流量守恒限制式，經過3.1節模式轉換後，此限制式與以往MCF的流量限制式相同，因此如果網路中有 n 個節點， F 矩陣之秩數必為 $n - 1$ 。而在(3.6)下半部中， $Dx = 0$ 為D節點分流限制式，代表MDCP中每一個節點 $i \in N_D$ 其流出弧與流入弧流量之比例，假如 q_i 為某一D節點 i 流出弧個數，則該節點需要 q_i 條分流限制式定義其比例關係，而總共有 $q = \sum_{i \in N_D} q_i$ 條分流限制式。對於任一個D節點 i 而言，其分流限制式之總合可組成該節點的流量守恒限制式($\sum_{j \in L(i)} x_{ij} = \sum_{i \in N_D} k_{ij} x_{i*i} = x_{i*i}$)；因此在所有與節點 i 相關的 $q_i + 1$ 條限制式中(流量守恒限制式加上分流比例限制式)，即包含一條多餘的函數限制式，而 p 個節點則有 p 條多餘限制式。綜合以上之結果，表示MDCP中的 $n + q$ 條流量守恒限制式與分流限制式中，共有 $p + 1$ 條多餘的函數限制式，因此該矩陣之秩數為 $n + q - p - 1$ ，亦為MDCP的基解個數。

Lemma 1. 在一個MDCP中，若 $|N| = n$ ， $|N_D| = p$ ，D節點 i 的流出弧個數 $|L(i)| = q_i$ ， $\sum_{i \in N_D} q_i = q$ 為。令 x 為(3.6)所對應的可行基解，而 G_B 為 x 所代表的基解圖形，則 G_B 擁有下列特性：

- (i) 基弧個數為 $n + q - p - 1$ 。
- (ii) 基解圖為一個連結圖。
- (iii) G_B 中的任何一個迴圈至少包括一個D節點。
- (iv) 每個D節點 i 必與 q_i 或 $q_i + 1$ 條基弧相連。

(v) 在 G_B 中若適當移除與每一個 D 節點 i 相連的 $q_i - 1$ 條弧後，可將 G_B 縮減為一個展開樹。

Proof:

- (i) 由前所述，因為 MDCP 中 F 與 D 矩陣之總秩數為 $n + q - p - 1$ ，故其基弧個數為 $n + q - p - 1$ 。
- (ii) 由於模式中每條弧之兩端均與節點相連，因此 $Fx = b$ 限制式與 MCF 的流量守恆限制式相同。傳統 MCF 之基解圖形為一個連結的展開樹 (Dantzig, 1951)；而 MDCP 模式加入 $Dx = 0$ 之限制式後，將增加 $q - p$ 條弧為基底而使基解圖產生迴圈，並不會破壞圖形的相連性，因此 G_B 為一個連結圖。
- (iii) 同 Fang and Qi(2003)，由 (i) 可知基解個數為 $n + q - p - 1 \geq n - 1$ ，所以 G_B 中必包含迴圈。在 MCF 中，其基解所對應圖形為一個展開樹 (Dantzig, 1951)。故 G_B 中任何一個迴圈勢必由 D 節點而引起，且至少經過一個 D 節點。
- (iv) 同 Fang and Qi(2003)。在 G_B 中，每個 D 節點 i 最多與 $q_i + 1$ 條弧相連，因此最多所有相連弧 ($q_i + 1$ 條弧) 皆為基底；又每個 D 節點 i 由 q_i 條有效限制式所構成，故最少應包含 q_i 條相鄰弧為基底。
- (v) 承 (ii)， G_B 中的迴圈由 D 節點所引起，又由 (i) 之基解總個數為 $n + q - p - 1$ ；因此 G_B 若適當移除每個 D 節點造成迴圈的 $q_i - 1$ 個基弧，共 $q - p$ 條弧後，則圖形可縮減為一個展開樹。

□

圖 3.5(a) 為轉換模式過後的 MDCP 範例，圖 3.5(b) 為其對應基解圖，該圖形為一個相連圖，基弧個數為 $n + q - p - 1 = 10$ ，圖中迴圈皆經過 D 節點，節點 2 與 $q_2 = 2$ 條弧相接，節點 3 與 $q_3 + 1 = 4$ 條弧相接；而扣除弧 (2,4)、弧 (3,6) 與弧 (3,7) 後，圖形可縮減為一個展開樹。

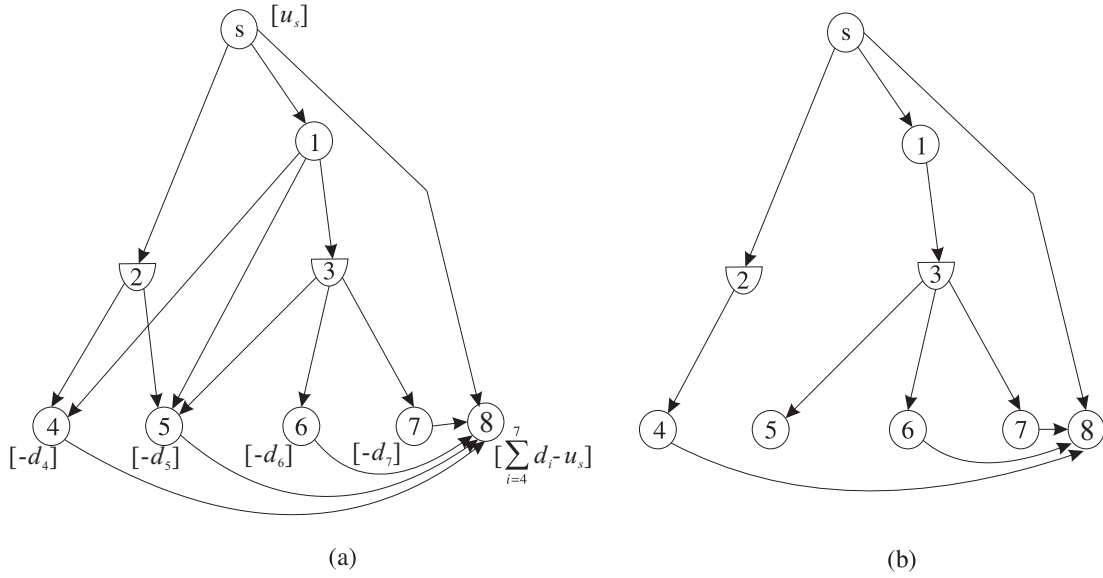


圖 3.5: 轉換後 MDCP 範例與其基解圖

3.3 對偶問題與最佳解條件

MCF 在面對具有容量上限之網路問題時有兩種做法：1. 以新增節點之方式轉換模式為無容量上限的網路；2. 使用上界技巧套用至網路問題。此兩種方式基本上均是對 $x_{ij} \leq u_{ij}$ 做處理，差別在於新增節點之方式是屬於對網路圖的先處理程序，而上界技巧之處理方式乃在於當變數達上界時才額外處理，並使得到達上界或下界之非基變數各有其最佳解條件(2.1.3節)。因此，為了能夠了解 MDCP 加入容量限制後對於對偶問題目標式的影響，本節選擇先將容量上界限制式做為一般函數限制式，之後再由此推導容量上界於 MDCP 對偶問題的影響。

若依照弧 (i, j) 為基變數與否，可將 A 切割成三個互斥的子集合使得 $A = B \cup L \cup U$ ，其中由基弧構成之集合為 B ，流量達下界的非基弧所構成之集合為 L ， U 則代表由達上界的非基弧構成之集合。另外，若依照弧之兩端節點類型為準則，可將 A 以不同之方式切割為三個互斥的子集合，使得 $A = A_{\hat{O}\hat{O}} \cup A_{\hat{O}D} \cup A_{D\hat{O}}$ ， $A_{\hat{O}\hat{O}}$ 為弧兩端均為 \hat{O} 節點構成之集合； $A_{\hat{O}D}$ 代表弧之頭端(head)為 D 節點，而尾端(tail)為 \hat{O} 節點構成之集合； $A_{D\hat{O}}$ 則為弧之頭端為 \hat{O} 節點，而尾端為 D 節點構成之集合。由於本研究問題為精簡後之網路問題，故兩端皆為 D 節點之弧不在討論

之列。此外，為了分析方便，假設MDCP中所有弧 $(i, j) \in A$ 皆有容量上限 u_{ij} (原本不具容量上限之弧可設其 u_{ij} 為一足夠大之正數)。

首先，先將弧容量限制式視為一般函數限制式，並同樣以新增節點之方式轉換網路圖，並改以 ij 共同表示處理弧 $(i, j) \in A$ 之容量上限新增的節點，其流量守恒限制式為：

$$-x_{ij} - s_{ij} = -u_{ij} \quad \forall (i, j) \in A \quad (3.7)$$

原本節點 $i \in N$ 的流量守恒限制式變為：

$$\sum_{j \in L(i)} x_{ij} + \sum_{(j, i) \in A} s_{ji} = \sum_{(j, i) \in A} u_{ji} + b_i \quad \forall i \in N \quad (3.8)$$

b_i 為原本節點的淨流量值，節點 s 淨流量值為 u_s ，節點 $i \in N_T, i \neq t$ 之淨流量值為 $-d_i$ ，T節點 t 的淨流量值為 $\sum_{i \in N_T, i \neq t} d_i - u_s$ ，其他節點之淨流量值則為0；而在轉換容量限制後，每個節點的淨流量值均增加 $\sum_{(j, i) \in A} u_{ji}$ 。D節點 i 的分流限制式則保持不變，即使轉換過後弧 (i^*, i) 並非節點 i 的流入弧(轉換前 $E(i) = \{i^*\}$)，但同樣對 x_{i^*i} 與 x_{ij} 的流量比例設限。

線性規劃中的主問題皆有其特殊對應的對偶問題，網路單形法亦是利用對偶性質做為其最佳解條件。理論上每一個節點的流量守恒限制式與D節點的分流限制式皆可對應到特定的對偶變數，但為了區分起見，節點 $i \in N_D$ 對應到對偶變數 $\tilde{\pi}_i$ ，而節點 $i \in N_{\hat{O}}$ 對應到 π_i ；另外，D節點 i 的分流限制式 $k_{ij}x_{i^*i} - x_{ij} = 0$ 可對應到對偶變數 v_{ij} ，(3.7)則對應到對偶變數 y_{ij} 。如此一來，MDCP對偶問題的目標式就可表示為：

$$\begin{aligned} \max \quad & u_s \pi_s + \left(\sum_{i \in N_T, i \neq t} d_i - u_s \right) \pi_t - \sum_{i \in N_T, i \neq t} d_i \pi_i + \sum_{i \in E(j)} \sum_{j \in N_D} u_{ij} \tilde{\pi}_j \\ & + \sum_{i \in E(j)} \sum_{j \in N_{\hat{O}}} u_{ij} \pi_j - \sum_{(i, j) \in A} u_{ij} y_{ij} \end{aligned} \quad (3.9)$$

(3.9)中第一項為節點 s 淨流量值 b_s 與其對偶變數 π_s 之乘積；第二項為節點 t 淨流量值 b_t 與其對偶變數 π_t 之乘積；第三項為節點 $i \in N_T, i \neq t$ 之淨流量值 b_i 與其對偶變數 π_i 之乘積總合；第四項為D節點轉換過後淨流量值改變量(即為原本之容

量上限 u_{ij} 與其對偶變數 $\tilde{\pi}_i$ 之乘積總合；第五項為 \hat{O} 節點轉換過後之淨流量值改變量與其對偶變數 π_i 之乘積總合；而最後一項為新增節點 ij 其淨流量值 u_{ij} 與對偶變數 y_{ij} 之乘積總合。由於網路中之弧為連接兩節點之「媒介」，而對偶變數則為與節點相關之變數，所以 MDCP 的對偶問題限制式就如同 MCF 般，可轉變成為對每條弧 $(i, j) \in A$ 設限。依照弧之兩端點相連的節點類型可將之分為三類：

1. 尾端與頭端皆為 \hat{O} 節點之弧；
2. 尾端為 \hat{O} 節點而頭端為 D 節點之弧；
3. 尾端為 D 節點而頭端為 \hat{O} 節點之弧。

依照此三種型態之弧分別探討其最佳解條件如下：

1. 尾端與頭端皆為 \hat{O} 節點之弧：此類型之弧轉換後如圖 3.6 所示，而與弧 (i, ij) 、弧 (j, ij) 有關的對偶可行解限制式為：

$$\left. \begin{aligned} \pi_i - y_{ij} &\leq c_{ij} & \forall (i, j) \in A_{\hat{O}\hat{O}} \\ \pi_j - y_{ij} &\leq 0 & \forall (i, j) \in A_{\hat{O}\hat{O}} \end{aligned} \right\} \quad (3.10)$$

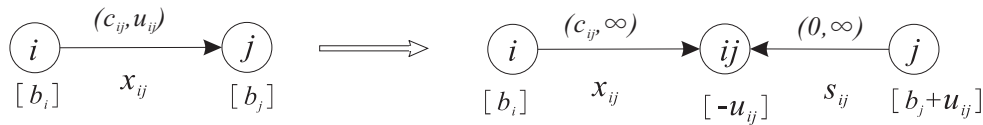


圖 3.6: MDCP 轉換容量限制之網路-兩端為 \hat{O} 節點之弧

(3.10) 第一式為在對偶問題中與 x_{ij} 相關之對偶變數所構成的限制式，第二式則為與 s_{ij} 相關之對偶變數所構成的限制式。由於新增之節點 ij 僅與 (i, ij) 、 (j, ij) 兩條弧相接，所以基於基解圖形的相連性，節點 ij 至少與節點 i 或與節點 j 相連，或是與兩節點 i 與 j 均相連。此即代表了三種情況： x_{ij} 與 s_{ij} 皆為基變數； x_{ij} 為非基變數，而 s_{ij} 為基變數；或是 x_{ij} 為基變數，而 s_{ij} 為非基變數。

若 x_{ij} 與 s_{ij} 皆為基變數，根據互補差額定理 (complementary slackness) 可得 (3.10) 中兩式必為臨界限制式 (tight constraint)，故可得出 $\pi_i - y_{ij} = c_{ij}$ 且 $\pi_j - y_{ij} = 0$ ，利用後式帶入前式可得 $\pi_i - \pi_j = c_{ij}$ ；代表原網路中若 x_{ij} 為基變數時所對應的對偶可行解條件。

若 x_{ij} 為非基變數， s_{ij} 為基變數 ($x_{ij} = 0, s_{ij} = u_{ij}$)，同樣根據互補差額定理，此時 $\pi_i - y_{ij} \leq c_{ij}$ 且 $\pi_j - y_{ij} = 0$ ，再利用後式代換前式可得 $\pi_i - \pi_j \leq c_{ij}$ 。此式代表為原網路中 x_{ij} 為到達下界之非基變數時，弧 (i, j) 所對應的對偶可行解條件。

若 x_{ij} 為基變數，而 s_{ij} 為非基變數 ($x_{ij} = u_{ij}, s_{ij} = 0$)，同樣根據互補差額定理，此時 $\pi_i - y_{ij} = c_{ij}$ 且 $\pi_j - y_{ij} \leq 0$ ，經代換過後則可得 $\pi_i - \pi_j \geq c_{ij}$ 。此式代表為原網路中 x_{ij} 為到達上界之非基變數時，弧 (i, j) 所對應的對偶可行解條件。

另外， π_i 、 π_j 與 y_{ij} 所對應的目標值也可依照節點 ij 與其連接弧的相連性分別討論，若原先之網路圖 x_{ij} 為基變數時，可利用 (3.10) 中第二項為臨界限制式，使得轉換過後與 u_{ij} 相關的對偶變數目標值為 $u_{ij}\pi_j - u_{ij}y_{ij} = u_{ij}(\pi_j - \pi_j) = 0$ ；同理，當 x_{ij} 為到達下界的非基變數時，其對偶變數目標值亦為 0；而當 x_{ij} 為到達上界的非基變數時，利用 (3.10) 中第一項為臨界限制式可使得 $u_{ij}\pi_j - u_{ij}y_{ij} = u_{ij}(\pi_j - \pi_i + c_{ij})$ ；統整後如下：

$$\left. \begin{aligned} u_{ij}\pi_j - u_{ij}y_{ij} &= 0 & \forall (i, j) \in A_{\hat{O}\hat{O}} \cap B \\ u_{ij}\pi_j - u_{ij}y_{ij} &= 0 & \forall (i, j) \in A_{\hat{O}\hat{O}} \cap L \\ u_{ij}\pi_j - u_{ij}y_{ij} &= u_{ij}(\pi_j - \pi_i + c_{ij}) & \forall (i, j) \in A_{\hat{O}\hat{O}} \cap U \end{aligned} \right\} \quad (3.11)$$

2. 尾端為 \hat{O} 節點，頭端為 D 節點之弧：此類型之弧轉換後之圖形如圖 3.7 所示，其對偶問題限制式為：

$$\left. \begin{aligned} \pi_i + \sum_{k \in L(j)} k_{jk}v_{jk} - y_{ij} &\leq c_{ij} & \forall (i, j) \in A_{\hat{O}D} \\ \tilde{\pi}_j - y_{ij} &\leq 0 & \forall (i, j) \in A_{\hat{O}D} \end{aligned} \right\} \quad (3.12)$$

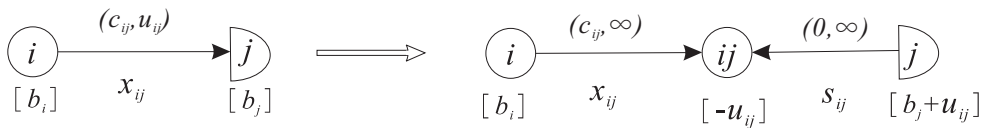


圖 3.7: MDCP 轉換容量限制之網路-尾端為 \hat{O} 節點，頭端為 D 節點之弧

(3.12) 第一式為對偶問題中與 x_{ij} 相關之對偶變數所構成的限制式，由於 D 節點 j 的分流限制式仍是對 x_{ij} 與 $x_{jk} : k \in L(j)$ 設限，因此 v_{jk} 歸類在與 x_{ij} 相關的限制式中；第二式則為 s_{ij} 所對應的對偶變數限制式。同樣藉由分析節點 ij 其相連情形，可得出三種情況：若原網路之 x_{ij} 為介於 0 到 u_{ij} 之基變數，其對偶可行解條件須滿足 $\pi_i - \tilde{\pi}_j + \sum_{k \in L(j)} k_{jk} v_{jk} = c_{ij}$ ；若 x_{ij} 為到達下界之非基變數，其對偶可行解條件須滿足 $\pi_i - \tilde{\pi}_j + \sum_{k \in L(j)} k_{jk} v_{jk} \leq c_{ij}$ ；若 x_{ij} 為到達上界之非基變數，其對偶可行解條件須滿足 $\pi_i - \tilde{\pi}_j + \sum_{k \in L(j)} k_{jk} v_{jk} \geq c_{ij}$ 。

π_i 、 $\tilde{\pi}_j$ 與 y_{ij} 所對應的目標值也可依照節點 ij 與其連接弧的相連性與臨界限制式的選擇，使得：

$$\left. \begin{aligned} u_{ij} \tilde{\pi}_j - u_{ij} y_{ij} &= 0 & \forall (i, j) \in A_{\hat{O}D} \cap B \\ u_{ij} \tilde{\pi}_j - u_{ij} y_{ij} &= 0 & \forall (i, j) \in A_{\hat{O}D} \cap L \\ u_{ij} \tilde{\pi}_j - u_{ij} y_{ij} &= u_{ij} (\tilde{\pi}_j - \sum_{k \in L(j)} k_{jk} v_{jk} - \pi_i + c_{ij}) & \forall (i, j) \in A_{\hat{O}D} \cap U \end{aligned} \right\} (3.13)$$

3. 尾端為 D 節點，頭端為 \hat{O} 節點之弧：此類型之弧轉換後之圖形如圖 3.8 所示，其對偶問題限制式為：

$$\left. \begin{aligned} \tilde{\pi}_i - v_{ij} - y_{ij} &\leq c_{ij} & \forall (i, j) \in A_{D\hat{O}} \\ \pi_j - y_{ij} &\leq 0 & \forall (i, j) \in A_{D\hat{O}} \end{aligned} \right\} (3.14)$$

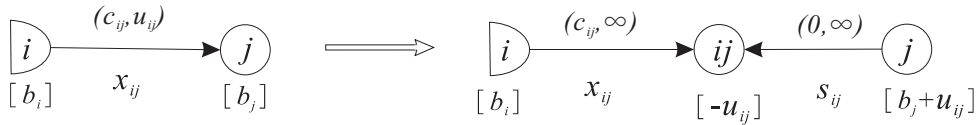


圖 3.8: MDCP 轉換容量限制之網路-尾端為 D 節點，頭端為 \hat{O} 節點之弧

(3.14) 第一式為在對偶問題中與 x_{ij} 相關之對偶變數所構成的限制式；第二式則為與 s_{ij} 相關的對偶問題限制式。同樣藉由分析節點 ij 其相連情形，可分析得出三種情況：若原網路之 x_{ij} 為介於 0 到 u_{ij} 之基變數，其對偶可行解條件須滿足 $\tilde{\pi}_i - v_{ij} - \pi_j = c_{ij}$ ；若 x_{ij} 為到達下界之非基變數，其對偶可行解條件須滿足 $\tilde{\pi}_i - v_{ij} - \pi_j \leq c_{ij}$ ；若 x_{ij} 為到達上界之非基變數，其對偶可行解條件須滿足 $\tilde{\pi}_i - v_{ij} - \pi_j \geq c_{ij}$ 。

$\tilde{\pi}_i$ 、 π_j 與 y_{ij} 所對應的目標值也可依照節點 ij 與其連接弧的相連性與臨界限制式的選擇，使得：

$$\left. \begin{aligned} u_{ij}\pi_j - u_{ij}y_{ij} &= 0 & \forall (i,j) \in A_{D\hat{O}} \cap B \\ u_{ij}\pi_j - u_{ij}y_{ij} &= 0 & \forall (i,j) \in A_{D\hat{O}} \cap L \\ u_{ij}\pi_j - u_{ij}y_{ij} &= u_{ij}(\pi_j - \tilde{\pi}_i + v_{ij} + c_{ij}) & \forall (i,j) \in A_{D\hat{O}} \cap U \end{aligned} \right\} \quad (3.15)$$

