國立交通大學

工 業 工 程 與 管 理 學 系

碩士論文

有限資源專案排程之巨集啟發式演算法的比較

A Comparison of Meta-heuristics Algorithms for Resource-Constrained Project Scheduling

研究生：施富騰

指導教授：巫木誠 博士

中華民國九十六年六月

有限資源專案排程之巨集啟發式演算法的比較

A Comparison of Meta-heuristics Algorithms for Resource-Constrained Project Scheduling

研 究 生：施富騰　　Student：Fu-Teng Shi

指導教授：巫木誠 博士　　Advisor：Dr. Muh-Cherng Wu

國立交通大學

工業工程與管理學系

碩士論文

A Thesis

Submitted to Department of Industrial Engineering and Management

College of Management

National Chiao Tung University

In Partial Fulfillment of the Requirements

For the Degree of Master of Science

In

Industrial Engineering

June 2007

Hsin-Chu, Taiwan, Republic of China

中 華 民 國 九 十 六 年 六 月

有限資源專案排程之巨集啟發式演算法的比較

研究生：施富騰　　指導教授：巫木誠博士

國立交通大學工業工程與管理研究所

中文摘要

有限資源的專案排程問題(Resource-Constrained Project Scheduling，RCPSP)具有複雜的求解特性。過去已有許多學者提出不同方法來求解，其中 Debels et al. (2006)演算法是目前相對較佳的方法。本研究結合共識因子和田口方法提出多種巨集演算法(meta-heuristics)，希望找出一種演算法，能在績效上改進 Debels et al. (2006)的演算法。本研究使用 1560 個案例，分成 9 種情境進行實驗。與 Debels et al. (2006)演算法相比，本研究所發展的演算法僅在 22%的情境表現較佳，在另 22%的情境績效相近，卻在 56%的情境中表現較不如過去的演算法。

關鍵辭：專案排程、資源限制、共識因子、田口方法

A Comparison of Meta-heuristics Algorithms for Resource-Constrained Project Scheduling

Student：Fu-Teng Shi　　Advisor：Dr. Muh-Cherng Wu

Institute of Industrial Engineering National Chiao Tung University

ABSTRACT

Resource-constrained project scheduling is a NP-hard problem. In the last few decades, several meta-heuristics algorithms have been proposed to solve the problem. The algorithm proposed by Debels et al. (2006) is by far the most leading one. To develop a better algorithm, we applied the notions of consensus and Taguchi genetic operators and proposed various meta-heuristics algorithms. Extensive numerical tests have been carried out. These tests include 1560 problem instances, which are categorized into 9 scenarios. Compared with the algorithm proposed by Debels et al. (2006), our algorithm excel in 22% scenarios, has a tie in 22% scenarios, and lose in 56% scenarios.

Keywords：Project scheduling、Resource-Constrained、Consensus、Taguchi methods

誌謝

本論文得以順利完成，首先要感謝恩師 巫木誠教授的悉心指導，使我在研究的過程中，對於解決問題的能力以及做人處事的方法和態度，都得到了很大的啟發，在此致上最崇高的敬意。同時感謝許錫美教授、彭德保教授和陳文智教授在論文口試時，於研究方向和研究內容給予的寶貴建議，讓本論文更臻完備。

感謝學長吳政翰在工作期間仍熱心為我解答許多問題，學長陳德珊和盧威豪學長的經驗分享，為我解決了不少研究碰到的困難，以及柏儒撥空幫助我研究的進行。博士班學長蘇泰盛、施昌甫與林珊慧，謝謝你們的經驗傳承與陪伴。好友進銘、振富、忠霖、光棓、秉琦、艾苓、廷勳以及研究室的同學和學弟妹們，不論是在課業上的學習或者生活的陪伴，都因為你們使我體會到成長的喜悅。

最後，要深深感謝我的父母、兄弟，由於你們不斷給我支持和鼓勵，讓我能無後顧之憂的完成學業，也必須感謝我的女友，常在我無助、沮喪的時候，不時的關心我、鼓勵我。你們使我能有更堅定的信心來完成論文，更永遠是我最大的支持與驕傲，希望能將我此刻的喜悅和一切與你們分享，謝謝你們。

富騰 于風城交大

中華民國九十六年六月

目錄

表目錄

圖目錄

第一章　緒論

1.1 研究問題

參考 Herroelen et al.(1998b)和 Debels et al.(2006)的定義，將有限資源的專案排程問題(Resource-Constrained Project Scheduling Problem，RCPSP)描述如下：

一個專案有 1 到 n 個作業(activity)，作業 A_i $(1 \leqq i \leqq n)$之間的相依關係可以描述為一個有向非循環圖 $G(V, E)$，如圖 1.1，V 表示所有作業 A_i (vertex)的集合，E 代表專案圖形的邊(edge)或弧(arc)的集合，表示 2 個 A_i 之間有直接順序關係(precedence relation without time lag)，d_i 為 A_i 的執行時間，執行期間不可中斷。此專案有 K 種可重覆使用的資源，a_k $(1 \leqq k \leqq K)$為資源 k 的數量限制；r_{ik} 代表 A_i 對資源 k 的需求量，專案執行期間必須滿足 $r_{ik} \leqq a_k$ 的限制。A_1 和作業 A_n 為空節點，其執行時間為 0 並且沒有資源需求。

一個合理(feasible)的專案排程結果以 S 表示，必須同時滿足相依關係和資源需求的限制。排程 S 由 A_i 的開始時間數列 $\boldsymbol{s_i}$ $(1 \leqq i \leqq n)$所組成，並且對應到結束時間數列 $\boldsymbol{f_i}$ $(f_i = s_i + d_i)$。一個 RCPSP 的目標為總完工時間為最小$(\min m_s = \max f_n)$的合理排程。

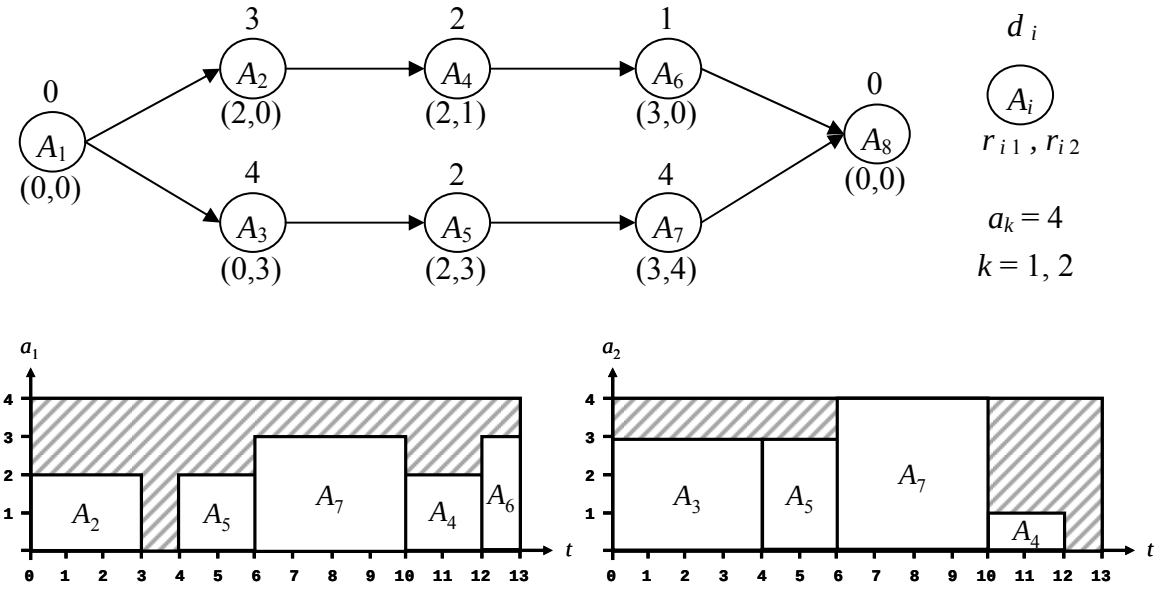


圖 1.1 專案排程問題

由此可知，RCPSP 之所以構成複雜的問題，在於作業間有相依關係的限制(precedence constraint)，同時因為資源有限(resource constraint)，使得作業執行時需考慮資源分派問題。如圖 1.1 的排序原為 A_1→A_2→A_3→A_5→A_7→A_4→A_6→A_8，若將 A_7 和 A_4 的排序調換後，可以得到圖 1.2 的結果，總完工時間可以從 13 減少為 11。由此可知，好的排序是影響總完工時間的關鍵。