由於 MDCP 中每個 D 節點的分流比例係數總和必須為 1，此特性不但使得 MDCP 主問題有多餘分流限制式的出現，亦可依此進一步簡化對偶變數之性質。在之前對偶可行解條件中，若 D 節點 i 有 q_i 條分流弧，可對應到 $q_i + 1$ 個對偶變數 (q_i 個 v_{ij} 與一個 $\tilde{\pi}_i$)，利用 $\sum_{j \in L(i)} k_{ij} = 1$ ，可得：

$$\sum_{j \in L(i)} k_{ij}(\tilde{\pi}_i - v_{ij}) = \sum_{j \in L(i)} (k_{ij}\tilde{\pi}_i - k_{ij}v_{ij}) = \tilde{\pi}_i - \sum_{j \in L(i)} k_{ij}v_{ij} \quad (3.16)$$

因此，若令 $\rho_{ij} = \tilde{\pi}_i - v_{ij}$ ，則 (3.16) 最右項可簡化為 $\sum_{j \in L(i)} k_{ij}\rho_{ij}$ ，再令 $\pi_i = \sum_{j \in L(i)} k_{ij}\rho_{ij}$ 做為節點 $i \in N_D$ 之「統合」變數，即可將前述之對偶可行解條件統整簡化為：

1. 若兩端皆為 \hat{O} 節點，其對偶可行解條件為：

$$\left. \begin{aligned} \pi_i - \pi_j &= c_{ij} & \forall (i,j) \in A_{\hat{O}\hat{O}} \cap B \\ \pi_i - \pi_j &\leq c_{ij} & \forall (i,j) \in A_{\hat{O}\hat{O}} \cap L \\ \pi_i - \pi_j &\geq c_{ij} & \forall (i,j) \in A_{\hat{O}\hat{O}} \cap U \end{aligned} \right\} \quad (3.17)$$

2. 若尾端為 \hat{O} 節點，頭端為 D 節點，其對偶可行解條件：

$$\left. \begin{aligned} \pi_i - \pi_j &= c_{ij} & \forall (i,j) \in A_{\hat{O}D} \cap B \\ \pi_i - \pi_j &\leq c_{ij} & \forall (i,j) \in A_{\hat{O}D} \cap L \\ \pi_i - \pi_j &\geq c_{ij} & \forall (i,j) \in A_{\hat{O}D} \cap U \end{aligned} \right\} \quad (3.18)$$

3. 若尾端為 D 節點，頭端為 \hat{O} 節點，其對偶可行解條件：

$$\left. \begin{aligned} \rho_{ij} - \pi_j &= c_{ij} & \forall (i,j) \in A_{D\hat{O}} \cap B \\ \rho_{ij} - \pi_j &\leq c_{ij} & \forall (i,j) \in A_{D\hat{O}} \cap L \\ \rho_{ij} - \pi_j &\geq c_{ij} & \forall (i,j) \in A_{D\hat{O}} \cap U \end{aligned} \right\} \quad (3.19)$$

利用同樣的整合程序，可對(3.13)中的第三式之右項進一步簡化，使得：

$$\begin{aligned} & u_{ij}(\tilde{\pi}_j - \sum_{k \in L(j)} k_{jk} v_{jk} - \pi_i + c_{ij}) \\ = & u_{ij}(\pi_j - \pi_i + c_{ij}) \end{aligned} \quad (3.20)$$

而(3.15)中的第三式之右項變為：

$$\begin{aligned} & u_{ij}(\pi_j - \tilde{\pi}_i + v_{ij} + c_{ij}) \\ = & u_{ij}(\pi_j - \rho_{ij} + c_{ij}) \end{aligned} \quad (3.21)$$

統整(3.11)、(3.20)與(3.21)後，可使得對偶問題目標式變為：

$$\begin{aligned} \max \quad & u_s \pi_s + \left(\sum_{i \in N_T, i \neq t} d_i - u_s \right) \pi_t - \sum_{i \in N_T, i \neq t} d_i \pi_i + \sum_{(i,j) \in U \cap (A_{\hat{O}\hat{O}} \cup A_{\hat{O}D})} u_{ij}(\pi_j - \pi_i + c_{ij}) \\ & + \sum_{(i,j) \in U \cap A_{D\hat{O}}} u_{ij}(\pi_j - \rho_{ij} + c_{ij}) \end{aligned} \quad (3.22)$$

此式代表在具容量上限的MDCP模式的對偶問題目標式中，其值除了前三項所表示的 b_i 與 π_i 的乘積外，尚包含後兩項容量上界對於對偶問題目標式影響之值。

對於D節點 i 而言， $\pi_i = \sum_{j \in L(i)} \rho_{ij}$ 代表的是 π_i 與 $\rho_{ij} : j \in L(i)$ 呈相依關係。如圖3.9所示，每個D節點 i 以 π_i 做為D節點整體的對偶變數；另外有 q_i 個 ρ_{ij} 分別與D節點之流出弧相關，而 $k_{ij}\rho_{ij}$ 之總合即為 π_i 之值。運用此種表示方式可以簡化原本對偶性質複雜的陳述方式，改以較簡單的方式陳述MDCP模式到達上界或下界時的對偶可行解條件，此處亦是本研究與Fang and Qi(2003)不同之處。

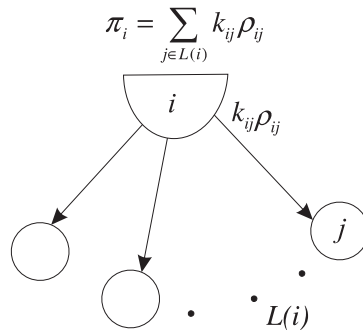


圖 3.9: D 節點之對偶變數關聯示意圖

3.4 小結

經由本章3.1節的推導過程，除了轉換原本MDCP模式，並使得MDCP網路中的每條弧兩端均有節點相連外，亦使得網路中的每一個節點皆可用一確切值來表達節點淨流量值。3.2節探討轉換過後MDCP模式的基解架構，此處免去了原本Fang and Qi(2003)在分析MDCP特性時所做的單位流量成本非負、無容量上限且需求下限嚴格為正之假設，以簡單之形式確保圖形的相連性，並探討轉換過後的基解性質。而3.3節探討MDCP對偶可行解特性，網路單形法亦是利用對偶可行解做為其最佳解條件。除此之外，本研究亦針對對偶變數做進一步的整合，以 ρ_{ij} 及 π_i 來整合與D節點有關的對偶變數，使對偶可行解條件能夠以清楚易懂的表示式表達；因此，下一個章節即可利用這些相關特性探討如何以圖形化之方式計算對偶變數、流量更新、對偶變數更新以及初始解設置等程序，建構出圖形化操作概念的網路單形法。

第四章

網路單形法

本研究所提出的網路單形法屬於主網路單形法 (primal network simplex algorithm)，其基本概念乃使每一步驟均維持在主問題的可行基解，並由互補差額定理 (complementary slackness)，亦即利用基變數必須滿足其對應的對偶偶臨界限制式 (tight constraint)，計算其相關的對偶變數；之後進行不斷的疊代過程以更新主變數與對偶變數，而當疊代至對偶可行解時，即可得到最佳解。為了能以圖形化方式探討網路單形法中的相關程序，本章導入基群集 (basic component) 做為圖形分割之依據，藉此將基解圖適當地分群，再以基群集為基礎建構出整體的網路單形法。本研究除了發展出一套有別於 Fang and Qi(2003) 的系統化對偶變數計算程序 (procedure: *compute_dual_variables*) 之外，並更進一步沿用基群集來推估網路單形法中的流量疊代程序 (procedure: *flow_update*)、對偶變數更新程序 (procedure: *dual_update*) 及初始解設立程序 (procedure: *initialization*) 等。本研究提出的網路單形法依照其操作概念，可列出基本步驟如下：

步驟0: 給定一初始基解 x 與基解圖 G_B 。

步驟1: 計算 G_B 對應的對偶變數 (procedure: *compute_dual_variables*)。

步驟2: 對每條非基變數之弧 $(i, j) \in L \cup U$ 檢查對偶可行解條件 (3.17-3.19)，選擇違反最佳解條件之弧 (k, l) 為進入變數；若皆滿足則為最佳解，演算法中止。

步驟3: 進行流量疊代，並選擇退出弧 (v, z) (procedure: *flow_update*)。

步驟4: 更新對偶變數，回步驟2 (procedure: *dual_update*)。

Algorithm 1 *MDCP_network_simplex*

Start with an initial basic flow x on a basic graph G_B ;
Calculate the dual basic solutions on G_B by procedure *compute_dual_variables*;
While (some nonbasic arc violates the optimality condition) **do**
 Select an entering arc (k, l) violating the optimality condition;
 Conduct flow-pivoting operations and determine the leaving arc (v, z) by
 procedure *flow_update*;
 Update dual variables by procedure *dual_update*;
End while;

4.1 對偶變數計算程序

雖然 Fang and Qi(2003) 提出一套計算對偶變數的演算法，但是該方法仍偏向直接求解聯立方程式，缺乏對偶變數之圖形特性相關探討。在傳統最小成本流量問題—MCF 中，由於每個節點 $i \in N$ 均有其獨自的對偶變數 π_i ，而且基解圖為一個展開樹，所以只要令網路中某一個節點的對偶變數為常數值（通常令展開樹根節點的對偶變數為 0），再依照互補差額定理— $(i, j) \in B$ 必須滿足 $\pi_i - \pi_j = c_{ij}$ 之特性，即可持續推斷出所有的對偶變數值。但是對於 MDCP 而言，除了節點 $i \in N$ 有對偶變數 π_i 外，每個 D 節點 i 尚有 q_i 個對偶變數 $\rho_{ij} : j \in L(i)$ ，並遵守 $\pi_i = \sum_{j \in L(i)} k_{ij} \rho_{ij}$ ，而使得對偶變數的推算更顯困難。如圖 4.1 所示，節點 i 為 \hat{O} 節點，而 k 與 j 皆為 D 節點；假如我們僅知道 π_i 的對偶變數值，僅能根據 $\rho_{ki} - \pi_i = c_{ki}$ 與 $\pi_i - \pi_j = c_{ij}$ 從而推論出 ρ_{ki} 與 π_j 之值，但是其他與節點 k 、節點 j 相關的對偶變數值則無法立即推得。亦即每當嘗試從 MDCP 中的某一個節點推算其他節點之對偶變數時，有可能因為遭遇到 D 節點因無法繼續往下推算。故本研究將基解圖適當地分群，使得同一群集內的所有對偶變數皆可由群集中的任意一個節點之對偶變數推導而得，並以此做為對偶變數計算之基礎。

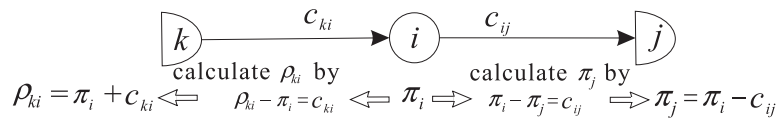


圖 4.1: 對偶變數之推算受限於 D 節點

4.1.1 基群集

由3.3節可知，每一個D節點 i 的對偶變數 π_i 未直接與其流出弧 $(i, j) : j \in L(i)$ 的對偶可行解條件相關，而是透過 $\pi_i = \sum_{j \in L(i)} k_{ij} \rho_{ij}$ 之方式間接影響；另一方面，為了能以圖形表示對偶變數之間的關聯，使得一個節點仍只對應到一個對偶變數，本研究針對每條流出弧 $(i, j) : j \in L(i)$ 額外定義一個方框點 i_j 與其相接，使得每個原始的節點 $i \in N$ 皆對應到對偶變數 π_i ，而新增的方框點 $i_j : i \in N_D, j \in L(i)$ 對應到對偶變數 ρ_{ij} ，如圖4.2所示。因此，一個「完整」的D節點 i 乃由一個節點 i (對應 π_i)與 q_i 個方框點 i_j (對應 ρ_{ij})所構成。而節點分割後在邏輯上仍以 $(i, j) : j \in L(i)$ 來表示D節點 i 的流出弧，亦即往後討論至D節點的節點度數(node-degree)或是流出弧個數時，需考慮這 $q_i + 1$ 個點，方能代表一個「完整」的D節點；之後利用D節點的分割即可將 $n + q$ 個對偶變數做適當分群。

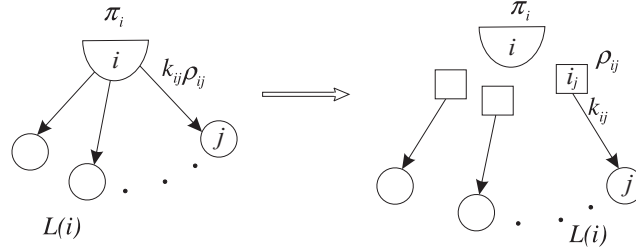


圖 4.2: 新增 i_j 以對應 ρ_{ij}

圖4.3(a)為一組MDCP的基解圖，弧上之數字代表其單位流量成本值；圖4.3(b)為拆解D節點後所分割之群集。在圖4.3(b)中，每個節點 $i \in N$ 均可對應到對偶變數 π_i ，而每個方框點 $i_j : i \in N_D, j \in L(i)$ 則可對應到對偶變數 ρ_{ij} ，其中 BC_i 代表第 i 個分群，假設節點1的對偶變數 $\pi_1 = t_1$ ，表示 π_1 為一暫時給定的變數值，之後沿基弧可推算節點2、節點4、節點7與節點9所對應的 π_2 、 π_4 、 π_7 與 π_9 ，使之能以 t_1 表示；而當推算至 ρ_{47} 與 ρ_{59} 時，因為遭遇D節點導致無法繼續推算其他的對偶變數，這些可推得之點與其連接弧即形成 BC_1 。引用同樣的概念，若令節點6的對偶變數值 $\pi_6 = t_2$ ，可推得 BC_2 及其所包含的對偶變數；令 $\pi_3 = t_3$ 則可得出 BC_3 及其所包含的對偶變數，此圖即包含三個基群集。

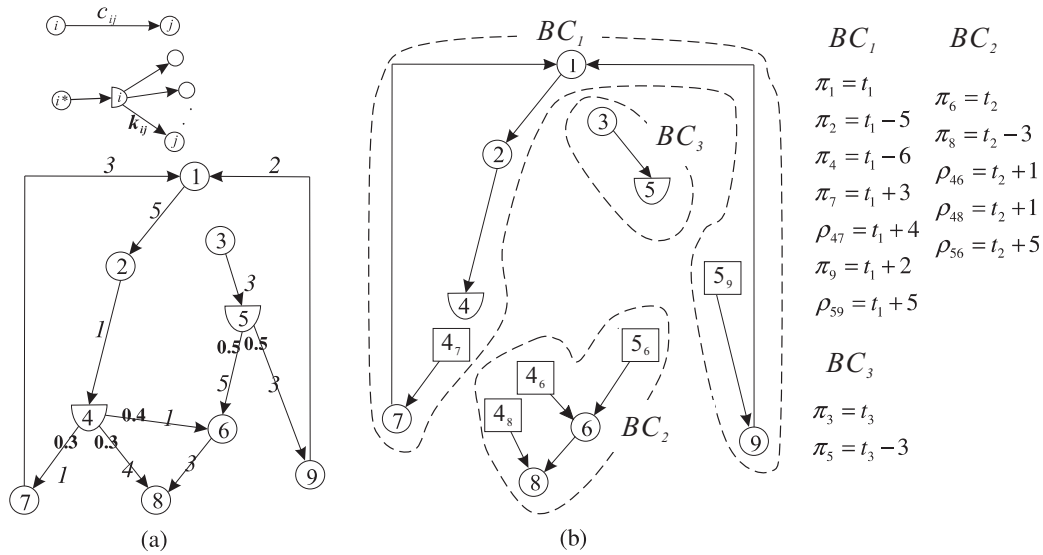


圖 4.3: 基群集 (BC)

對基解圖採取分群的主要用意在於找出各個能被同一個符號標示的對偶變數，使得 BC_i 中的對偶變數值皆能以變數 t_i 表示，而基群集的種類主要有兩種：含有 \hat{O} 節點的群集與不含 \hat{O} 節點（僅包含 D 節點或方框點）之群集。針對含有 \hat{O} 節點之群集，我們可任意選擇某 \hat{O} 節點為群集的起始節點並沿基弧搜尋該群集範圍，若遭遇到無法進一步推算的 D 節點即返回，繼續搜尋直至所有可連通的 \hat{O} 節點均被搜尋到為止；如果基解圖中的某 D 節點 i 僅與 q_i 條弧相連，該 D 節點所代表的 $q_i + 1$ 個節點中就會有一節點未與基弧相接而自成群集。如圖 4.4(a)，因為節點 i 之流入弧 (i^*, i) 不為基弧，使得節點 i 不與任何弧相接而自成群集；圖 4.4(b) 則因弧 (i, j) 不為基弧，造成 i_j 不與任何弧相接而自成群集。

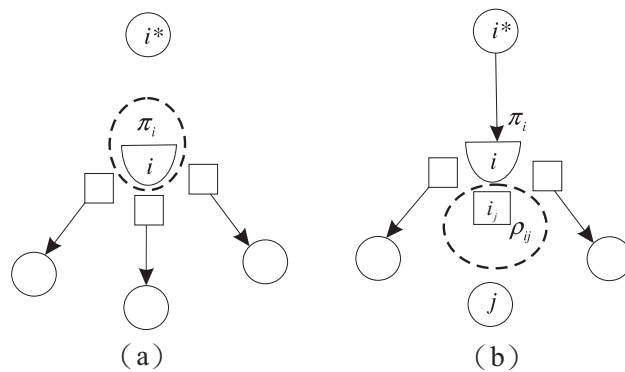


圖 4.4: 若 D 節點有一條相鄰弧不為基底，存在自成群集之節點（虛線圈選處）

Lemma 2. 若 x 為精簡後 (即任兩 D 節點不相鄰) 之 MDCP 所對應的可行基解，而 G_B 為 x 所代表的圖形，其中 $|N| = n$ ， $|N_D| = p$ ，D 節點 i 的流出弧個數 $|L(i)| = q_i$ ， $\sum_{i \in N_D} q_i = q$ 。則 G_B 在拆解 D 節點，進行基群集的分群後，可得基群集之特性如下：

- (i) 基群集數目恆為 $p + 1$ 。
- (ii) 群集內節點與弧為一樹狀結構 (tree)。
- (iii) 包含 \hat{O} 節點之群集數目最多為 $\min\{p + 1, n - p\}$ ；亦即相連弧皆為基底之 D 節點個數最多為 $\min\{p, n - p - 1\}$ 。

Proof:

- (i) 由 Lemma 1 可知，每個 D 節點適當移除 $q_i - 1$ 條弧後可形成一展開樹，而且移除之後，在 G_B 中的每個 D 節點的度數必為 1 或 2；如果再針對此展開樹的每個 D 節點各移除一條弧，共移除 p 條弧後，則可將圖形拆解為 $p + 1$ 個子群集 (亦即在原 G_B 中，若每個 D 節點移除 q_i 條弧後，可分群為 $p + 1$ 群)。另外，由於 MDCP 的迴圈乃由 D 節點所引起，所以每一個群集內的 \hat{O} 節點仍為一樹，且分屬於此 $p + 1$ 群中，亦為基群集之基礎；之後即可將分割後的 D 節點適當歸類於此 $p + 1$ 群之中，分述如下：

1. 若某 D 節點 i 與 $q_i + 1$ 條弧相連：此種情形因為 D 節點 i 透過弧 (i^*, i) 與節點 i^* 相接，故 D 節點 i 所屬的群集即為節點 i^* 所屬之群集；而方框點 $i_j : j \in L(i)$ 因為透過弧 (i, j) 與節點 j 相接，使得各個 i_j 分屬節點 $j \in L(i)$ 所屬之群集。
2. 若某 D 節點 i 僅與 q_i 條弧相連：此種情形在上述對每個 D 節點刪除 q_i 條弧並使之分為 $p + 1$ 群後，該節點則存在某一對偶變數自成群集。如果原本 (i^*, i) 不為基弧 (非基變數)，則節點 i 會因而自成群集，而其他節點 $i_j : j \in L(i)$ 則與節點 j 所屬同樣之分群；若原本某條流出弧 (i, k) 不為基弧，則節點 i 仍在節點 i^* 所屬之群集，方框點 $i_j : j \in L(i), j \neq k$ 分屬於節點 j 所屬之群集，而方框點 i_k 會因而自成群集。

綜合以上，則可將所有的對偶變數歸類於 $p+1$ 個子群集中。

- (ii) 由 Lemma 1，由於 MDCP 之迴圈必包含 D 節點，亦即迴圈由 D 節點所引起；故在拆解 D 節點後，單獨看各群集內的節點與弧為一樹狀結構，不包含迴圈。
- (iii) 假如 $n - p > p$ ，由 (ii) 知基群集個數為 $p + 1$ ，故內含 \hat{O} 節點之群集個數最多為 $p + 1$ 且交集於 p 個 D 節點；假如 $n - p \leq p$ ，因為一個 \hat{O} 節點只能存在於一個群集，故此 $n - p$ 個 \hat{O} 節點最多分屬於 $n - p$ 個分群，且交集於 $n - p - 1$ 個 D 節點。因此包含 \hat{O} 節點之群集個數最多為 $\min\{p + 1, n - p\}$ ；而相連弧皆為基底之 D 節點個數最多為 $\min\{p, n - p - 1\}$ 。

□

基解圖在分割 D 節點後，以方框點 $i_j : j \in L(i)$ 對應 D 節點 i 的對偶變數 ρ_{ij} ，則可提出尋找基群集之程序，使基解圖 G_B 拆解為 $p + 1$ 個群集 ($G_B = \bigcup_{i=1}^{p+1} BC_i$) 的步驟如下所示：

步驟 0: 初始化群集標號 $k = 1$ ，且 $BC_i = \phi : i = 1, \dots, p + 1$ 。

步驟 1: 令某一未被分群的 \hat{O} 節點 r 之對偶變數 $\pi_r = t_k$ 為群集 k 的起始點，利用 DFS (Depth-First Search Algorithm) 或 BFS (Breadth-First Search Algorithm) 以依據 $\pi_p - \pi_l = c_{pl} : \forall (p, l) \in B$ 推算其他 \hat{O} 節點之對偶變數並拓展群集，若遭遇 D 節點即返回，繼續搜尋直至所有可連通的 \hat{O} 節點全部被搜尋到為止。

步驟 2: 令 $k = k + 1$ 。若存在某 \hat{O} 節點仍未被分群，回步驟 1；否則繼續下一步驟。

步驟 3: 對每一個 D 節點 i 的流入弧 (i^*, i) 做檢查，倘若 (i^*, i) 為基弧，則將節點 i 與弧 (i^*, i) 放入節點 i^* 所屬分群，且 $\pi_i = \pi_{i^*} - c_{i^*i}$ ；若 (i^*, i) 不為基弧，則令 $\pi_i = t_k$ 且節點 i 自成群集為 BC_k ，且 $k = k + 1$ 。

步驟4: 對每一個D節點 i 的所有流出弧 (i, j) 做檢查，倘若弧 (i, j) 為基弧，則 $\rho_{ij} = c_{ij} + \pi_j$ ，且將方框點 i_j 與弧 (i, j) 放入節點 j 所屬群集；如果 (i, j) 不為基弧，令 $\rho_{ij} = t_k$ ，而方框點 i_j 自成群集為 BC_k ，且 $k = k + 1$ 。當所有D節點均被檢驗後，此時基群集即分類完畢。

上述演算法中，步驟1與步驟2主要搜尋出所有包含 \hat{O} 節點所在之群集，由於演算法乃根據所有基弧持續拓展群集，故步驟1與步驟2花費 $\theta(n + q)$ 。步驟3針對每一個D節點之流入弧做檢查，耗時 $\theta(p)$ ；而步驟4則是對D節點所有流出弧做檢查，耗時 $\theta(q)$ ；演算法整體的時間複雜度為 $\theta(n + q)$ ，虛擬碼如下所示。

Procedure *BC_classification*;

$k := 1$; $BC_i := \phi : \forall i = 1, \dots, p + 1$;

While (existing some node $i \in N_{\hat{O}}$ not classified into some basic component) **do**

Let $\pi_r = t_k$ for some unvisited node $r \in N_{\hat{O}}$, then add node r into BC_k ;

Starting from node r , add the arcs and the \hat{O} -nodes visited by DFS or

BFS into BC_k , and calculate dual variables by $\pi_p - \pi_l = c_{pl} : \forall (p, l) \in B$;

$k++$;

While end;

For (each incoming arc (i^*, i) to node $i \in N_D$) **do**

If (arc (i^*, i) is a basic arc) **then**

$\pi_i = \pi_{i^*} - c_{i^*i}$;

Add node i and arc (i^*, i) into the same component of node i^* ;

Else if (arc (i^*, i) is a non-basic arc) **then**

$\pi_i = t_k$ and add node i into BC_k ;

$k++$;

For (each outgoing arc $(i, j) : j \in L(i)$ from node $i \in N_D$) **do**

If (arc (i, j) is a basic arc) **then**

$\rho_{ij} = \pi_j + c_{ij}$;

Add node i_j and arc (i, j) into the same component of node j ;

Else if (arc (i, j) is a non-basic arc) **then**

$\rho_{ij} = t_k$ and add node i_j into BC_k ;

$k++$;

4.1.2 對偶變數求算方法

透過基群集之分類，可將原本 $n + q$ 個對偶變數(n 個 $\pi_i : i \in N_D$ 與 q 個 $\rho_{ij} : i \in N_D, j \in L(i)$)適當歸類為 $p + 1$ 群，其中D節點為群集之間的「聯結點」，

而這些節點的對偶變數相依方程式 $\pi_i = \sum_{j \in L(i)} k_{ij} \rho_{ij} : \forall i \in N_D$ 即為聯繫這些群集之關鍵，亦為所需求解的方程式。如同MCF般，因為 n 條流量守恒限制式包含一條多餘限制式，故可先令某一群集 r 的變數 $t_r = 0$ ；之後先利用各個D節點的對偶變數相依方程式求解出先前所設立的變數，再將之代回即可求解出所有的對偶變數值。如圖 4.3(b)，先令 $t_1 = 0$ ，再求解節點4與節點5的相依方程式 — $t_1 - 6 = 0.3(t_1 + 4) + 0.3(t_2 + 1) + 0.4(t_2 + 1)$ 與 $t_3 - 3 = 0.5(t_1 + 5) + 0.5(t_2 + 5)$ ，得出 $t_2 = -11.286$ 與 $t_3 = 2.357$ ，並將之代回全部的對偶變數，即可求解出全部的對偶變數，如圖 4.5所示。

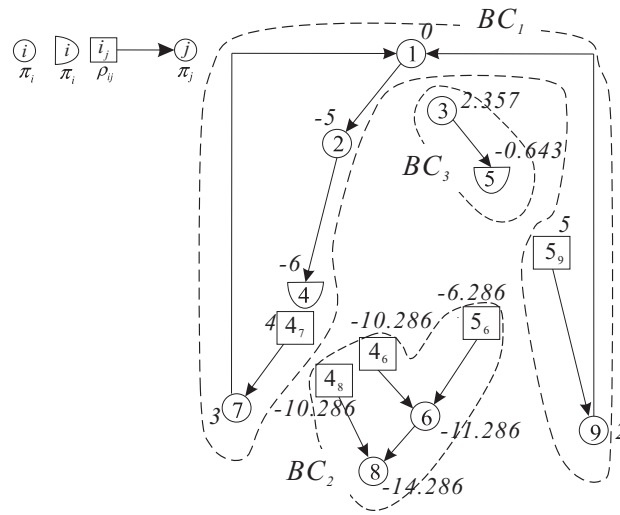


圖 4.5: 圖 4.3求解後之對偶變數值

然而，如果某些群集只與一個D節點相關，則可把此類D節點的方程式留至最後才處理，以減少最後求解聯立方程組的大小。例如圖 4.4(a) 中的 π_i 與圖 4.4(b) 中的 ρ_{ij} 等各自成群的對偶變數，或是 4.3(b) 中的 BC_3 ，由於其內的對偶變數只出現在該D節點的相依方程式，因此可以留至最後才處理。假設此類型D節點所構成的集合為 N_{D1} ，我們可先求解由 $\pi_i = \sum_{j \in L(i)} k_{ij} \rho_{ij} : i \in N_D \setminus N_{D1}$ 所構成的聯立方程組，等到這些群集變數求解完畢後，再針對其它D對點的相依方程式進行求解，以得出其他的變數值。依據此準則可提出計算對偶變數程序的演算法如下：

步驟0: 初始化標號集合 $H = \phi$ ，節點集合 $N_{D1} = \phi$ 。

步驟1: 利用 procedure: *BC_classification* 得出基群集。

步驟2: 檢查是否存在只與單一 D 節點相關之群集，將此類 D 節點放入 N_{D1} ，並將該群集的標號放入 H 。

步驟3: 令某群組變數 $t_r = 0 : r \notin H$ ，求解由 $\pi_i = \sum_{j \in L(i)} k_{ij} \rho_{ij} : \forall i \in N_D \setminus N_{D1}$ 所構成之聯立方程組得出 $t_k : k \notin H$ 。

步驟4: 針對所有節點 $i \in N_{D1}$ ，利用 $\pi_i = \sum_{j \in L(i)} k_{ij} \rho_{ij}$ 求解尚未被解出之變數 $t_k : k \in H$ 。

步驟5: 將所有 $t_i : i = 1, \dots, p+1$ 值代回求解出全部對偶變數。

步驟1中尋找基群集的時間複雜度為 $\theta(n+q)$ ，步驟2可先統計每D節點與群集的相連情形，再挑出符合此類的群集與D節點，可在 $\theta(p+q)$ 時間完成。另外，由 Lemma 2可知欲求解之方程式總數最多為 $\min\{p, n-p-1\}$ ；因求解大小為 k 的聯立方程組耗時 $O(k^3)$ ，故步驟3共耗時 $O(\min\{p^3, (n-p)^3\})$ 。另外步驟4乃逐一求解尚未解出的對偶變數，每求解一節點耗時 $O(q_i+1)$ ，最多求解 p 個D節點，因此時間複雜度為 $O(p+q)$ ；步驟5則是將全部 $t_i : i = 1, \dots, p+1$ 代回求解 $n+q$ 個對偶變數，耗時 $O(p+q)$ 。綜合以上之程序，可在 $O(\min\{p^3, (n-p)^3\} + n+q)$ 計算出對偶變數，其虛擬碼如下所示。

procedure *compute_dual_variables*;

$H = \phi; N_{D1} = \phi;$

Use procedure: *BC_classification* to get basic components;

Identify those basic components associated with only one D-node, add their indices into H , and add the D-nodes into N_{D1} ;

Set $t_r = 0$ for some r not in H ;

Solve the system of equations made up of $\pi_i = \sum_{j \in L(i)} k_{ij} \rho_{ij} : \forall i \in N_D \setminus N_{D1}$;

Solve the remaining $t_k : k \in H$ by $\pi_i = \sum_{j \in L(i)} k_{ij} \rho_{ij} : i \in N_{D1}$;

Calculate all dual variables from the solved $t_i : i = 1, \dots, p+1$;

4.2 流量疊代—基變數的進入與退出

以往單形法在求解線性規劃問題時，即是藉由基變數的進入與退出進行反覆的疊代過程以達成目標值的改進；網路單形法則是將之改以圖形化的方式實行。在MCF中，由於基解所對應的圖形為一展開樹，使得當某一變數 x_{kl} 為進入變數(進入弧)時，得以在樞紐圖(pivoting graph)中形成獨特之迴圈，並可沿此迴圈推送流量，選擇最先到達上界或是下界之弧為退出弧；另外，也由於MCF網路架構的簡單性，使得迴圈中之弧與進入弧的流量改變量呈等比例增加或減少。但在MDCP中，因為D節點的分流限制式之影響，不但使其基解圖不為一展開樹，亦使得樞紐圖中的流量改變量呈現更複雜的關係。圖4.6(a)為一個基解圖，其中弧(2,6)為進入變數；而圖4.6(b)代表弧(2,6)進入且推送流量 Δ 後各弧的流量改變量，未繪出之部分則代表其流量值不變；由此範例中亦可看出網路單形法在執行基變數的進入與退出的選擇時將遭遇的一項重要問題：如何計算樞紐圖中的流量比例關係。

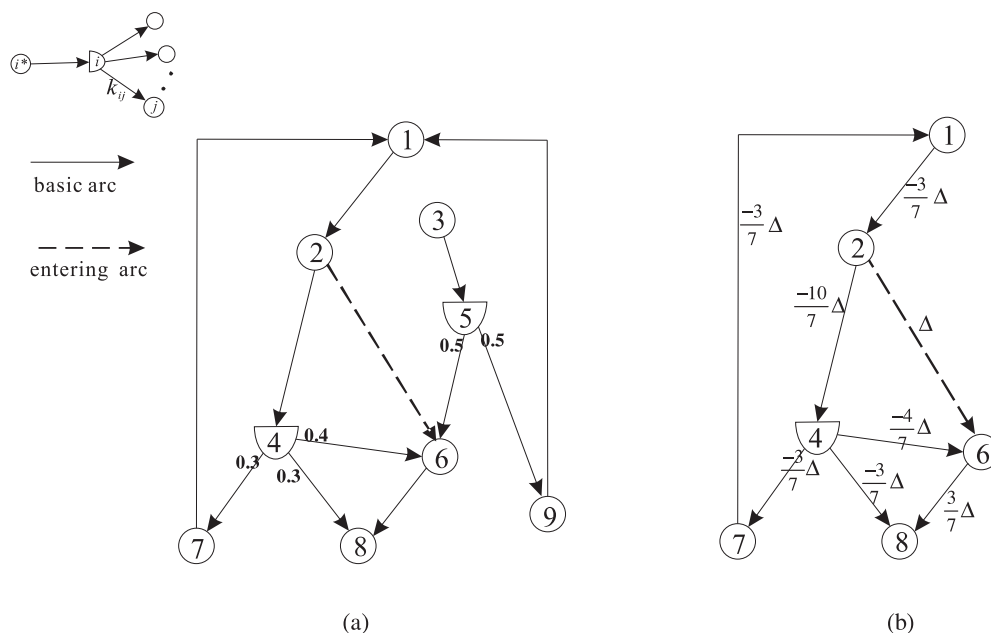


圖 4.6: MDCP 樞紐圖的流量比例關係複雜

Ting(2005) 求解最大流量問題時利用 compatible component 進行圖形的分群以進行流量比例的計算，並指出必須適當的分群以及設置變數才能進行流量比例

個可推送流量之迴圈，且其流量比例關係簡單。之後只要沿此迴圈等比例推送流量，同樣選擇最先到達上界或下界之弧為退出弧，即可完成一次 iteration(若 $(7,9)$ 為流量達上界之進入弧，則反向推送流量)。

依據現行常用的圖形搜尋法不難找出群集內由 \hat{O} 節點構成的迴圈，例如依據流入弧與流出弧使用 DFS 或 BFS 等，最多費時 $O(n+q-p)$ (代表全部之弧均集中在此群集內)即可找出此迴圈，其他諸如流量更新等判斷方式亦與 MCF 相同，在此不加以贅述，其時間複雜度為 $O(n+q-p)$ ，虛擬碼則如下所示。

procedure *flow_update_same_BC*;

Add the entering arc (k,l) into the BC_i of node k and node l ;

Identify the unique cycle in BC_i ;

If ($x_{kl} = 0$) **then**

Send flows along the orientation of (k,l) in the cycle;

Else then

Send flows along the opposite orientation of (k,l) in the cycle;

Determine the leaving arc (v,z) which first achieves its upper bound or lower bound, then update flows;

4.2.2 進入弧之兩端點位於不同群集

當進入弧連接的兩端點發生在兩不同群集時，由於聯繫此兩群集之媒介為 D 節點，因此進入弧若有流量產生時，勢必會使其流通至聯繫兩群集的 D 節點(可能一個或多個)並拓展至其他群集。其中進入弧 (k,l) 可依照兩端點的節點類型區分為三類：

1. 若節點 k 與節點 l 皆為 \hat{O} 節點時，進入弧將會連結節點 k (對應 π_k) 與節點(對應 π_l) 所在的兩群集，使之合併為一群集。
2. 若節點 k 為 \hat{O} 節點而節點 l 為 D 節點時，則進入弧將會連結節點 k (對應 π_k) 所在之群集與原本獨自成群的節點 l (對應 π_l)，使之合併為一群集。
3. 若節點 k 為 D 節點而節點 l 為 \hat{O} 節點時，則進入弧將會連結獨自成群的節點 k_l (對應 ρ_{kl}) 與節點 l (對應 π_l) 所在之群集，使之合併為一群集。

而不論是上述何者情形，進入弧加入基底之後均會使得基群集從 $p + 1$ 群變為 p 群，且群集內之 D 節點仍呈樹狀結構。

在知道進入弧加入圖形所造成群集數目的改變後，可對樞紐圖中確定無法通過流量之分支進行刪除，以進一步簡化基群集的架構並降低計算流量的困難度。如同 MCF 般，當樞紐圖中的 \hat{O} 節點其節點度數為 1 時，顯然該節點與其連接弧可通過之流量為 0，故可先給予刪除；另外，如果某 D 節點 i 在樞紐圖中只有 q_i 條相連弧，因為流入弧的流量必須與流出弧的流量維持比例關係，使得此類的節點與其連接弧亦無法通過流量而可刪除。經由反覆執行這兩種策略即可先將樞紐圖做預先處理，直至所有 \hat{O} 節點之相連度數均至少為 2，而每個 D 節點 i 均與 $q_i + 1$ 條弧相連為止。以圖 4.6(a) 為例，因為節點 3 之節點度數為 1，故可刪除節點 3 與弧 (3, 5)，而後因為節點 5 的度數變為 $2(q_5 + 1 = 3)$ ，可再刪除該節點與相連弧，反覆檢查可刪除之分支之後，又可使得弧 (5, 6)、(5, 9) 與 (9, 1) 等弧亦被刪除。因此，藉由上述策略的執行，可提出刪除分支的虛擬碼如下：

```
procedure delete_branch;
```

```
  While (existing some  $\hat{O}$ -node  $i$  with node-degree=1 or some  $D$ -node  $i$ 
    with node-degree= $q_i$ ) do
```

```
    Remove node  $i$  and all its connecting arcs;
```

```
  While end;
```

刪除流量分支之程序可藉由先挑選符合刪除條件的節點，並放入佇列 (queue) 中，之後從佇列挑選出節點，再進而沿其相連弧繼續下一次的刪除過程，最多刪除基解圖中所有節點與弧 (但是不可能發生)，故可知其時間複雜度為 $O(n + q)$ 。

如令 $\tilde{G} = (\tilde{N}, \tilde{A})$ 表示樞紐圖在刪除分支後所構成的子圖 (\tilde{N} 為節點集合， \tilde{A} 為弧集合)，可再對 \tilde{G} 之特性做進一步的簡化，因為第一項策略刪除節點度數為 1 的 \hat{O} 節點，使得原本群集中樹之葉端 (leaf) 必為 D 節點 (節點 $i \in N_D$ 或是方框點 $i_j : i \in N_D, j \in L(i)$)；第二項策略刪除僅與 q_i 條弧相連之 D 節點 i ，該策略除了刪除部分基群集內的分支外，亦會使得只與某一 D 節點相關的群集被刪除 (包含自成群集的 $\pi_i : i \in N_D$ 、 $\rho_{ij} : i \in N_D, j \in L(i)$ ，或是如圖 4.3(b) 中的 BC_3)，使得所需

考慮推送流量的基群集從 p 群縮減至 \tilde{p} 群，亦代表樞紐圖中只包含 \tilde{p} 個D節點，而 \tilde{G} 的特性演變為：

1. 群集內的節點仍為一樹狀結構，但樹之葉端必為D節點(D節點 i 或是 i_j)。
2. 樞紐圖中的所有D節點必與 $q_i + 1$ 條弧相連。

因此，在得知刪除分支對群集特性的改變之後，就可開始運用這些特性進行樞紐圖中的流量比例的推算，且由Lemma 2可知，基解圖中包含 \hat{O} 節點之群集最多有 $\min\{p+1, n-p\}$ 個，進入弧加入圖形之後將會合併兩群集，使得刪除分支後的群集個數最多為 $\tilde{p} \leq \min\{p, n-p\}$ 。

本研究採取類似Ting(2005)求解流量變化的方法，亦對某些弧設立變數代表其變化量，再進而求解某些節點的流量守恒限制式以得到樞紐圖中的流量比例關係；與Ting(2005)不同之處在於本研究乃針對每一個D節點 i 的流入弧 (i^*, i) 設置變數。如圖4.8所示，若對弧 $(2, 3)$ 設置其流量改變量為 a ，則其流出弧 $(3, 4)$ 、弧

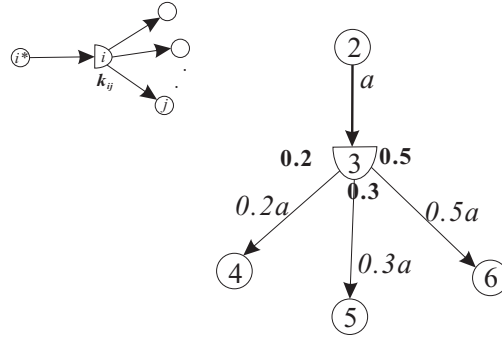


圖 4.8: 針對D節點的流入弧設置變數，並推算其流出弧流量改變量

$(3, 5)$ 與弧 $(3, 6)$ 可由D節點的分流比例限制式得出其流量改變量各為 $0.2a$ 、 $0.3a$ 與 $0.5a$ 。因此，如果群集中有 \tilde{p} 個D節點，則共設置 \tilde{p} 個變數先標示出所有與D節點相連之弧的改變量。

由於刪除分支後各群集內之節點仍成一樹狀結構，且葉端必為D節點(D節點 i 或是方框點 i_j)，又與這些D節點相連之弧皆可給予一變數表示，故此時就可由之前所標記之變數，以群集為單位，由樹的葉端持續往根節點推算群

集中每一條弧的流量改變量，而樹的根節點就為求解整體樞紐圖流量比例關係的「關鍵節點」(key-node)。如圖 4.9，此群集中由於葉端皆為 D 節點，而且其相連弧已給予變數表示；因此可由樹之葉端逐一往根節點推算群集中各弧的流量改變量，直至遭遇至樹的根節點 r 為止，而節點 r 的流量守恒限制式： $-0.5a - 0.5b - 0.3c = 0.4d - e$ 即為計算流量比例所需之方程式。

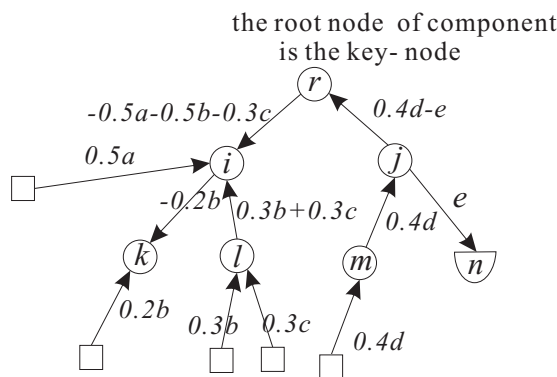
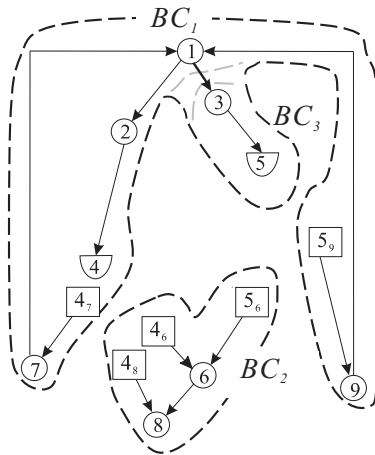
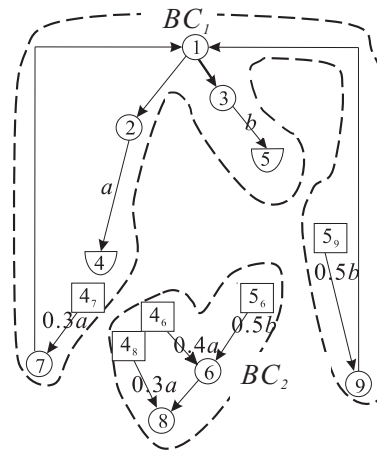


圖 4.9: 從樹之葉端往根節點推算流量

為了統一表示起見，令 f_{ij} 表示所有弧 $(i, j) \in \tilde{A}$ 的流量係數，亦即當進入弧 (k, l) 推送一單位流量時在 (i, j) 上的流量改變量。先對每條弧 $(i^*, i) : i^* \in N_{\tilde{O}}, i \in N_D$ 之流量係數 f_{i^*i} 設置變數 (a, b, \dots 等)，一共設置 \tilde{p} 個變數，再以前述之方法對 \tilde{p} 個基群集各自推算群集中的流量比例，即可標示出樞紐圖中所有的流量比例係數，且一共得到 \tilde{p} 個關鍵節點。另外，由於原本 n 個節點的流量守恒限制式即已包含一條多餘限制式，代表此 \tilde{p} 個關鍵節點的流量守恒限制式中，可捨棄一條多餘的限制式；因此求解由 $\tilde{p} - 1$ 條關鍵節點之方程式與加入 $f_{kl} = 1$ 或 $f_{kl} = -1$ 所構成之聯立方程組後 (f_{kl} 表示進入弧 (k, l) 的流量比例，若 $x_{kl} = 0$ 則求解 $f_{kl} = 1$ ，表示沿 (k, l) 正向推送流量；若 $x_{kl} = u_{ij}$ 則求解 $f_{kl} = -1$ ，表示反向推送流量)，再將之代回各弧即可得到樞紐圖中各弧流量改變量的比例關係。圖 4.10 即為一計算流量比例的範例，弧 $(1, 3)$ 為原先流量達下界的進入弧；首先，加入弧 $(1, 3)$ 後將會連接兩個群集 BC_1 與 BC_3 。由於此圖不存在無法通過流量之分支，所以先對 D 節點的流入弧設置變數，並依照上述的推論方法標示樞紐圖中的流量比例，再加上 $f_{13} = 1$ 進行求解後，即可得出樞紐圖中的流量比例值。

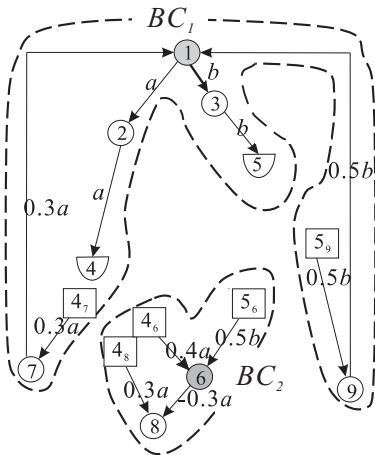


$$\begin{aligned} k_{46} &= 0.4 \\ k_{47} &= 0.3 \\ k_{48} &= 0.3 \\ k_{56} &= 0.5 \\ k_{59} &= 0.5 \end{aligned}$$



(a) Add the entering arc (1,3) .