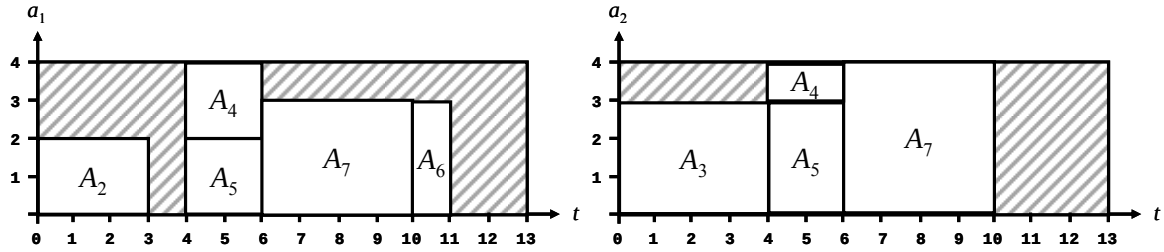


圖 1.2 改變 A_7 和 A_4 的執行順序

1.2 研究假設

(1) 單一專案問題，Herroelen et al.(1998a)將其分類為 *m*, 1|*cpm*|*C*max。

(2) 作業有執行順序限制，執行期間不可中斷。

(3) 資源合作：一個作業可能同時需要 2 種以上的資源才能完成。

(4) 多種資源，每一種資源都有數量限制

(5) 每一個資源可被重覆使用(renewable resource)。

(6) 資源共乘：一種資源在同一個時間點可服務多個作業。

(7) 目標：最小化總完工時間(minimize makespan)。

1.3 研究目的

RCPSP 過去已有不少文獻發表，但碰到複雜的情境時，求解空間變得相當廣大，因此在求解品質上，過去的方法仍有改善的空間。所以本研究首次應用共識因子和田口方法，和過去的標竿演算法(Debels et al., 2006)做各種可能的組合並且跟 Debels et al. (2006)及過去其他文獻進行實驗比較，確認是否可以發展出較佳的演算法，並且分析共識因子及田口方法的優點和缺點提出改善建議。

1.4 章節安排

本論文其他章節安排如下：第二章為文獻回顧；第三章說明本研究方法，包括如何將演算法組合與實驗方法的說明，以及介紹解的表達法和演算法各模組的內容；第四章說明實驗數據及分析實驗結果；第五章是結論與後續改善方向。

第二章　文獻回顧

本章討論過去學者對 RCPSP 的整理分類，以及曾針對 RCPSP 發表過的求解方法，接著回顧本研究比較的標竿演算法與求解方法的文獻，最後介紹本研究所使用的 RCPSP 測試集(test dataset)。

2.1 專案排程問題

專案排程問題(project scheduling problem)的種類相當廣泛，從廣義的分類法來看，過去 Herroelen (2005)，從專案規劃方法的變異程度(variability)和相依程度(dependency)做分類，RCPSP 是屬於低變異程度(deterministic)和低相依程度(single project)的專案問題。

也有其他學者針對 RCPSP 的相關議題做討論。Herroelen et al. (1998b)在文中首先對 RCPSP 做了回顧性的探討，接著分別對 PRCPSP(preemption RCPSP)、考慮淨現值的 RCPSP、…等延伸議題做求解方法和實驗結果上的討論；Brucker et al. (1999)同樣的針對 RCPSP、MRCPSP(multi-mode RCPSP)等各種不同的有限資源專案問題，對問題做分類、定義，並且分析求解方法和實驗結果。

2.2 RCPSP 求解方法

近十年來，已有相當多的文獻研究 RCPSP 的求解方法，像是早期以數學模型來求解的方法有整數規劃(Integer Programming)、動態規劃(Dynamic Programming)以及分支界線法(Branch-and-Bound procedure)。過去以分支界線法來求解的文獻有 Demeulemeester & Herroelen (1992, 1997)、Brucker et al. (1998)、Mingozzi et al. (1998)和 Sprecher (2000)，雖然分支界線法能在中小問題求得最佳解(exact solution approaches)，但隨著專案問題的複雜度增加，求解空間也以指數成長，分支界線法便無法在短時間內求得最佳解。

所以便開始有學者發展啟發式解法(meta-heuristic)，期望能在較短時間內求得近似最佳解。Cho & Kim (1997)提出模擬退火法求解 RCPSP，Zhang et al. (2006)則是將粒子群最佳化(particle swarm optimization)應用在 RCPSP 上，並且自行建立一個簡單的專案問題，跟傳統的基因演算法比較求解績效。