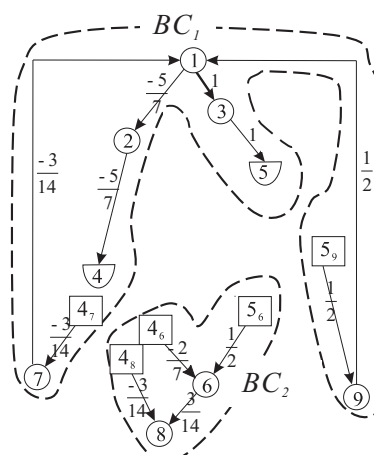
(b) Set variables to arc (2,4) and arc (3,5).



solve

$$\begin{aligned} a + b - 0.3a - 0.5b &= 0 \\ \text{and } f_{13} &= 1 \quad (b=1) \end{aligned}$$

$$a = -5/7, \quad b = 1$$



(c) Calculate all f_{ij} by variables a and b (node 1 and 6 are key-nodes).

(d) Get all f_{ij} by solving the flow balance constraint of node 1 and $f_{13}=1$.

圖 4.10: 計算樞紐圖流量比例範例

得出樞紐圖中的流量比例之後，即可依此比例推送流量，並利用已求解的流量比例對樞紐圖中所有弧 $(i, j) \in \tilde{A}$ 做最小值檢定以選擇退出弧，如果 $f_{ij} < 0$ ，表示當樞紐圖推送流量時，弧 (i, j) 之流量值往下界移動，可得 $e_{ij} = -x_{ij}/f_{ij}$ ；如果 $f_{ij} > 0$ ，表示當樞紐圖推送流量時，弧 (i, j) 之流量值往上界移動，可得 $e_{ij} = (u_{ij} - x_{ij})/f_{ij}$ ；如果 $f_{ij} = 0$ ，代表不可推送流量，可令其為 M （因為我們無法保證所有不可通過流量之弧均可被刪除，故仍須考慮此情形）；而後即可選擇 $e_{vz} = \min\{e_{ij} : (i, j) \in \tilde{A}\}$ 之弧 (v, z) 為退出弧並更新流量，即完成流量之疊代。綜合以上刪除分支、計算流量比例與最小值檢定，可提出流量疊代之程序如下：

步驟1: 加入進入弧 (k, l) 形成樞紐圖，使用 procedure: *delete_branch* 刪除分支得到 \tilde{G} ，且令 f_{ij} 為圖中每條弧 $(i, j) \in \tilde{A}$ 的流量比例係數。

步驟2: 對 \tilde{G} 中每一個 D 節點 i ，選擇其流入弧 (i^*, i) 之流量係數 f_{i^*i} 設置變數，並往下推算其所有流出弧係數 $f_{ij} : j \in L(i)$ 。

步驟3: 對於 \tilde{G} 中的每一個群集，任選其中一 \hat{O} 節點為根節點（關鍵節點），並由群集內樹之葉端往根節點推算各 $f_{ij} : (i, j) \in \tilde{A}$ ，且得到 \tilde{p} 個關鍵節點的流量守恒限制式。

步驟4: 若 $x_{kl} = 0$ ，求解 $\tilde{p} - 1$ 條關鍵節點的流量守恒限制式與 $f_{kl} = 1$ 所構成之聯立方程組；否則（即 $x_{kl} = u_{kl}$ ）求解 $\tilde{p} - 1$ 條關鍵節點流量守恒限制式與 $f_{kl} = -1$ 所構成之聯立方程組。之後將解代回計算所有 $f_{ij} : (i, j) \in \tilde{A}$ 。

步驟5: 對所有弧 $(i, j) \in \tilde{A}$ 計算 e_{ij} ，若 $f_{ij} < 0$ ，令 $e_{ij} = -x_{ij}/f_{ij}$ ；若 $f_{ij} > 0$ ，令 $e_{ij} = (u_{ij} - x_{ij})/f_{ij}$ ；若 $f_{ij} = 0$ ，令 $e_{ij} = M$ 。並選擇弧 $(v, z) = \arg \min\{e_{ij} : (i, j) \in \tilde{A}\}$ 為退出弧。

步驟6: 對 $e_{ij} \neq M : (i, j) \in \tilde{A}$ 之弧 (i, j) 更新流量值，使 $x_{ij} = x_{ij} + f_{ij}e_{vz}$ 。

步驟1 去除分支所花時間為 $O(n + q - p)$ ，步驟2 對所有與 D 節點的相連弧設置變數，耗時 $O(\tilde{q} + \tilde{p})$ （ \tilde{q} 為 \tilde{p} 個 D 節點流出弧總數）；步驟3 中根據群集內之樹，將所有弧之係數改以所設置之變數表示，由於樞紐圖包含 $n + q - p$ 條弧，而步驟

2 已標示出 $p + q$ 條弧條，因此步驟3最多標記 $n + q - p - (p + q) = n - 2p$ 條弧，每條弧最多以 \tilde{p} 個變數表示，故步驟3的時間複雜度為 $O(n\tilde{p})$ ；同樣因為求解大小為 k 的聯立方程組耗時 $O(k^3)$ ，故步驟4耗時 $O(\tilde{p}^3)$ ，步驟5與步驟6做最小值檢定並更新流量花費 $O(n + \tilde{q} - \tilde{p})$ ，又 $\tilde{p} \leq \min\{p, n - p\}$ ，因此整體演算法之複雜度為 $O(\min\{p^3 + np, (n - p)^3 + n(n - p)\} + q)$ ，其虛擬碼如下所示。

procedure *flow_update_diff_BC*;

Add the entering arc (k, l) into G_B ;

Apply procedure: *delete_branch* to get \tilde{G} ;

For (each node $i \in N_D$ in \tilde{G}) **do**

Assign a variable to f_{i^*i} where (i^*, i) is the incoming arc of node i ;

Calculate all $f_{ij} : j \in L(i)$ by $f_{ij} = k_{ij}f_{i^*i}$;

For (each basic component in \tilde{G}) **do**

Set some \hat{O} -node as the key-node, and calculate all f_{ij} in the component from leaves to the key node;

If ($x_{kl} = 0$) **then**

Solve the system of equations by $f_{kl} = 1$ and all the flow-balance constraints of the key-nodes except one to get all $f_{ij} : (i, j) \in \tilde{A}$;

Else then

Solve the system of equations by $f_{kl} = -1$ and all the flow-balance constraints of the key-nodes except one to get all $f_{ij} : (i, j) \in \tilde{A}$;

For (each arc $(i, j) \in \tilde{A}$) **do**

If ($f_{ij} < 0$) **then** $e_{ij} := -x_{ij}/f_{ij}$;

Else if ($f_{ij} > 0$) **then** $e_{ij} := (u_{ij} - x_{ij})/f_{ij}$;

Else then $e_{ij} := M$;

Select arc (v, z) as the leaving arc where $(v, z) := \arg \min\{e_{ij} : (i, j) \in \tilde{A}\}$;

For(each arc $(i, j) \in \tilde{A}$) **do**

Update flow: $x_{ij} := x_{ij} + f_{ij}e_{vz}$ for each $e_{ij} \neq M$;

4.3 對偶變數更新－更新對偶變數與基群集

傳統網路問題－MCF在流量疊代後並不需要重新計算新基解所對應的對偶變數，只要移除退出弧將圖形分解成為兩個子展開樹，以其中一個子展開樹為中心，再使進入弧加入圖形，更新另一展開樹內的對偶變數即完成對偶變數之更新。但在MDCP中，同樣因為D節點分流限制式所產生迴圈的影響，使得對偶變

數更新之方式更趨複雜；因此本研究同樣以基群集進行對偶變數更新與群集之更新。

在4.1.1節曾提到基群集恆為 $p+1$ 群，可利用標號1到 $p+1$ 替群集做區別，而如果退出弧移出基解圖後，則會使得退出弧所在之群集再分割為兩個子群集，使原本的 $p+1$ 群變為 $p+2$ 群，故以 BC_{p+2} 代表分割後新增的群集。亦即，如果退出弧所在群集為第 j 群，而在移除退出弧後，將其中一個群集仍保持標號為 j ，另一個則給予新的標號為 BC_{p+2} ，使得群集的總數變為 $p+2$ 。例如圖4.11中，假如 $(6,8)$ 為退出弧，在移除退出弧後即可將 BC_2 可分裂成 BC_2 與 BC_4 兩群集。

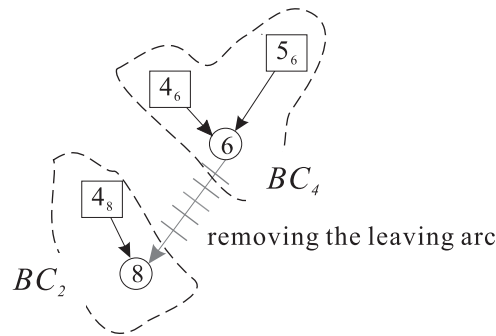


圖 4.11: 移除退出弧，群集分割為兩子群集

同一基群集內的對偶變數其相對關係必須保持不變，以便能維持基弧必須滿足 $\pi_i - \pi_j = c_{ij} : (i, j) \in B$ 或 $\rho_{ij} - \pi_j = c_{ij} : (i, j) \in B$ 之條件；因此如欲做對偶變數更新，也應以基群集為單位並利用此特性做有效率的更新。每個群集可令 δ_i 為 BC_i 內對偶變數的改變量，使得 BC_i 內的對偶變數滿足 $\pi_j^{new} = \pi_j^{old} + \delta_i$ 或是 $\rho_{jk}^{new} = \rho_{jk}^{old} + \delta_i$ (old與new分別表示更新前後的對偶變數值)。另外，進入弧 (k, l) 加入圖形後，會使得尾端所在的群集 BC_g 與頭端所在之群集 BC_h 再合併為一群集，並使得尾端為 \widehat{O} 節點的 (k, l) 需滿足：

$$\begin{aligned}
 \pi_k^{new} - \pi_l^{new} - c_{kl} &= 0 \\
 \Rightarrow \pi_k^{old} + \delta_g - \pi_l^{old} - \delta_h - c_{kl} &= 0 \\
 \Rightarrow \delta_h &= \delta_g - \pi_l^{old} + \pi_k^{old} - c_{kl}
 \end{aligned} \tag{4.1}$$

而尾端為D節點的 (k, l) 需滿足：

$$\begin{aligned}
& \rho_{kl}^{new} - \pi_l^{new} - c_{kl} = 0 \\
\Rightarrow & \rho_{kl}^{old} + \delta_g - \pi_l^{old} - \delta_h - c_{kl} = 0 \\
\Rightarrow & \delta_h = \delta_g - \pi_l^{old} + \rho_{kl}^{old} - c_{kl}
\end{aligned} \tag{4.2}$$

因此可利用(4.1)或是(4.2)先將 BC_g 與 BC_h 內的對偶變數改變量均給予同一變數表示，並使之合併為同一群集 BC_g ；之後只要先令其中一個群集的改變量為0，再求解由 $\pi_i = \sum_{j \in L(i)} \rho_{ij} : i \in N_D$ 構成之方程組，即可求解出各個 δ_i 值，並將 δ_i 代回更新各群組的對偶變數，且將原先新增之群集 BC_{p+2} 標號更改為 BC_h ，就完成對偶變數與基群集之更新。同4.1.2節般，可依照相同的準則對這 p 條方程式做求解次序之選擇，因此亦可提出更新的程序如下：

步驟0: 初始化 $N_{D1} = \phi$ ， $H = \phi$ 。

步驟1: 將退出弧 (v, z) 移除，使得第 j 個基群集分割為兩個子群 BC_j 與 BC_{p+2} 。

步驟2: 設 δ_i 為 $BC_i : i = 1, \dots, p+2$ 之對偶變數改變量。

步驟3: 使進入弧 (k, l) 進入圖形，利用(4.1)或(4.2)合併 BC_h 至 BC_g ，將合併後之對偶變數的改變量皆以 δ_g 表示之。

步驟4: 檢查是否存在只與單一D節點相關之群集，將此類D節點放入 N_{D1} ，並將該群集的標號放入 H 。

步驟5: 令某群組變數 $\delta_r = 0 : r \notin H$ ，由 $\pi_i = \sum_{j \in L(i)} k_{ij} \rho_{ij} : \forall i \in N_D \setminus N_{D1}$ 所構成之聯立方程組求解 $\delta_k : k \notin H$ 。

步驟6: 針對所有節點 $i \in N_{D1}$ ，利用 $\pi_i = \sum_{j \in L(i)} k_{ij} \rho_{ij}$ 求解剩餘尚未被解出之變數 $\delta_k : \forall k \in H$ 。

步驟7: 將所有 $\delta_i : i = 1, \dots, p+2$ 代回，依據 $\pi_j^{new} = \pi_j^{old} + \delta_i$ 或 $\rho_{jk}^{new} = \rho_{jk}^{old} + \delta_i$ 更新所有的對偶變數。

步驟8: 最後將 BC_{p+2} 更改為 BC_h 。

對偶更新之程序類似 procedure: *compute_dual_variables*，不同之處在於更新對偶變數的過程中，我們所設置的 δ_i 乃為群集 BC_i 的對偶變數整體變化量，不需要重新拓展群集以及計算群集內對偶變數值的相對關係。步驟1先根據退出弧拆解為 $p+2$ 群，待步驟3再使之合併為 $p+1$ 群，由於此程序求解變數的計算步驟與 procedure: *compute_dual_variables* 類似，故同樣耗時 $O(\min\{p^3, (n-p)^3\} + n + q)$ ；虛擬碼則如下所示。

procedure *dual_update*;

$N_{D1} = \phi; H = \phi;$

Remove the leaving arc (v, z) to split some BC_j into BC_j and BC_{p+2} ;

Let δ_i represent the change of dual variables for each dual variable in BC_i ;

Suppose k belongs to BC_g and l belongs to BC_h for the entering arc (k, l) ;

Calculate δ_h by δ_g using $\delta_h = \delta_g - \pi_l^{old} + \pi_k^{old} - c_{kl}$ or $\delta_h = \delta_g - \pi_l^{old} + \rho_{kl}^{old} - c_{kl}$;

Add the entering arc (k, l) to merge BC_h into BC_g

Identify those basic components associated with only one D-node, add their indices into H , and add the D-nodes into N_{D1} ;

Set $\delta_r = 0$ for some r not in H ;

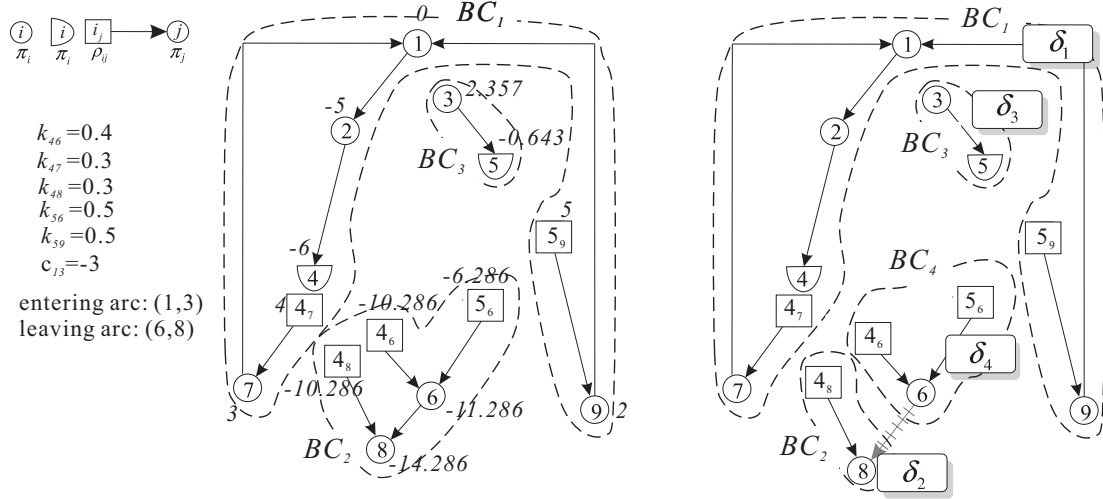
Solve the system of equations $\pi_i = \sum_{j \in L(i)} k_{ij} \rho_{ij} : \forall i \in N_D \setminus N_{D1}$;

Solve the remaining δ_i by $\pi_i = \sum_{j \in L(i)} k_{ij} \rho_{ij} : i \in N_{D1}$;

Update each dual variable by $\pi_j^{new} = \pi_j^{old} + \delta_i$ or $\rho_{jk}^{new} = \rho_{jk}^{old} + \delta_i$;

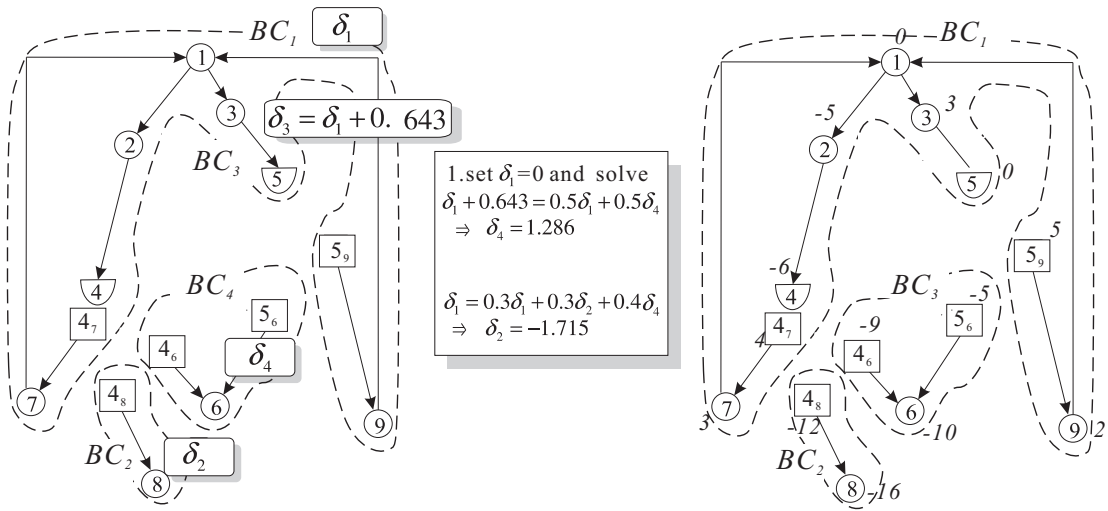
Change the index of BC_{p+2} to BC_h ;

圖 4.12 即為一次完整的對偶變數疊代過程，首先，在移除退出弧 (6, 8) 後，可將 BC_2 分割為兩群集 BC_2 與 BC_4 ，使得群集數變為四群，並令 δ_1 、 δ_2 、 δ_3 與 δ_4 為四個群集個別的對偶變數改變量；之後利用進入弧合併 BC_1 與 BC_3 ，並由 (4.1) 得出 $\delta_3 = \delta_1 + 0.643$ ；最後，令 $\delta_1 = 0$ 並求解方程式後即可得出各群集改變量，只要將 δ_i 代回更新對偶變數，並將 BC_4 標號更改為 BC_3 ，即完成此次對偶變數的更新程序。



(a) The basic graph before updating dual variables.

(b) Remove the leaving arc (6,8) and set variable $\delta_1, \delta_2, \delta_3$ and δ_4 .



(c) Merge BC_1 and BC_3 by the entering arc (1,3) and get $\delta_3 = \delta_1 + 0.643$.

(d) Update dual variables using δ_i .

圖 4.12: 對偶變數更新範例

4.4 初始解

初始解為單形法中重要的一個起步，一般來說，如果原問題的限制式均為「 \leq 」之形式，可對每條限制式加入適當的超額變數使之轉換成標準型式，並令這些超額變數為一開始的初始基解。但是對於問題限制式包含「 $=$ 」或「 \geq 」形態的限制式，除了「 \geq 」形式的限制式必須加入超額變數之外，尚需要對原本「 $=$ 」或是「 \geq 」型態之限制式額外加入人工變數 (artificial variable) 並令這些人工變數為初始基解，且在目標式中對人工變數給予一相當大的懲罰值 (最小化模式為 M ，最大化模式為 $-M$)。在傳統的 MCF 中，亦有其類似的應用程序，首先令一節點 r 為展開樹的根節點 (root node)，再對所有 $b_i > 0$ 相連之節點 i 新增弧 (i, r) ，並令 $c_{ir} = M$ ， $x_{ir} = b_i$ ；對於其他 $b_i \leq 0$ 之節點 i 則新增弧 (r, i) ，令 $c_{ri} = M$ ， $x_{ri} = -b_i$ ，另外這些新增之弧皆無容量上限，此法即為大 M 法的應用 (Big-M method)。

但在 MDCP 中，由於模式除流量守恒限制式外，尚有 D 節點的分流限制式，如果欲單純對每條限制式新增人工變數並令其為基解，很難在保持網路結構的條件下，對 D 節點的分流限制式適當地加入人工變數。所以本研究擬針對 \hat{O} 節點新增人工變數，而 D 節點則適當地選取連接弧使其滿足以下兩項基解圖擁有的特性：1. 弧個數為 $n + q - p - 1$ ；2. 若每個 D 節點 i 扣除 $q_i - 1$ 條弧後可形成一展開樹。因此，若令一開始 MDCP 中之流量皆為 0，可依循下列步驟設置初始解：

步驟 1: 選擇節點 s 為基解圖的基準節點；之後對於其他 \hat{O} 節點 $i \in N_{\hat{O}} \setminus \{s\}$ 新增弧

(s, i) ，並令其為基弧，其中 $x_{si} = -b_i$ 、 $u_{si} = M$ 且 $c_{si} = M$ 。

步驟 2: 對所有 D 節點 i ，選擇其所有流出弧 $(i, j) : j \in L(i)$ 為基底，並令 $x_{ij} = 0$ 。

步驟 1 與主要替所有的 \hat{O} 節點指派 $n - p - 1$ 條弧為基底，故時間複雜度為 $\theta(n - p)$ ；步驟 2 針對每一個 D 節點 i 選擇 q_i 條流出弧為基底，共指派 $\sum_{i \in N_D} q_i = q$ 條弧，因此步驟 2 之時間複雜度為 $\theta(q)$ ；而整體時間複雜度為 $\theta(n + q - p)$ ，共新增 $n - p - 1$ 個人工變數，其虛擬碼如下所示。

procedure *initialization*;

For (each node $i \in N_{\hat{O}} \setminus \{s\}$) **do**
 Create a new arc (s, i) ;
 Set $x_{si} := -b_i$, $c_{si} := M$, and $u_{si} := M$;
 Let x_{si} be in basis;
For (each arc $(i, j) : i \in N_D, j \in L(i)$) **do**
 Let $x_{ij} := 0$ and be in basis;

在3.2節中提到基解所對應的圖形含有 $n + q - p - 1$ 條弧，迴圈由D節點所引起，且對每個D節點 i 適當刪除 $q_i - 1$ 條基弧後亦可形成一展開樹；但是這並不代表所有符合這些特性之圖形均為基解圖。如圖4.13，該圖符合基解個數為 $n + q - p - 1 = 6$ ，迴圈亦由D節點引起，且扣除弧 $(2, 5)$ 與 $(3, 4)$ 後為一展開樹；然而若嘗試根據互補差額定理計算其對偶變數，並依照先前計算對偶變數之步驟，令 $\pi_1 = t_1$ 、 $\pi_4 = t_2$ 與 $\pi_5 = t_3$ ，推算其他對偶變數值並求解D節點的聯立方程式後，可發現此兩項方程式無解；因此，一組可行基解之圖形，除必須維持基解圖所擁有的特性外，最後尚需滿足圖形對應的對偶變數為基解（亦即需可依照互補差額定理之聯立方程組推算出對應的對偶變數值）。

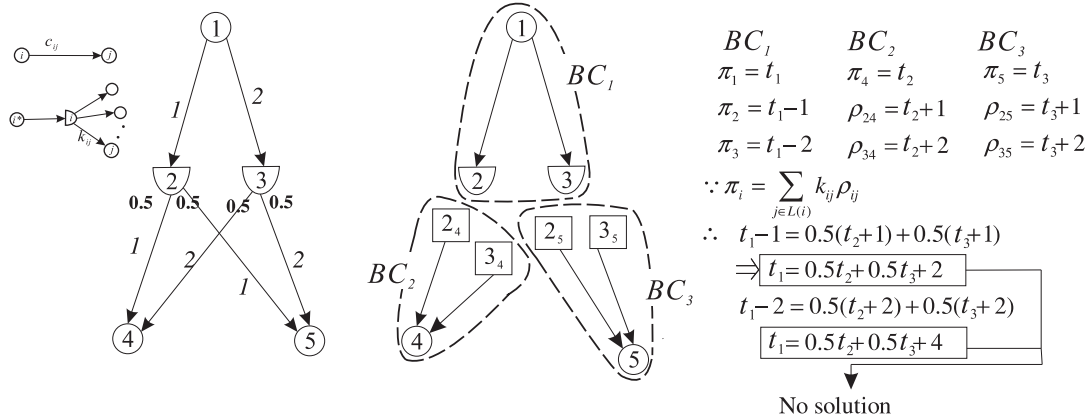


圖 4.13: 非基解圖其對偶變數無解

Lemma 3. 若為精簡過後之網路圖，亦即網路中任兩D節點不相鄰，則 *procedure: initialization* 所構成之圖形可由基弧根據互補差額定理解出其對應的對偶變數，亦即該圖形為主問題的可行基解。

Proof: 步驟1針對所有節點 \hat{O} 節點新增成本為 M 且無容量上限之弧，並適當分配流量值以滿足淨流量；故當步驟1結束後，就可構成由節點 s 到其他 \hat{O} 節點所構成的一對多展開樹，此時若令 $\pi_s = 0$ ，就可依據 $\pi_s - \pi_i = c_{si} : i \in N_{\hat{O}} \setminus \{s\}$ 算出所有 \hat{O} 節點的對偶變數。另外步驟2中對每個D節點選擇其流出弧為基底，由於精簡後之網路任兩D節點不相鄰，因此所有D節點的流出弧必流往 \hat{O} 節點，故可由 $\rho_{ij} - \pi_j = c_{ij} : j \in L(i)$ 計算出所有的 ρ_{ij} 值；而待所有的 ρ_{ij} 值算出後，由於每個D節點的流入弧不為基底，所以可再依照 $\pi_i = \sum_{j \in L(i)} k_{ij} \rho_{ij} : i \in N_D$ 計算出所有D節點的 π_i 值。如圖4.14，為該程序所創設並進行分群後之圖形，可由下至上依序計算對偶變數值。由於該程序之圖形所對應的對偶變數為基解，因此為一個基解圖。

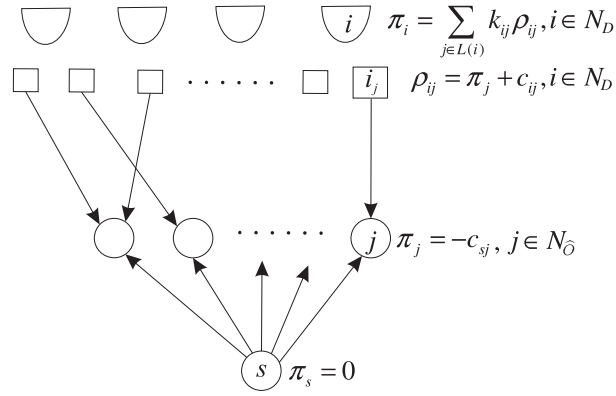


圖 4.14: 初始解示意圖 (適用於精簡過後之網路)

對於D節點而言，挑選所有流出弧為基底並非初始解唯一的選擇方式，只要在與節點 $i \in N_D$ 相連的 $q_i + 1$ 中條弧適當選擇其中的 q_i 條為基弧均能符合初始解的性質；或是也可新增弧 (s, i) (成本值與容量上限亦為 M)，再額外挑選與其相鄰的 $q_i - 1$ 條弧為基底，以上的篩選方式若安排適當亦能搜尋出一初始基解；但本研究基於選擇程序簡單的前提下，只選擇D節點的 q_i 條流出弧為基弧。另外由於MDCP的模式目標為最小化系統成本，所以新增成本為 M 的人工變數並不會影響到原問題的最佳解，且如果當達到最佳解之後人工變數仍為基底，則代表該MDCP模式無可行解。

4.5 單形法、修正單形法與網路單形法的比較

從主問題開始求解的單形法主要有兩種形式：表格形式的單形法（簡稱單形法）與修正單形法（revised simplex method），在使用上也顯得各有優劣。單形法操作較為簡單，亦是單形法理論的基礎，唯其每一個步驟皆需維持並更新全部的限制式矩陣，如果將之求解限制式數目為 $n + q - p$ （限制式矩陣之秩數）、弧數目為 m 的 MDCP 時，需要 $O(m(n + q - p))$ 空間以儲存限制式矩陣，檢查最佳解則需時 $O(m)$ ，另外每做一次 iteration 則需時 $O(m(n + q - p))$ 。而如果採用修正單形法求解 MDCP 時，則可以避免儲存全部的限制式矩陣，使得其儲存空間縮減為 $O((n + q - p)^2)$ ，而每做一次 iteration 亦只需時 $O((n + q - p)^2)$ 更新其所需資訊，但是修正單形法卻需要 $O(m(n + q - p))$ 時間才能更新其 Z 列值（亦即更新 Z 列值代表的對偶變數），因此若要檢查問題是否已到達最佳解則需時 $O(m(n + q - p))$ 。

本研究嘗試以圖形式的操作探討網路單形法，可避免儲存原先單形法或修正單形法所需的矩陣，僅儲存如 c_{ij} 、 x_{ij} 、 u_{ij} 、 $f_{ij} \cdots$ 等弧資訊，或是如節點類型、節點度數、對偶變數 \cdots 等與節點相關之資訊等，因此所需儲存空間僅佔 $O(m + n + q) = O(m)$ ；而最佳解之檢查則採用檢查弧兩端點所對應的對偶變數，確認其差額是否滿足其最佳解條件，故時間複雜度為 $O(m)$ ；另外由 4.2 節與 4.3 節中可知，每執行一次完整的疊代—包括流量的疊代及對偶變數之更新，其時間複雜度為 $O(\min\{p^3 + np, ((n - p)^3 + n(n - p))\} + q)$ 。因此若令 $n' = n - p$ 來代表 \hat{O} 節點之個數，可列三種單形法所需的空間與其各步驟之時間的比較表如下：

表 4.1: 單形法、修正單形法與網路單形法之比較

	simplex (tableau)	revised simplex	network simplex
memory	$O(m(n' + q))$	$O((n' + q)^2)$	$O(m)$
optimality check	$O(m)$	$O(m(n' + q))$	$O(m)$
per-iteration time	$O(m(n' + q))$	$O((n' + q)^2)$	$O(\min\{p^3 + np + q, n'^3 + nn' + q\})$

相較之下，網路單形法的儲存空間佔了極大的優勢，而在最佳解檢查時間方面亦只需要 $O(m)$ ；但是為節省空間所付出之代價就反應在流量疊代以及對偶

變數更新上—最多需要求解大小為 $\min\{p, n'\}$ 的聯立方程組，而這也使其與其他兩種單形法在理論上不易比較出 per-iteration 所花的時間多寡。

若單以網路單形法中所需求解的聯立方程式數目為 $\min\{p, n'\}$ 來看，似乎在 $n' = p = n/2$ 時對網路單形法的時間複雜度會最為不利；但本研究所設限之聯立方程組數目為「其鄰近弧皆為基弧的 D 節點個數」，乃為必需求解方程式數目的上界，實際情況有可能網路中之部分 D 節點 i 與 $q_i + 1$ 條弧相連，而使得需要求解的聯立方程組數目遠小於 $\min\{p, n'\}$ 。如圖 4.15 所示的某 MDCP 之基解圖，將之

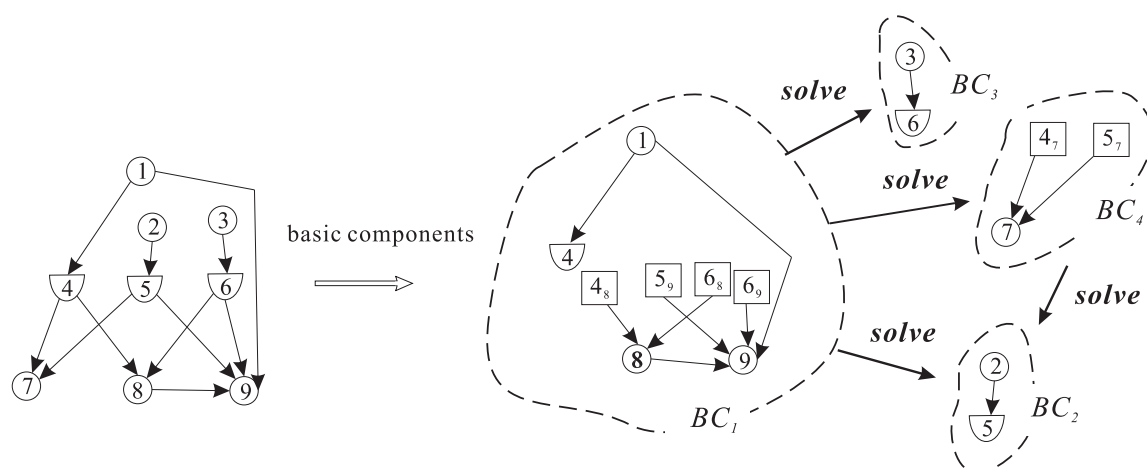


圖 4.15: D 節點皆與所有相連弧之基解圖亦有可能不用求解聯立方程組

分類為四個基群集後，由於 BC_2 與 BC_3 皆只與一 D 節點相連，因此可留至最後計算；只要先令 BC_1 的群集變數 $t_1 = 0$ ，再由節點 4 的對偶變數相依方程式求解 BC_4 的群集變數 t_4 ，最後分別由節點 5 與節點 6 求解出 BC_2 與 BC_3 的群集變數 t_2 與 t_3 ，即可求得全部的對偶變數值。而流量疊代之情形亦相同，若進入弧發生在同一群集，或是其他 D 節點為可刪除之分支時，同樣不用求解數目為 $\min\{p, n'\}$ 的聯立方程組。因此，實際欲求解方程組之數目與圖形內群集間的連接情形具有高度的相關性。

聯立方程組的矩陣特性乃決定於群集間的相連狀況，其中最複雜之情形即為每個群集間均互相連接，而使得必須求解之矩陣的每一個元素皆不為 0。如圖 4.16，此範例中由於 $p \leq n'$ 且群集均互相連接，因此勢必求解此三個 D 節點所構成的聯立方程組，且求解矩陣內每一個元素皆不為 0。然而當此情形發生之時，

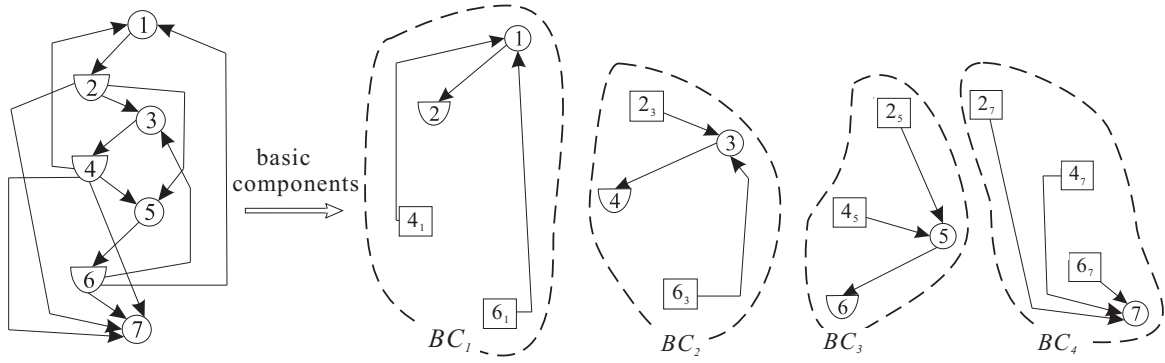


圖 4.16: 每一個群組均互相連接

表示每一個D節點 i 至少需要 p 條流出弧，使得流出弧總數為 $q = \sum_{i \in N_D} q_i \geq p^2$ ，連帶使得修正單形法 per-iteration 求解時間變為 $O(n' + q)^2 = O((n' + p^2)^2) \geq O(p^4)$ ；此外，當 $p \geq n'$ 時亦可得出類似的結果，最多 n' 個群集皆互相連接，使得修正單形法 per-iteration 求解時間變為 $O(n'^4)$ 。原則上當 $q \geq \min\{p\sqrt{p}, n'\sqrt{n'}\}$ 時，由 $O((n' + q)^2) \geq O((n' + \min\{p\sqrt{p}, n'\sqrt{n'}\})^2) \geq O(\min\{p^3, n'^3\})$ 可得網路單形法的求解時間較有優勢；但在 $q \leq \min\{p\sqrt{p}, n'\sqrt{n'}\}$ 時，實際求解時間則取決於群集間的相連狀況，因此若與修正單形法相比，無法斷定孰優孰劣。

雖然修正單形法 per-iteration 之時間複雜度為 $O((n' + q)^2)$ ，但如要確定最佳解，亦即檢查Z列值所對應的對偶變數是否可行時，仍需要 $O(m(n' + q))$ 時間；即表示修正單形法做一次完整疊代所需之時間仍為 $O(m(n' + q))$ 。若由此點分析，則當 $m \geq \min\{p^2, n'^2\}$ 時，本研究所提出之網路單形法較佔優勢。因此當MDCP問題D節點與 \hat{O} 節點比例越懸殊，且網路中弧的個數越多，則本研究所提出的網路單形法將越有利。

4.6 小結

在本章中完整呈現了網路單形法求解MDCP的每一個詳細步驟。不但發展出有別於Fang and Qi(2003)，且以圖形操作為概念的對偶變數求解程序，亦發展出他們未深入探討的流量疊代、對偶變數更新與初始解設立等等網路單形法之詳細程序。在求解對偶變數時，我們利用基解圖的適當分群，從而發展出有系統的

求解方法。在流量疊代中，由於MDCP之D節點所造成的迴圈，使得當進入弧兩端點發生在兩不同群集時，無法立即判斷出可以通過流量的子圖以及流量的比例關係；因此，本研究先採取刪除不可通過流量分支的程序，再設定適當的變數，並同樣以群集為單位來進行求解，之後即可進行最小值檢定，選出退出弧並更新流量。而在對偶變數更新中，我們則利用進入弧與退出弧造成群集間連通關係的改變來有效率地更新對偶變數，避免重新用群集的分群來計算所有的對偶變數。另外，本研究亦提出修正的大M法設置MDCP的初始解，並可證明所搜尋之圖形為一可行基解。

網路單形法可以省略原本單形法中所需儲存的龐大矩陣資訊，但其代價就將反應在流量疊代以及對偶變數更新之過程所需求解的聯立方程式，而求解方程式所需的時間與群集間相連的狀況呈高度相關。理論上來說，網路單形法的理論複雜度無法確保一定較原始的單形法佔優勢；然而以實務上來說，因為網路單形法能夠大大地節省儲存空間，且其大部分的子步驟均為有效率的圖形式推導，因而不論空間與時間皆能較原始的網路單形法擁有更好的求解效率。

第五章

MDCP 相關議題

本章主要探討四項與 MDCP 相關之議題。第一項議題探討的是 Fang and Qi(2003) 在文獻中曾提到的 dead-cycle 現象，他們認為網路單形法在做基變數進入與退出的選擇時，樞紐圖 (pivoting graph) 中若有兩條弧均來自於同一個 D 節點，即形成 dead-cycle，並認為該情形無法通過流量，無助於單形法的計算；然而，根據我們進一步的探討後可發現 dead-cycle 其實並不存在，Fang and Qi(2003) 之誤判乃肇因於對基解圖的判讀錯誤所致，而此結果又將連帶地影響樞紐圖的判斷。在第二項議題中，我們提出有別於第三章中以 u_s 為中心的轉換方式，改以 T 節點的需求為中心以轉換原 MDCP，而此種新的轉換方式同樣能以確切值來表達每個節點的淨流量值。第三項議題探討另一種流量疊代的分群方法，Ting(2005) 在求解包含 D 節點的最大流量問題時，提出 compatible component 用以計算所搜尋子圖上的流量比例關係，該方法同樣可應用在樞紐圖中流量的疊代；因此本議題將探討此方法應用於流量疊代時的操作情形，並與本研究所提出的基群集分群法互相比較。最後，在本章的第四項議題中將以網路單形法探討包含 D 節點的最大流量問題，並用以解釋 Lin(2005) 論文中提到的對偶變數分群之現象。

5.1 樞紐圖的錯誤判讀－不存在的 Dead-cycle

Fang and Qi(2003) 在其論文中曾提到，當進入弧加入圖形後而形成樞紐圖時，如果樞紐圖中有兩條弧均來自於同一 D 節點，即形成 dead-cycle，並認為 dead-cycle 無助於單形法的計算。圖 5.1 為 Fang and Qi(2003) 所提出 dead-cycle 範例，弧 (5, 9) 為進入變數。此範例中弧 (4, 9)、弧 (4, 10) 均來自同一 D 節點－節點

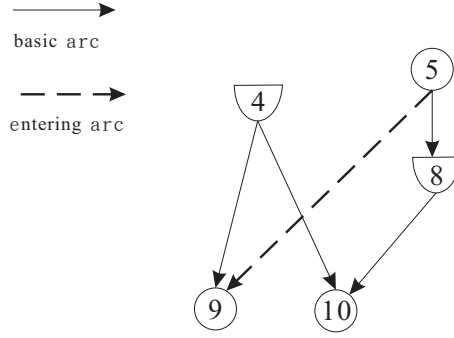


圖 5.1: 樞紐圖中形成 dead-cycle (來源: Fang and Qi, 2003)

4，形成 dead-cycle，因為無法依照 $5 \rightarrow 9 \rightarrow 4 \rightarrow 10 \rightarrow 8 \rightarrow 5$ 之方向推送流量，所以在網路單形法的操作過程中可以忽略。然而經過更詳盡的分析思考之後，我們會發現 dead-cycle 其實並不可能存在，其原因將闡述如下。

MDCP 亦是線性規劃的特例問題，首先回顧一下線性規劃的標準形式：

$$\begin{aligned} \min \quad & cx \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

今假設此問題表示一個網路問題， x 表示由 x_{ij} 所構成的弧向量，維度為 $m \times 1$ ； c 表示網路中之成本向量，維度為 $1 \times m$ 。而假設網路問題之函數限制式數目為 k ，因此 A 矩陣之維度為 $k \times m$ ，其中 a_{pl} 表示矩陣中第 p 列第 l 行元素。網路單形法中的操作事實上則與單形法中的每一個步驟相符，其中網路單形法中依據樞紐圖推送流量並選擇退出變數之步驟則與單形法中的最小值檢定相呼應，表示若第 l 條弧為進入變數，依據 $r_{\hat{p}} = \min\{b_p/a_{pl} : a_{pl} \neq 0, \forall p = 1, \dots, k\}$ 選擇具有最小比例之弧為退出變數。 $a_{pl} > 0$ 代表第 p 列其基變數會隨著第 l 條弧的流量增加而減少； $a_{pl} < 0$ 代表第 p 列其基變數會隨著第 l 條弧的流量增加而增加； $a_{pl} = 0$ 則代表第 p 列其基變數不隨著第 l 條弧增加而變化。所以，如果 dead-cycle 存在，並使得網路無法通過流量而無法選出退出弧，就相當於在單形法中的第 l 條弧為進入變數卻無法尋找出任何 $a_{pl} \neq 0 : p = 1, \dots, k$ 之元素進行最小值檢定而使得 $r_{\hat{p}}$ 不存在，也就是第 l 欄中的 $a_{pl} = 0 : \forall p = 1, \dots, k$ ，由於此種情況並不存在於單形法的運作中，亦即表示 dead-cycle 並不存在。

誤判 dead-cycle 存在的原因為其對基解圖的判斷方式不正確所致。網路模式中的每一個節點均代表其獨自的限制式，又由 (2.6)、(2.7) 與 (2.8) 可知 MDCP 中每一個 S 節點、T 節點、O 節點各自需要一條限制式去定義該節點；而由 (2.10) 可知一個 D 節點 i 假如有 q_i 條分流弧，則需要 q_i 條限制式去定義此節點。因此在轉換前的模式中，基解圖加入進入弧形成樞紐圖後，其弧個數應為 $n + q - p + 1$ 才能代表樞紐圖的合理性。由此觀之，即可發現圖 5.1 犯了兩項錯誤：第一，節點 4、節點 8 為 D 節點，但該圖並未畫出節點 8 其他分流弧與其相關節點的圖形；第二，即使假設節點 8 只有一個流出弧，代表一條限制式，但節點 4 有 2 條流出弧，表示該節點由 2 條限制式所定義；而該圖所繪出的 5 個節點應代表 6 條限制式，樞紐圖中的弧個數應為 $6 + 1 = 7$ 條，但是其所繪出的弧只有 5 條，顯示此樞紐圖並非一個完整的樞紐圖，自然無法計算其流量比例關係。以圖 5.2 為例，雖然 (1, 4) 為進入弧，並與弧 (2, 4)、弧 (2, 5)、弧 (3, 5) 及弧 (1, 3) 形成 dead-cycle，但在繪出完整的基解圖之後，可以依據此樞紐圖推送流量，圖中的正負號僅代表流量增加或減少之情形，實際數值則需視 D 節點分流比例係數而定。

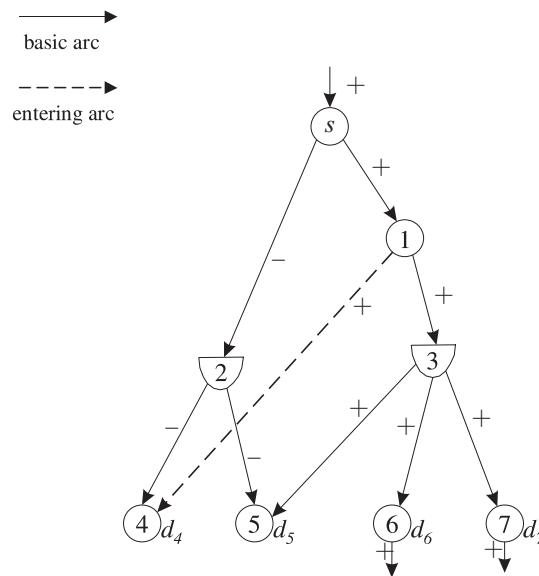


圖 5.2: dead-cycle 並不存在，可依此圖推送流量

除 dead-cycle 外，在 Fang and Qi(2003) 的論文中亦可發現他們對樞紐圖形判斷方式有誤的其他證據。舉例而言，圖 5.3(a) 為其論文中所探討的樞紐圖，在此

圖例中，共有8個節點，其中包含一個D節點，且該節點有3條流出弧，因此樞紐圖中的弧個數應為 $n + q - p + 1 = 8 + 3 - 1 + 1 = 11$ ；但圖中發現弧僅有10條，顯示其圖型判斷錯誤。因此，若嘗試將此樞紐圖上之流量均以同一變數 w 表示，形成如圖5.3(b)時，即可發現節點 s 並不滿足流量守恆限制式，若欲滿足流量守恆限制式，勢必造成 $w = 0$ 而造成此圖形無法推送流量。

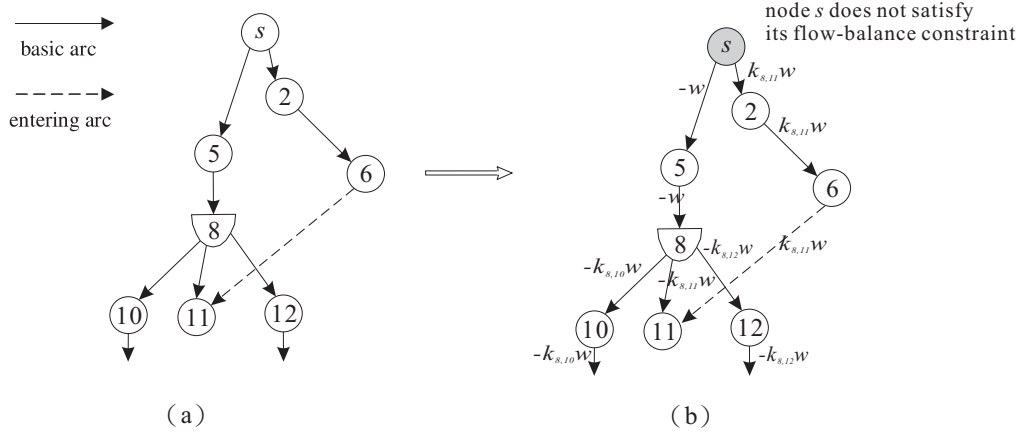


圖 5.3: Fang and Qi 對樞紐圖判讀錯誤 (來源:修改自 Fang and Qi, 2003)

5.2 以 T 節點之需求為基礎的模式轉換

在3.1節中我們提出了一套方法轉換原 MDCP 模式，該方法主要是以 u_s 為中心所做的轉換；但是如果原先的產能 $u_s = \infty$ ，雖然在應用上可設其為一足夠大的正數，但所設置的大小又將是一項難題。因此，如果改以 T 節點 i 的需求 d_i 為中心，則可提出另外一種轉換模式的方法，而且轉換後每一個節點同樣能以一確切值來表示其淨流量值。

首先，先對所有 T 節點 i 的需求下限限制式 (2.15) 加入超額變數 x'_i ，可得：

$$\begin{aligned} x_i &\geq d_i && \text{for } i \in N_T \\ \implies x_i &= d_i + x'_i && \text{for } i \in N_T \end{aligned} \quad (5.1)$$

之後同樣利用 (5.1) 代換 x_i ，不同之處在於我們將超額變數 x'_i 回流至單一的 S 節點

s ，亦即使 x'_i 變為 x_{is} ，再代換 (2.7) 可得：

$$\begin{aligned}
& x_i - \sum_{j \in E(i)} x_{ji} = 0 && \text{for } i \in N_T \\
\Rightarrow & x'_i + d_i - \sum_{j \in E(i)} x_{ji} = 0 && \text{for } i \in N_T \\
\Rightarrow & x_{is} - \sum_{j \in E(i)} x_{ji} = -d_i && \text{for } i \in N_T
\end{aligned} \tag{5.2}$$

由於 MDCP 模式除了 S 節點與 T 節點外，其他節點之淨流量皆為 0，代表送至需求節點之流量必定來自供給節點，可得出 $x_s = \sum_{i \in N_T} x_i$ ，藉由此式與 (5.2) 所得之結果代換節點 s 之流量守恆限制式 (2.6) 可得：

$$\begin{aligned}
& \sum_{j \in L(s)} x_{sj} - x_s = 0 && \text{for } s \\
\Rightarrow & \sum_{j \in L(s)} x_{sj} - \sum_{i \in N_T} \sum_{j \in E(i)} x_{ji} = 0 && \text{for } s \\
\Rightarrow & \sum_{j \in L(s)} x_{sj} - \sum_{i \in N_T} (x_{is} + d_i) = 0 && \text{for } s \\
\Rightarrow & \sum_{j \in L(s)} x_{sj} - \sum_{i \in N_T} x_{is} = \sum_{i \in N_T} d_i && \text{for } s
\end{aligned} \tag{5.3}$$

(5.2) 及 (5.3) 即為每一 T 節點 i 與單一 S 節點 s 轉換後的流量守恆限制式，其中節點 s 的淨流量 $b_s = \sum_{i \in N_T} d_i$ ，而節點 $i \in N_T$ 之淨流量 $b_i = -d_i$ 。最後，為了能使供給節點的產能上限 u_s 能夠以弧容量的型態加入模式中，再將節點 s 分離成兩個節點 s 與 s' ，先後令 $L(s') = L(s)$ 、 $E(s') = \{s\}$ 與 $L(s) = \{s'\}$ ，使得節點 s 的流量守恆限制式為：

$$x_{ss'} - \sum_{i \in N_T} x_{is} = \sum_{i \in N_T} d_i \quad \text{for } s \tag{5.4}$$

而節點 s' 的流量守恆限制式為：

$$\sum_{i \in L(s')} x_{s'i} - x_{ss'} = 0 \quad \text{for } s' \tag{5.5}$$

如此一來，就能將 u_s 轉換為弧 (s, s') 的容量限制，如下所示：

$$\begin{aligned}
 x_s &\leq u_s \\
 \Rightarrow \sum_{i \in N_T} \sum_{j \in E(i)} x_{ji} &\leq u_s \\
 \Rightarrow \sum_{i \in N_T} x_{is} + \sum_{i \in N_T} d_i &\leq u_s \\
 \Rightarrow x_{ss'} &\leq u_s = u_{ss'} \quad (5.6)
 \end{aligned}$$

節點 s' 之淨流量為 0，轉換過後可歸類為 O 節點，而 (5.2)、(5.4) 與 (5.5) 即為模式轉換後產生的限制式，圖形上之轉變如圖 5.4 與圖 5.5 所示。

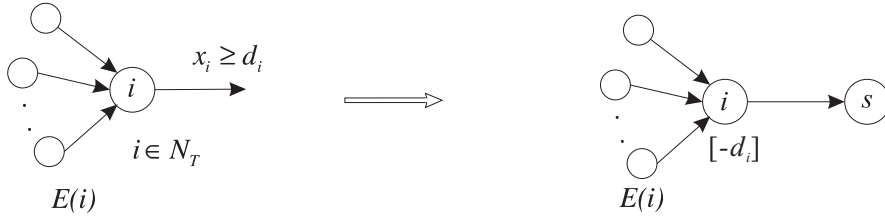


圖 5.4: 以需求為基礎轉換 T 節點

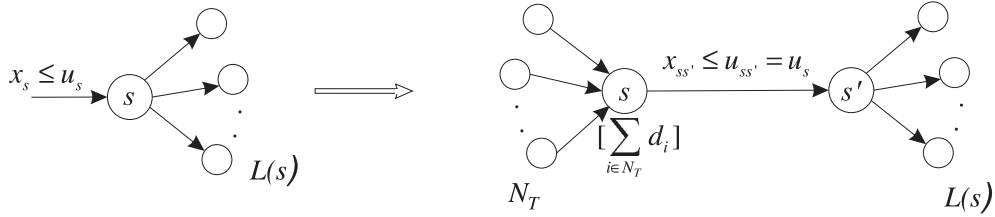


圖 5.5: 以需求為基礎轉換 S 節點

引用相同的概念，亦可將多個 S 節點、多個 T 節點的 MDCP 轉換為一對多的 MDCP。首先，新增一個節點 s ，令所有 T 節點 j 流入量與需求量的差額 ($x_{js} = \sum_{k \in E(j)} x_{kj} - d_j$) 回流至節點 s ，並令節點 s 的淨流量為 $\sum_{j \in N_T} d_j$ ；最後，再新增弧 (s, i) 連通到節點 $i \in N_S$ ，並將 u_{si} 設為原先節點 $i \in N_S$ 的產能上限，而原先 S 節點經轉換後則轉變為淨流量為 0 的 O 節點，轉換之圖形如圖 5.6 所示。

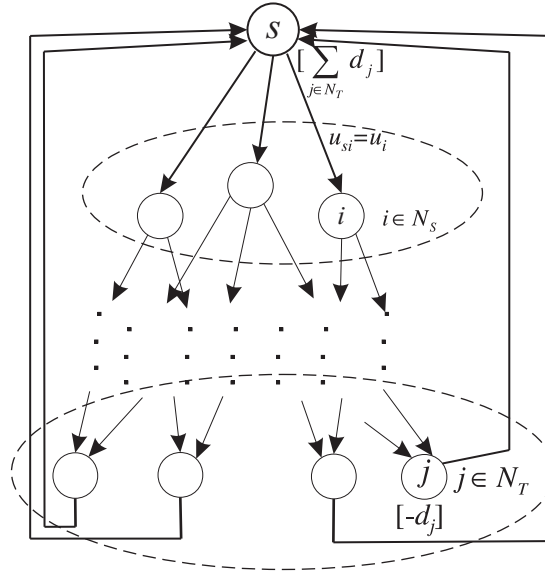


圖 5.6: 以需求為基礎轉換多個供給點、多個需求點之MDCP網路

不論是一對多或是多對多的MDCP，在以T節點*i*的需求 d_i 為基礎進行轉換後，同樣可轉換所有相關節點為新的一對多的MDCP模式。如果同樣以 $G = (N, A)$ 表示轉換後的網路系統， N 代表節點集合， A 代表弧集合，且 $|N| = n$ ， $|A| = m$ ，可列出一對多的MDCP模式如下：

$$\begin{aligned}
 & \min \sum_{(i,j) \in A} c_{ij} x_{ij} && (\text{MDCP}_{new2}) \\
 \text{s.t.} \quad & \sum_{i \in L(s)} x_{si} - \sum_{i \in N_T} x_{is} = \sum_{i \in N_T} d_i && \text{for } s \\
 & \sum_{j \in L(i)} x_{ij} - \sum_{j \in E(i)} x_{ji} = 0 && \forall i \in N_O \cup N_D \\
 & x_{is} - \sum_{j \in E(i)} x_{ji} = -d_i && \forall i \in N_T \\
 & x_{ij} - k_{ij} x_{li} = 0 && \forall i \in N_D, (l, i) \in A \\
 & 0 \leq x_{ij} \leq u_{ij} && \forall (i, j) \in A
 \end{aligned}$$

轉換後之模式，如同3.1節之模式般，所有節點的流量守恒限制式與MCF相同，皆能夠以一確切值表示其淨流量值，也因此，除了能夠符合Lemma 1所提出的基

解性質外，亦能夠使用本研究所提出的網路單形法將之求解。另外，在上述轉換過程中，我們將節點 $i \in N_T$ 的流入量與需求量下限之差額 ($x_{is} = \sum_{j \in E(i)} x_{ji} - d_i$) 回流至節點 s ，此類轉換之用意一方面是為了解供給量與需求量之間的關聯，另一方面則藉此做為確保基解圖為相連圖所採取之手段。在原 MDCP 模式中，並不會有流量回流至供給節點，若對照原模式，實際上節點 s 送出的流量為 $\sum_{i \in N_T} d_i + \sum_{i \in N_T} x_{is}$ ，其中 $\sum_{i \in N_T} d_i$ 為滿足 T 節點需求所送出的供給量， $\sum_{i \in N_T} x_{is}$ 則可視為遵守 D 節點分流限制所額外送出之供給量；而對於任一 T 節點 i 而言，流出弧的流量 x_{is} 代表該節點所接收多於需求 d_i 之流量。

5.3 應用 compatible component 求解樞紐圖流量比例

5.3.1 compatible component 求解概念

Ting(2005) 在求解包含 D 節點的最大流量問題時提出了 compatible component 之方法，此方法將所有節點度數超過 2 的 \hat{O} 節點視為分割節點 (dividing node)，利用這些節點的分割將圖形拆解成數個子集合，之後再由這些分割節點計算圖形中的流量比例關係。圖 5.7(a) 為一可推送流量的殘餘網路圖 (residual network)，

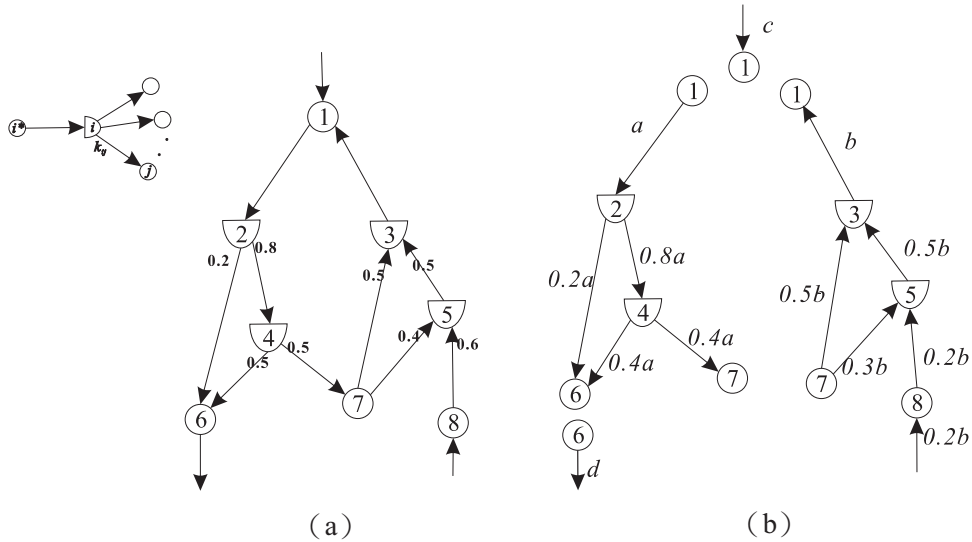


圖 5.7: Compatible component(來源：Ting, 2005)

節點 1 為起點，節點 6 與節點 8 為終點；其中節點 1、節點 6 與節點 7 為節點度數

超過2的分割節點，將之拆解後可將圖形分割成四個群集，如圖 5.7(b) 所示。在 compatible component 中，同一群集內的流量關係均能以同一變數表示，因此若求解此三個節點的流量守恆限制式，得出 $b = c = 0.5a$ 與 $d = 0.6a$ ，即可將整個子圖化為以 a 表達之形式，而得出子圖上的流量比例關係。

若分群後得到 w 個 compatible components (對應到 w 個流量變數) 與 y 個分割節點 (對應到 y 組流量守恆方程式)，Ting(2005) 依照 y 與 w 的相對關係可分為下列幾種情形討論：

1. $y \geq w$ ：在此種狀況中，由於將所有變數均化為單一變數只需要 $y - 1$ 條方程式，因此此種情況會求得所有變數為 0 的唯一解。以圖 5.8 為例，圖中節點 1 與節點 5 皆為節點度數為 3 的分割節點，因為 $y = w = 2$ ，使得求解分割節點的流量守恆限制式 $a + b = a$ 與 $0.4a = 0.6a$ 後僅能得到 $a = b = 0$ 的唯一解。雖然節點 5 為節點度數為 3 的分割節點，但因該節點不為兩個群集的交集點，所以該節點的流量守恆限制式在求解後將迫使 $a = 0$ 。

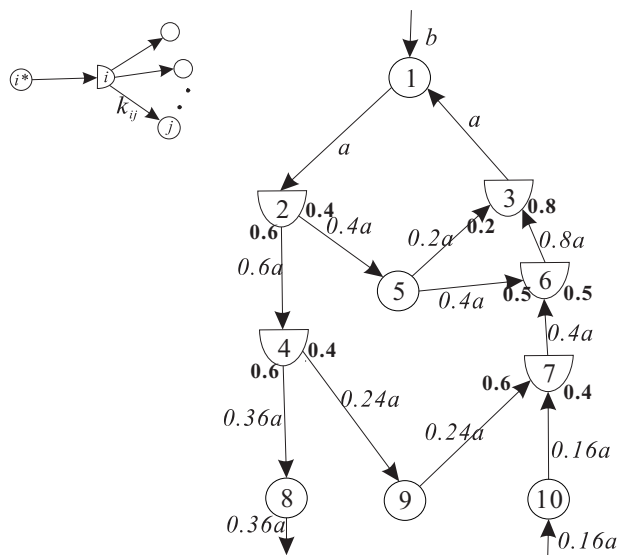


圖 5.8: 不可通過流量之子圖 (來源：Ting, 2005)

2. 當 $y = w - 1$ ：若此時，在求解大小為 y 的聯立方程組之後即可將子圖中的變數均化為同一變數表示，但所求解出的解為了要遵守變數的非負限制式，而導致流量為 0，如圖 5.9 所示。

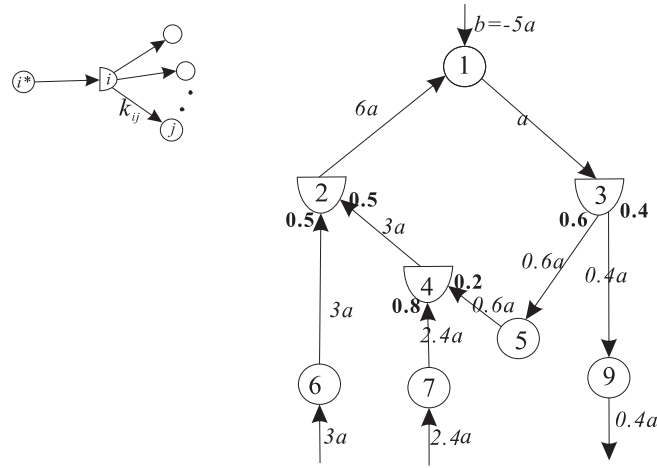


圖 5.9: 求解後之變數違反非負限制 (來源：Ting, 2005)

3. 當 $y \leq w - 2$ ：此種情形可能無法將求解出的所有群集變數均以同一變數表示，不過 Ting(2005) 尚無法尋找出符合此種情況的範例。雖然 Ting(2005) 並無提出證明方式證明此情形不存在；但是她認為如果有該種情形發生，或許代表有兩個以上各自獨立的子圖，而可以將之拆解為數個子圖並個別求解。

5.3.2 應用 compatible component 於流量疊代

Ting(2005) 所提出的 compatible component 之概念亦可應用於網路單形法中流量疊代，但由於其搜尋出之子圖不包含節點度數為 1 的 \hat{O} 節點或是節點度數為 q_i 的 D 節點 i ，所以如欲將其方法應用於流量疊代，仍需要先刪除不可通過流量之分支後才可應用。另外，由於流量疊代可以依據樞紐圖進行流量比例之計算，不需事先搜尋出殘餘網路圖，求解之值若為負僅代表該弧流量減少，因此可以放寬原本 compatible component 中所設置變數的非負限制。

原先 Ting(2005) 所求解的最大流量問題乃源自於 Fang and Qi(2003) 提出的模型，未經過模式轉換；由於轉換前流量守恒限制式的秩數(rank)為 n ，而轉換過後流量守恒限制式秩數變為 $n - 1$ ，包含一條多餘的限制式。所以假如分群後得到 w 個 compatible components 與 y 個分割節點，那麼僅在 $y = w$ 時，才能保證樞紐

圖中的改變量可解；之後再視進入弧 (k, l) 上之流量原先為達下界或上界來限制 $f_{kl} = 1$ 或 $f_{kl} = -1$ ，才可求解出樞紐圖中的流量比例。

以一簡單例子來看，圖 5.10(a) 表示刪除分支後的樞紐圖，由於節點 2 與節點 6 皆為節點度數超過 3 的 \hat{O} 節點，因此可藉由此兩節點的分割而分為兩群，由節點 2 與節點 6 皆可得到 $0.7a = b$ 之等式，去除其中一條多餘的限制式後，再加上 $b = 1$ ，即希望往 $(2, 6)$ 正向推送一單位流量，就可求解出樞紐圖中的流量比例而得出 $a = -10/7, b = 1$ 。

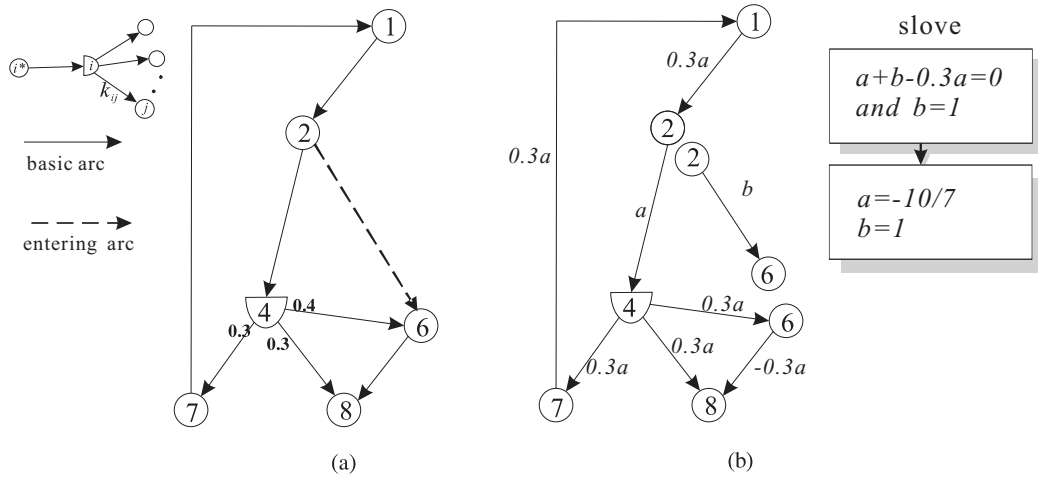


圖 5.10: 應用 compatible component 求解樞紐圖流量比例

Lemma 4. 利用 compatible component 對刪除分支後的樞紐圖進行分群時，如果圖中包含 y 個分割節點 (節點度數超過 3 的 \hat{O} 節點)，則在分群後群集數目恆為 y 群。

Proof: 樞紐圖可視為基解圖加入進入弧後所形成的圖形，所以若先移除進入弧 (k, l) 後，該圖形亦會符合 Lemma 1 中的基解特性—弧個數為 $n + q - p - 1$ 。又傳統 MCF 的樞紐圖含有唯一之迴圈，所有的節點度數均為 2；因此在 MDCP 的樞紐圖中，節點度數超過 2 之節點必為 D 節點所造成之結果，所以若對所有的分割節點適當扣除共 $q - p$ 條弧後，圖形可縮減為一展開樹，且使得所有 \hat{O} 節點的節點度數必不超過 2。之後，若分割此 y 個分割節點，則可將圖形分為 $y + 1$ 群，加入進入弧後將會合併兩群集使得群集共為 y 群。另外，由於之前所移除的 $q - p$ 條弧均

與分割節點相連，且其他不為分割節點的 \hat{O} 節點均已適當分群；因此若將先前移除之弧再放入圖形中時，則一定可以適當歸類於此 y 群集，即為使用 compatible component 分群之結果。□

值得一提的是，在使用 compatible component 對刪除分支後的樞紐圖進行分群時，我們可透過 Lemma 4 證明基解圖中若對 y 個分割節點進行分割，其分群數目恆為 y 群，不需要像 Ting(2005) 一樣依照群集與分割節點的相對關係分為三種情況討論；至於為何 Ting(2005) 在推送流量時會遭遇到如圖 5.8 中不可通過流量的子圖，則可用單形法的角度去解釋。

Ting(2005) 在求解包含 D 節點的最大流量問題時，其演算法所搜尋出的子圖，其實可視為網路單形法中流量疊代時所產生的樞紐圖；由於原 MDCP 的基弧個數為 $n + q - p$ ，因此基解圖在加入進入弧而形成樞紐圖時，其弧個數必須為 $n + q - p + 1$ 才能代表圖形可過流量。由此點觀之，圖 5.8 存在 10 個節點，其中有 5 個 D 節點，所有 D 節點的流出弧個數總和為 9，因此弧個數應為 $10 + 9 - 5 + 1 = 15$ 才得以代表一個可計算流量比例的樞紐圖，但此圖弧個數僅有 14 條弧，其數目不足以構成一個可推送流量的子圖；但是只要再加一條弧之後，例如在圖 5.8 加入弧 $(9, 8)$ 後，該圖形即可推送流量，如圖 5.11 所示。

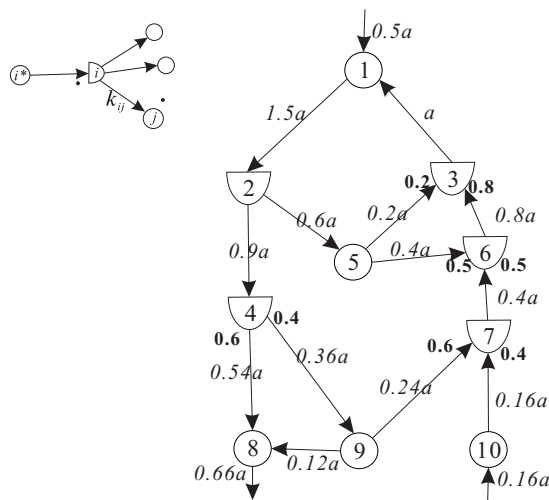


圖 5.11: 加入弧 $(9, 8)$ 後，圖形可通過流量

5.3.3 compatible component 與基群集之比較

應用 compatible component 求解流量疊代時，提供一套與本研究所發展的基群集分群法不同的計算方式。compatible component 之方法著眼於節點度數大於 2 的 \hat{O} 節點，其中不論 $n-p$ 與 p 的相對大小為何，如果樞紐圖中每一個 \hat{O} 節點均為分割節點時，那麼使用 compatible component 最多可能分為 $n-p$ 群，需要求解 $n-p$ 條聯立方程組。與之相比，本研究使用基群集分析時，可保證刪除分支後樞紐圖中的 D 節點個數小於或等於 \hat{O} 節點個數 (4.2.2 節， $\tilde{p} \leq \min\{p, n-p\}$)。另外，由於分群數目牽扯到所需設置變數的多寡，進而影響所需求解的聯立方程組，所以若以分群數目的大小為判斷準則，那麼 compatible component 與本研究所提出的基分群則各有優劣。

假如樞紐圖中節點度數大於 2 的 \hat{O} 節點多於 D 節點個數時 ($y \geq p$)，使用 compatible component 的分群數 y 就會多於基群集的個數，使得本研究之分群法較優。但是如果樞紐圖中的分割節點比 D 節點的個數為少時 ($y \leq p$)，compatible component 的分群數就會優於本研究所提出的基群集分群法。以圖 5.12(a) 為例，在此圖中節點 1 與節點 7 為分割節點，以 compatible component 分群後可分為兩群，而且在適當設置變數後亦可標示出這兩群集中的流量變化量，如圖 5.12(b) 所示；如果利用基群集進行分群則可分為 3 群，而再對 D 節點的流入弧設置變數，並先設節點 3、節點 4 與節點 7 為群集的根節點後，亦可標示樞紐圖中的流量比例，如圖 5.12(c) 所示。

由圖 5.12(c) 中的範例來看，我們可發現利用基群集推算流量比例的方法仍有其改善空間，由於基群集推算的方法主要是根據各群集內之節點為展開樹之特性，再對每一個群集進行推算；因此，如果某群集內的樹僅為一個無分支的樹，亦即起點為某一個 D 節點 (i 或 i_j)，途中至少經過一個 \hat{O} 節點，而終點為另一個 D 節點 (i 或 i_j) 的一條路徑，使得群集內根節點的節點度數等於 2 時，那麼本研究針對每個 D 節點的流入弧設置變數的方法就會造成設置變數數目上的浪費。

由此觀之，雖然我們針對包含 p 個節點的樞紐圖可以得到 p 個基群集，但實際上可能並不需要對這 p 個 D 節點的流入弧均設置變數；我們可先尋找出至

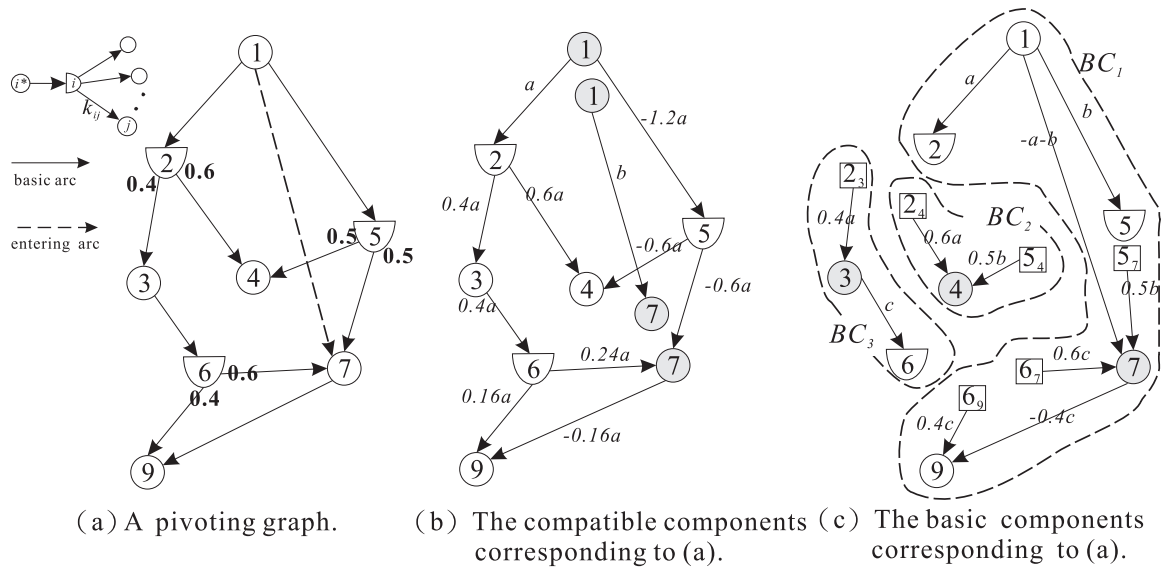


圖 5.12: compatible component 與基群集分群方式不同

少包含一個節點度數超過2的 \hat{O} 節點之群集，並同樣對該群集內D節點的流入弧設置變數。之後，先針對此D節點繼續往下推算流量比例值，若遇到相連弧未標示變數的D節點則可利用已設置的變數給予標示，進而利用分流比例限制式推算該D節點相連弧的流量比例係數；而如果遇到節點度數為2的 \hat{O} 節點則可繼續往下推算，之後重複推算的過程直至遭遇到節點度數超過2的 \hat{O} 節點而停止。在引用此項法則之後，由於原先那些群集內之樹為「路徑」的群集必定有一端與D節點相連，因而使得上述的推算方法必可由該端點開始逐一推算出該路徑上所有弧的流量比例值，使其得以兼顧基群集與compatible component兩者的優點。以圖5.12(c)中的 BC_3 為例，我們可從弧(1,2)往下推算而得弧(3,6)的流量比例係數為 $0.4a$ ，而不用對(3,6)另外設置變數。此法則若引用得宜，甚至可進一步減少所需設置的變數，而縮減所需求解聯立方程組的大小。舉例而言，假如圖5.12(c)中，我們僅對D節點2的流入弧(1,2)設立變數 a ，只要依循 $(1,2) \rightarrow (2,3) \rightarrow (2,4) \rightarrow (3,6) \rightarrow (6,7) \rightarrow (6,9) \rightarrow (5,4) \rightarrow (5,7) \rightarrow (1,5)$ 之順序，利用 a 即可標示出這些弧的流量比例，亦推算出樞紐圖中所有D節點流入弧的流量比例係數，之後即可進一步推得弧(1,7)與(7,9)的流量比例(見圖5.13)；因此只需要一個變數 a 就可標示出樞紐圖中所有的流量比例係數。

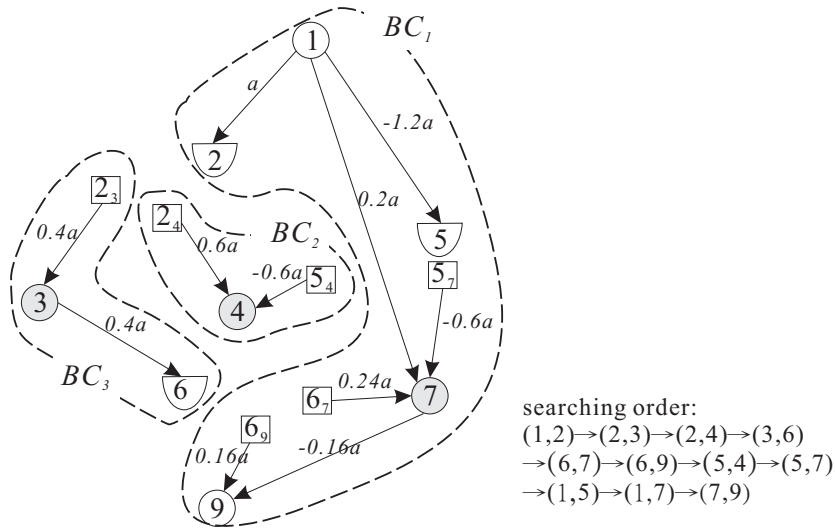


圖 5.13: 適當的搜尋可減少樞紐圖中所設置的變數個數

引用類似的觀念，當使用基群集求解對偶變數或是更新對偶變數時，我們亦可能可以不用對每個群組設置變數，例如在圖 5.14 中，我們僅對 BC_1 與 BC_4 設置變數 t_1 與 t_4 ，接著由 $\pi_5 = 0.5\rho_{54} + 0.5\rho_{57}$ 與 $\pi_6 = 0.4\rho_{69} + 0.6\rho_{67}$ 分別計算出 $\pi_6 = t_4 + 0.4$ ，而 $\rho_{54} = 2t_1 - t_4 - 5$ ，之後可繼續推算 BC_2 與 BC_3 內的對偶變數；因此，只需用兩個變數就可標示出四個群集內的對偶變數，減少了所求解聯立方程組的大小。

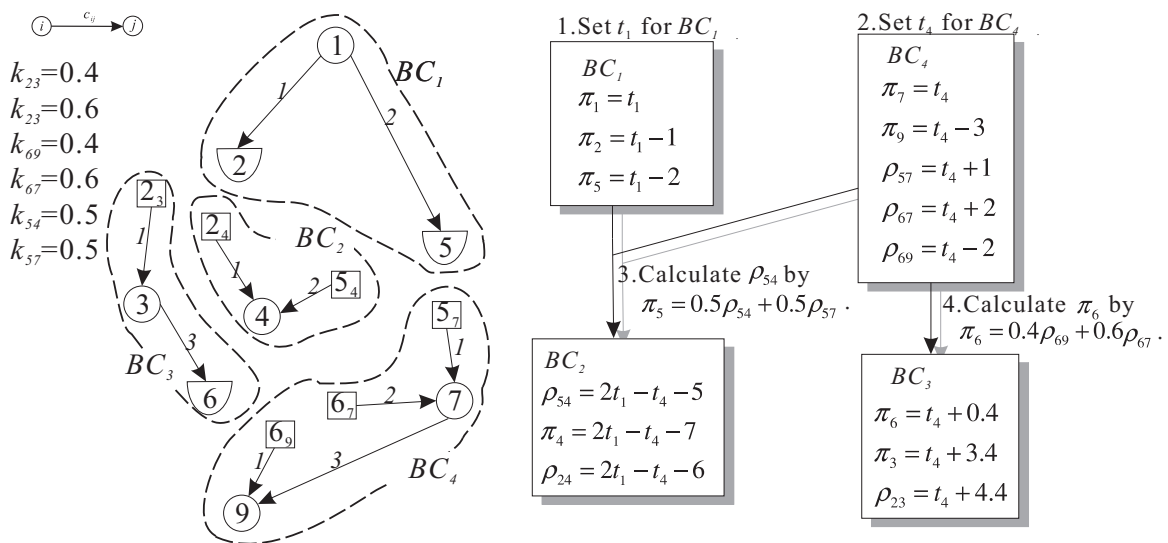


圖 5.14: 視群組特性可減少計算對偶變數時所設置的變數個數

5.4 應用網路單形法探討法最大流量問題

Lin(2005) 在其論文中曾提到，包含D節點的最大流量問題求解至最佳解時，其對偶變數可依照值的不同而分類為數群，不似傳統最大流量問題恰可分為對偶變數為0與1的兩群集，該論文僅觀察到此一現象而缺乏相關理論之探討。本節將以3.5節中所得的對偶性質來分析包含D節點的最大流量問題之性質，以解釋Lin(2005)論文中所提及對偶變數分群之現象；除此之外，亦將以網路單形法的角度探討最大流量問題所擁有的最大流量最小割值定理(max-flow-min-cut)。

5.4.1 傳統最大流量問題之對偶性質

傳統最大流量問題若新增一條從終點 t 至起點 s 的弧，並令 $c_{ts} = -1$ 且 $u_{ts} = \infty$ 之後，可轉變為目標式為最小化 $-c_{ts}x_{ts}$ 的MCF，如圖5.15所示。

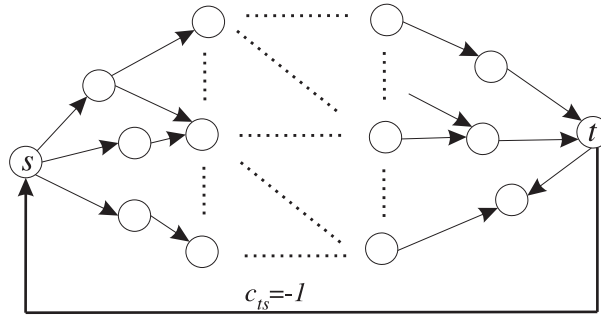


圖 5.15: 傳統最大流量問題可轉換為MCF

一般使用單形法求解傳統最大流量問題時，可從起點利用BFS尋找一展開樹為其初始基解，且基解弧的流量皆為0。由於除 $c_{ts} = -1$ 之外，其他弧之單位成本皆為0，使得其對偶變數 $\pi_i = 0 : \forall i \in N$ ，且只有弧 (t, s) 不滿足其最佳解條件 $\pi_t - \pi_s \leq -1$ ，因此當弧 (t, s) 為進入弧時，其作用就相當於從節點 s 開始推送流量至節點 t ，並使此路徑最先到達上界之弧為退出變數，而完成一次流量的疊代。

當弧 (t, s) 為基弧之後，若再移除弧 (t, s) ，將會使得基解圖所對應的展開樹分割為兩子群集 S 與 \bar{S} ，如圖5.16；如果嘗試計算此基解的對偶變數，並令 $\pi_t = 0$ ，則同屬 S 群的節點其對偶變數值恆為1，而同屬 \bar{S} 的節點其對偶變數值則

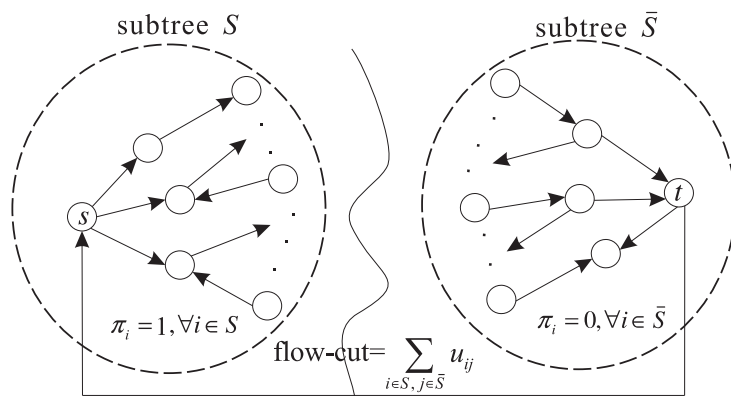


圖 5.16: 傳統最大流量問題其對偶特性

均為0。若再進一步分析，只有 $(i, j) : i \in S, j \in \bar{S}$ 且其流量其流量達下界之弧或是 $(i, j) : i \in \bar{S}, j \in S$ 且其流量達上界之弧會違反對偶可行解條件；且當有流量通過之後，亦會使得 (t, s) 恆為基弧而永不退出基底。Goldfarb and Hao(1990) 即是應用單形法求解最大流量問題所擁有之特性，得以省略原先單形法中需要計算對偶變數的程序，並加入一些具有效率的篩選準則，使單形法能夠在 $O(n^2m)$ 求解出最大流量，其求解過程類似 Ahuja and Orlin(1991) 利用節點標籤所提出的最短增益路徑法 (shortest augmenting path algorithm)。

另外，MCF 可視為 MDCP 去除 D 節點的特例問題，而最大流量問題的節點淨流量值皆為0，且除 c_{ts} 外其餘弧之單位流量成本皆為0；因此由 (3.22) 可將最大流量問題的目標式簡化為：

$$\begin{aligned} & \max \sum_{(i,j) \in U} u_{ij}(\pi_j - \pi_i) \\ \Rightarrow \min & \sum_{(i,j) \in U} u_{ij}(\pi_i - \pi_j) \end{aligned} \quad (5.7)$$

傳統的最大流量問題當求解至最佳解時，可將其對偶變數代入 (5.7)，且此類問題的對偶變數恆為0與1兩種，代表式中 $\pi_i - \pi_j$ 之差值恆為1 (π_i 恆為1而 π_j 恆為0)，因此其對偶問題目標式值可再簡化為「所有達上界的非基變數其容量之總和」。另外，這些流量達上界的非基弧必定屬於 $\{(i, j) : i \in S, j \in \bar{S}\}$ ；而弧 $(i, j) : i \in \bar{S}, j \in S$ 必定為流量達下界之弧；這些特性即與傳統對最大流量最小割

值定理的認知相同，因此最大流量最小割值定理就可視為傳統最大流量問題對偶性質的一種展現。

5.4.2 包含D節點之最大流量問題其對偶性質

Lin(2005)提出包含D節點的最大流量問題求解至最佳解時，可依照對偶變數值的不同而分類為數群；本節則是將此特性利用網路單形法的角度解釋。

包含D節點的最大流量問題若新增一條從終點 t 至起點 s 的弧，並令 $c_{ts} = -1$ 且 $u_{ts} = \infty$ 後，同樣可轉變目標式為最小化 $-c_{ts}x_{ts}$ 的MDCP；但由於MDCP的基解圖並非展開樹，而是由 $n + q - p - 1$ 條弧且包含數個迴圈所構成之圖形，使得當弧 (t, s) 有流量通過為基底時，去除弧 (t, s) 並無法將基解圖形切割成為兩個子集合，而且對偶變數亦有可能出現非0或1的其他值。圖5.17 (a)為一最大流量問題範例，弧中的流量皆為0，節點1為起點，節點7為終點；圖5.17(b)則為該流量所對應的一組基解圖及對偶變數值。由圖中可看出若移除弧 $(7, 1)$ 後圖形無法進一步拆解成兩個子圖形，且圖中的對偶變數可分為4群，除了0與1外，尚有0.25與1.05等。

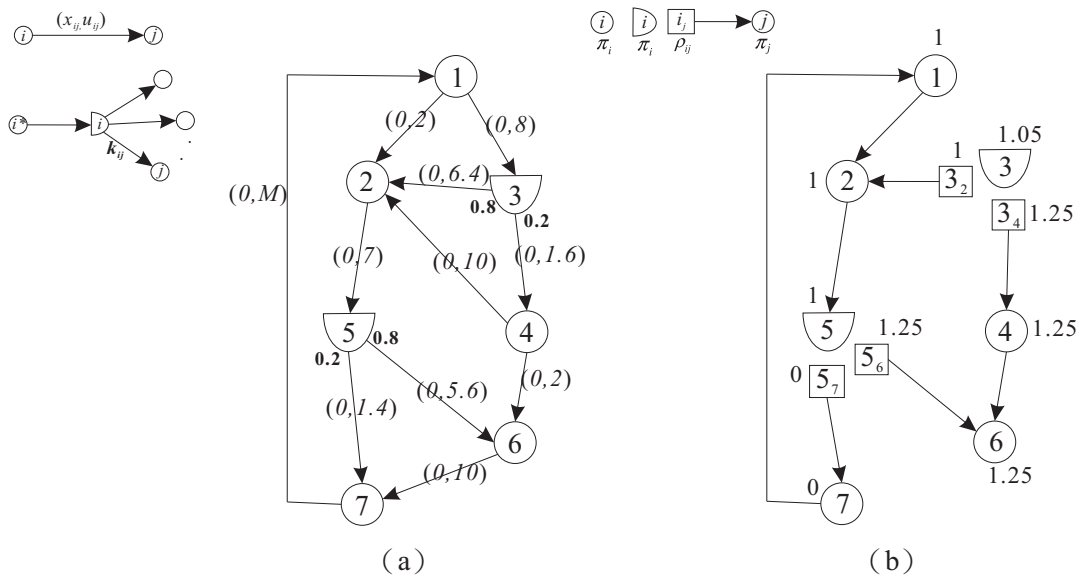


圖 5.17: 包含D節點之最大流量問題範例1

在4.1.1節中我們利用基群集來探討對偶變數的分群，事實上Lin(2005)所觀察到的分群可用基群集的分群來解釋。由於基群集包含 $p + 1$ 群，而且對每一個

群集而言，群集中所有的對偶變數皆能由其內的某一對偶變數值推得，又因為最大流量問題僅 $c_{ts} = -1$ ，而其他 $c_{ij} = 0 : (i, j) \in A \setminus \{(t, s)\}$ ；所以當有流量產生使得 x_{ts} 恆為基變數時，節點 t 與節點 s 所在的基群集包含 0 與 1 兩種對偶變數值，但其他 p 群則可能出現非 0 與 1 的其他值。因此，在加入 D 節點的最大流量問題中，其對偶數值可能出現 $p + 2$ 種數值，如圖 5.17(b) 中，D 節點個數為 2，圖中則最多出現 4 種對偶變數值。由於此項差異，將使得在使用網路單形法求解包含 D 節點的最大流量問題時，無法採用二分法做為非基變數進入的準則，不似傳統最大流量問題可分為 S 與 \bar{S} 兩群集；而需要先計算出每個群集中的對偶變數值之後，再檢查連接群集之間的非基弧是否滿足其最佳解條件才能挑選進入弧並進行下一次的疊代過程。

另外，我們也可針對包含 D 節點的最大流量問題之目標式做進一步簡化。同樣除了弧 (t, s) 外，其他弧成本值皆為 0，而且所有節點淨流量值為 0；因此，(3.22) 可簡化為：

$$\begin{aligned} & \max \sum_{(i,j) \in U \cap (A_{\bar{D}\bar{D}} \cup A_{\bar{D}D})} u_{ij}(\pi_j - \pi_i) + \sum_{(i,j) \in U \cap A_{D\bar{D}}} u_{ij}(\pi_j - \rho_{ij}) \\ \Rightarrow \min & \sum_{(i,j) \in U \cap (A_{\bar{D}\bar{D}} \cup A_{\bar{D}D})} u_{ij}(\pi_i - \pi_j) + \sum_{(i,j) \in U \cap A_{D\bar{D}}} u_{ij}(\rho_{ij} - \pi_j) \end{aligned} \quad (5.8)$$

由 (5.8) 可得出類似 (5.7) 的結果，但由於在包含 D 節點的最大流量問題中，其對偶變數值不單只為 0 與 1 兩種，因此並無法直接簡化成以往對最大流量問題所認知的最大流量最小割值定理。圖 5.18(a) 與 (b) 為一包含 D 節點的最大流量問題之最佳解與其所對應的基解圖， x_{12} 與 x_{58} 為到達上界的非基變數，在此範例中已達最大流量 3.5，此值亦為 $u_{12}(\pi_1 - \pi_2) + u_{58}(\pi_5 - \pi_8) = 1 + 1.25 \times 2 = 3.5$ 。

雖然在包含 D 節點的最大流量問題中，其非基弧兩端點的對偶變數之差值不為 1，導致無法直接推導成傳統最大流量問題所擁有的最大流量最小割值定理；但是對偶變數分群的現象顯示出此類最大流量問題仍有其特殊之處，並與本研究提出的基群集相符合。因此，往後若能進一步探討基群集對於最大流量問題的影響，並適時的修正或放寬割值的定義，或許可提出一種適用於此類最大流量問題的改良式「最大流量最小割值定理」。

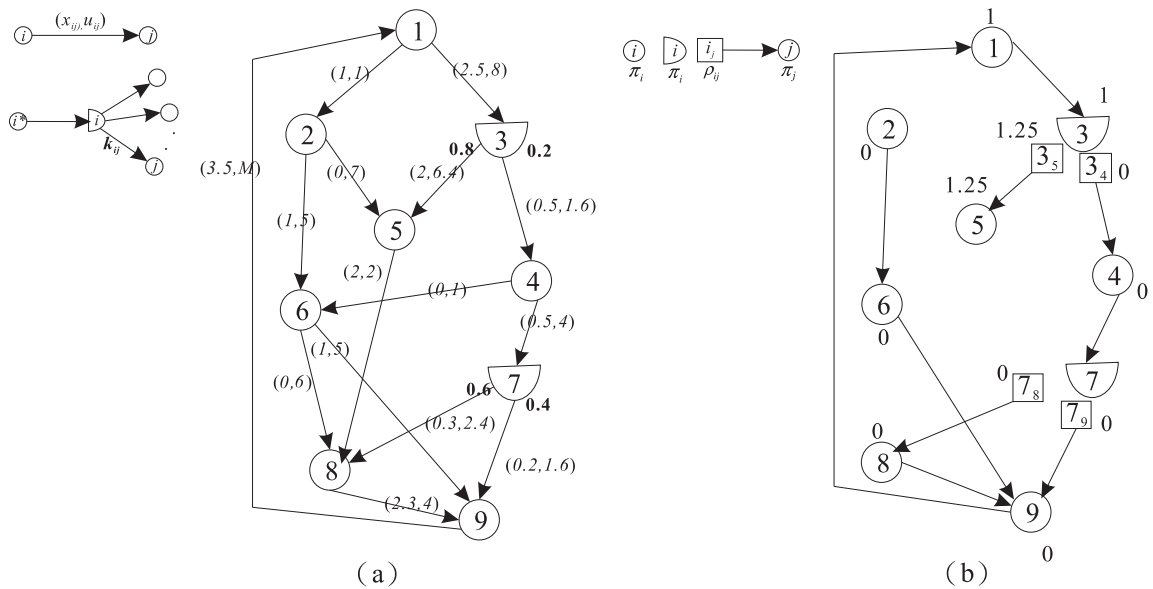


圖 5.18: 包含D節點之最大流量問題範例2

5.5 小結

本章探討其他與MDCP相關的四項議題，第一項議題主要說明了Fang and Qi(2003)對於樞紐圖的錯誤判讀的原因，解釋dead-cycle並不存在；第二項議題則提供另外一種模式轉換的方式。而在第三項議題中，藉由探討Ting(2005)所提出的compatible component於樞紐圖中的分群情形，我們可發現本研究所提出的網路單形法不論在流量疊代或是對偶變數更新等程序，尚可利用進一步的代換，或是增加一些變數的設立準則使得所設置的變數盡量減少，以進一步縮小最後所需求解的聯立方程組。最後，我們利用網路單形法探討包含D節點的最大流量問題，並可利用基群集解釋其分群現象，說明對偶變數最多可分群為 $p+2$ 群；而將來若要分析此類的最大流量問題是否有特殊之演算法，或是提出適用於此類最大流量問題的「最大流量最小割值定理」時，可用本研究所提出的基群集為基礎，以做更進一步的探討。

第六章

結論與未來發展方向

6.1 結論

本研究的主要目的乃在於發展一套圖形操作且系統化的網路單形法，使之能以求解具容量上限的MDCP。首先，為了能將MDCP模式套用於具容量限制之網路，並確保基解圖的相連性，進而簡化問題之特性，本研究在第三章中將模式做適當的轉換，使其在流量守恆限制式矩陣的架構上與傳統MCF相同；除此之外，面對具有多個S節點與多T節點的MDCP時，也可利用類似之程序轉換為一對多的MDCP，而使得本研究所探討的一對多MDCP具有相當的代表性。經由模式轉換，我們可以確保圖形的相連性，並除去Fang and Qi(2003)所做的弧容量不設限、網路中成本非負以及需求下限嚴格限制為正等諸多的假設條件；而藉由探討MDCP對偶問題之性質，我們可用較簡單之方式陳述每條弧所對應的對偶可行解條件，並依此做為網路單形法的最佳解條件。

在第四章中，導入基群集之概念，將基解圖拆解為 $p+1$ 群，並能夠引用此分群探討其對偶變數之計算、流量疊代、對偶變數更新以及初始解之設立等程序，建立一套有系統，且以圖形操作為基礎的網路單形法。第五章探討四項與MDCP相關之議題，其中除了糾正Fang and Qi(2003)文獻中對於樞紐圖的判斷錯誤與提出另一種轉換模式的方式之外，我們亦提出可增加本研究的網路單形法之實用效率的法則，以再進一步減少最後所需求解方程組的大小。最後，當我們利用網路單形法分析包含D節點的最大流量問題時，可以解釋Lin(2005)在其論文中所提及的對偶變數分群之現象，並可推導出對偶變數值最多可分為 $p+2$ 群；而

將來如欲提出一套適用於包含D節點的最大流量問題的「最大流量最小割值定理」，或許可從本研究的基群集著手並加以擴充相關之定義。

6.2 未來研究方向

雖然本研究已提出網路單形法整體之架構，但仍有幾個方向可供未來探討與分析：

1. anti-cycling：由於網路單形法為一種特殊的單形法，因此單形法所遇到的衰退解(degenerate solution)亦會在以網路單形法求解的過程中出現。然而，本研究尚未提供一套圖形化之準則，以避免因為衰退解而會出現的cycling現象。傳統以網路單形法求解MCF時，可使用strongly feasible spanning tree之概念以避免發生cycling(Ahuja et al., 1993)。在strongly feasible spanning tree中，每條流量達下界之弧其頭端必維持指向根節點，而每條流量達上界之弧其頭端必維持指離根節點；當退出弧有多重選擇時，只要沿推送流量方向選擇距離進入弧最遠之弧為退出弧，即可在每一個iteration均維持此特殊架構而能夠避免發生cycling。圖6.1(a)即為一個strongly feasible spanning tree之範例，弧(4, 2)、(7, 5)因其流量達下界，故其方向向上指向根節點1；而弧(2, 3)、(6, 8)之流量已達上界，故其方向向下指離根節點1。假如(9, 10)為流量達下界的進入弧，(6, 8)與(7, 5)均符合退出弧之條件，若沿推送流量之方向選擇最遠之弧(7, 5)為退出弧即可在下一個iteration繼續維持此架構。

在本研究所提出的網路單形法中，我們拆解造成迴圈的D節點而將基解圖拆解為 $p + 1$ 群，並以群集為單位來探討網路單形法中的每一步驟。本研究亦曾嘗試採行使每一群集內的樹各自維持strongly feasible spanning tree之架構，來探討是否能有anti-cycling等的特殊準則，然而我們發現此strongly feasible spanning tree的特殊架構在進入弧合併兩群集之樹時，將難以繼續維持；除此之外，由於MDCP的樞紐圖為一個包含許多迴圈的複雜圖形，所以無法如同在求解MCF那般可以直接沿推送方向選擇距離最遠之弧為退出

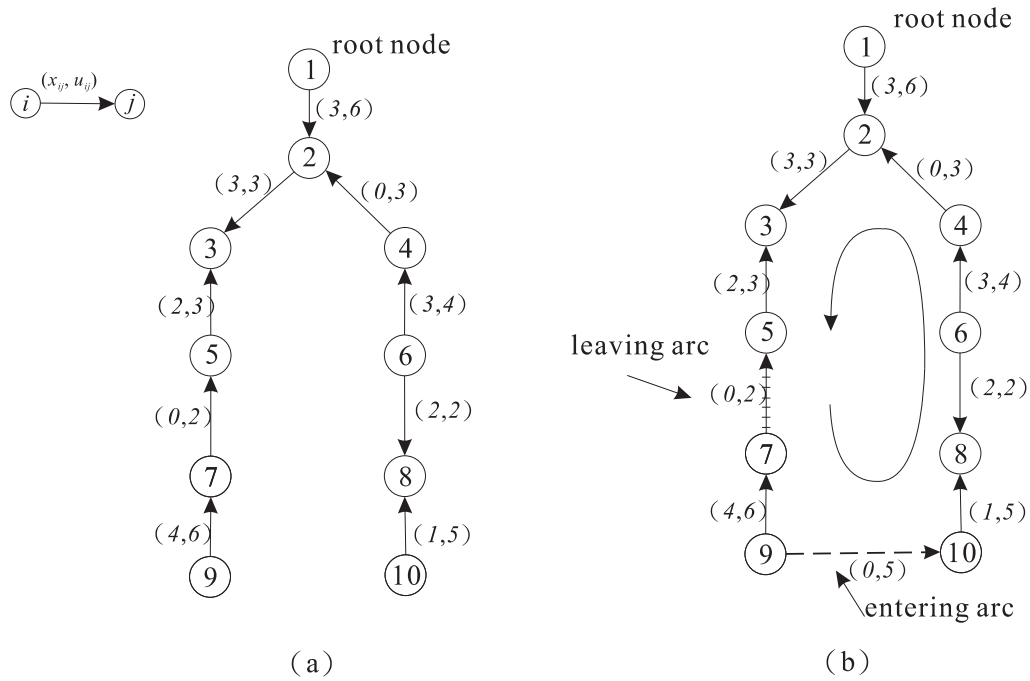


圖 6.1: strongly feasible spanning tree

弧。雖然如此，但以往MCF所對應的strongly feasible spanning tree，其實可視為單形法中為避免發生cycling所採用的lexicography法則之應用；因此，如何以lexicography法則結合網路單形法，設計出可避免發生cycling的特殊圖形架構，可能是未來可以嘗試的一個研究方向。

2. 資料結構與演算法實際效率：本研究所提出之演算程序尚未提及資料結構相關之設計，如何針對MDCP設計出一套可供操作且有效率的資料結構，如何有效率地實作本研究所提出的圖形式網路單形法，以及其實作之效率驗證等等議題，皆為未來非常具有挑戰性的研究方向。

參考文獻

- Ahuja, R. K. and Orlin, J. B. Distance-directed augmenting path algorithms for maximum flow and parametric maximum flow problems. *Naval Research Logistics Quarterly*, **38**, 413–430, 1991.
- Ahuja, R. K., Goldberg, A. V., Orlin, J. B. and Tarjan, R. E. Finding minimum-cost flows by double scaling. *Mathematical Programming*, **53**, 243–266, 1992.
- Ahuja, R. K., Magnanti, T. and Orlin, J. B. *Network flows: Theory, algorithms and applications*. Englewood Cliffs, New Jersey, U.S.A.: Prentice Hall, 1993.
- Ali, A. I., Padman, R. and Thiagarajan, H. Dual algorithms for pure network problems. *Operations Research*, **37**(1), 159–171, 1989.
- Armstrong, R. D., Chen, W., Goldfarb, D. and Jin, Z. Strongly polynomial dual simplex methods for the maximum flow problem. *Mathematical Programming*, **80**, 17–33, 1998.
- Armstrong, R. D. and Jin, Z. A new strongly polynomial dual network simplex algorithm. *Mathematical Programming*, **78**, 131–148, 1997.
- Bellman, R. E. On a routing problem. *Quarterly of Applied Mathematics*, **16**, 87–90, 1958.
- Cohen, E. and Megiddo, N. Algorithms and complexity analysis for some flow problems. *Algorithmica*, **11**(3), 320–340, 1994.

- Dantzig, G. Application of the simplex method to a transportation problem. In *Activity analysis of production and allocation*. New York, N. Y.: John Wiley & Sons Inc., 359–373, 1951.
- Dantzig, G. B. *Linear programming and extensions*. Princeton, N.J.: Princeton University Press, 1962.
- Dial, R. Algorithm 360: Shortest path forest with topological ordering. *Communications of the ACM*, **12**(11), 632–633, 1969.
- Dijkstra, E. W. A note on two problems in connexion with graphs. *Numerische Mathematik*, **1**, 269–271, 1959.
- Fang, S. C. and Qi, L. Manufacturing network flows: A generalized network flow model for manufacturing process modeling. *Optimization Methods and Software*, **18**, 143–165, 2003.
- Ford, L. R. *Network flow theory*. Santa Monica, California: The RAND Corp., 1956.
- Ford, L. R. and Fulkerson, D. R. Maximal flow through a network. *Canadian Journal of Mathematics*, **8**, 399–404, 1956.
- Ford, L. R. and Fulkerson, D. R. *Flows in networks*. Princeton, N.J.: Princeton University Press, 1962.
- Fredman, M. L. and Tarjan, R. E. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM*, **34**(3), 596–615, 1987.
- Goldberg, A. V. and Tarjan, R. E. A new approach to the maximum flow problem. *Journal of the ACM*, **35**, 921–940, 1988.
- Goldfarb, D. and Chen, W. On strongly polynomial dual simplex algorithms for the maximum flow problem. *Mathematical Programming*, **80**, 159–168, 1997.

- Goldfarb, D. and Hao, J. A primal simplex algorithm that solves the maximum flow problem in at most nm pivots and $O(n^2m)$ time. *Mathematical Programming*, **47**, 353–365, 1990.
- Goldfarb, D. and Jin, Z. An $O(nm)$ -time network simplex algorithm for the shortest path problem. *Operations Research*, **47**(3), 445–448, 1999.
- Hoffman, A. J. and Kruskal, J. B. Integral boundary points of convex polyhedra. In H. Kuhn and A. Tucker (Eds.), *Linear inequalities and related systems* (pp. 233–246). Princeton, NJ: Princeton University Press, 1956.
- Johnson, E. L. On shortest paths and sorting. In *Proceedings of the ACM 25th annual conference*, 510–517, 1972.
- Karzanov, A. V. Determining the maximal flow in a network by the method of preflows. *Soviet Math. Dokl*, **15**, 434–437, 1974.
- Lin, J. C. Maximum flows in distribution networks: problem generator and algorithms. *Master's thesis, National Cheng Kung University*, 2005.
- Mo, J., Qi, L. and Wei, Z. A network simplex algorithm for simple manufacturing network model. *Journal of Industrial and Management Optimization*, **1**(2), 251–273, 2005.
- Orlin, J. B. A polynomial time primal network simplex algorithm for minimum cost flows. *Mathematical Programming*, **78**, 109–129, 1997.
- Orlin, J. B., Plotkin, S. A. and Tardos, E. Polynomial dual network simplex algorithms. *Mathematical Programming*, **60**, 255–276, 1993.
- Ting, M. J. Maximum flow problem in the distribution network flow model. *Master's thesis, National Cheng Kung University*, 2005.

Yang, Y. H. Uncapacitated minimum cost flow problem in a distribution network.
Master's thesis, National Cheng Kung University, 2005.