從回顧性文獻來看，Hartmann & Kolisch (2000)將當時各種啟發式解法加以整理分類，並且列出各個方法在 PSPLIB 測試集的求解績效，而最好的方法是 Boulimen & Lecocoq (1998)以模擬退火法和 Hartmann (1998)使用基因演算法來求解。接著在 Kolisch & Hartmann (2006)提出的文獻中，同樣將近年來新的求解方法做整理，並且指出各類方法的特性，以及過去有哪些文獻是使用這些方法的，最後在實驗結果的部分，有 6 個新的方法的求解績效優於 Boulimen & Lecocoq (1998)和 Hartmann (1998)，包括：Hartmann (2002)提出新型的基因演算法；Kochetov & Stolyar (2003)結合基因演算法和禁忌搜尋法；Valls et al. (2003a)提出新型的基因演算法；Alcaraz et al. (2004)將 Alcaraz & Maroto (2001)的基因演算法改良；Valls et al. (2005)提出新的技術來改良過去的方法；Debels et al. (2006)結合分散搜尋法及模擬電磁演算法來求解。其中以 Kochetov & Stolyar (2003)、Valls et al. (2003a)、Alcaraz et al. (2004)和 Debels et al. (2006) 4 種方法績效最好。而 Debels et al. (2006)提出的分散搜尋法結合模擬電磁演算法的架構，能有效在短時間內得到好的求解品質，並且跟該文獻的其他方法比較也有很好的績效。

2.3 PSPLIB 測試集

為了驗證演算法是否有足夠的能力求解 RCPSP，會使用各種測試集(test data set)來試驗。像是 Cho & Kim (1997)以 Patterson 所建構的 110 個專案問題做為測試集，最後有 105 個問題都能求得最佳解。

本研究使用的測試集 PSPLIB，全名為 project scheduling library，是由 Kolisch & Sprecher (1996)建置而成，其建置的方式使用 Kolisch et al. (1995)所提出的“ProGen”產生器，可以藉由參數的調整來產生出複雜度相異的專案問題。

PSPLIB 只適用於測試 RCPSP 和 MRCPSP 兩種專案問題。測試集共有 4 種主要情境，以任務(job)數目做劃分，分別有 J30、J60、J90 和 J120。其中前 3 種情境內各有 480 個例子， J120 的情境中有 600 個例子，而本研究只比較 J30、J60 和 J120 情境(Kolisch & Hartmann, 2006)。

其中每一個例子裡都包含有任務相依關係及資源需求限制等資訊。由於每篇文獻的演算法是在不同的電腦上測試，為了有一致的比較基準，PSPLIB 提供了 3 種停止條件，分別是演化過程產生 1,000、5,000 或 50,000 個解就停止，並且在相同的停止條件下比較求解品質。

跟其他演算法比較求解品質時，本研究以 3 種情境在 3 種不同的停止條件下合計所有例子的總完工時間 (sum of makespan)來比較，並且在附錄附上各種求解組合跟 PSPLIB (http://129.187.106.231/psplib/main.html)提供的最佳解的比較結果：J30 是以最佳解(best solutions)為比較基準，J60 和 J120 因為還沒有真正的最佳解，所以同樣用 PSPLIB 提供目前能找到的最佳解(currently best solutions)來比較。最佳解的公佈時間分別是：J30 為 1997 年 8 月 4 日公佈，J60 是 2007 年 6 月 9 日更新，J120 則是 2007 年 6 月 8 日更新。

第三章　研究方法

本由於研究目的是將各種求解方法組合，以期能得到較佳的結果，所以這章介紹求解方法。首先以架構圖(generic framework)介紹整個的演算法組合有哪些，並且說明如何進行實驗；接著介紹各種演算法模組的內容。茲分別介紹如下：

3.1 研究架構與實驗方法

研究架構的部分，以 Debels et al. (2006)的演算法架構為基礎，另外再加入共識因子(consensus operator)和田口方法(Taguchi methods)，成為產生更新解的方法，架構如圖 3.1。

本架構主要分成 5 個模組：初始解產生流程、母體、更新解產生器、更新解修正器以及彙總和母體更新器，其中彙總和母體更新器視為同一個模組。前 4 個模組上面代表著該模組建構或使用的方法，依序給予不同的編號，並且可以經由不同方法組合成各種結構相異的演算法。本架構主要的變化在「更新解產生器」這個模組，其他的模組都只使用一種方法。

更新解產生器中，C1、C2 為 Debels et al. (2006)所提出的方法，C3 至 C15 則為本研究所提出的方法。共識因子由 3×4 個參數所組成，參數組合內容如表 3.1。"Weighting"分成 No weighing、Linear weighting 和 Power weighting 3 種；"Rate (α)"則有 1、0.8、0.8 combined 1.2 和 0.8 and 1.2 等 4 種。有關以上 5 個模組的說明和相關參數設定，一併在 3.2 節完整介紹。

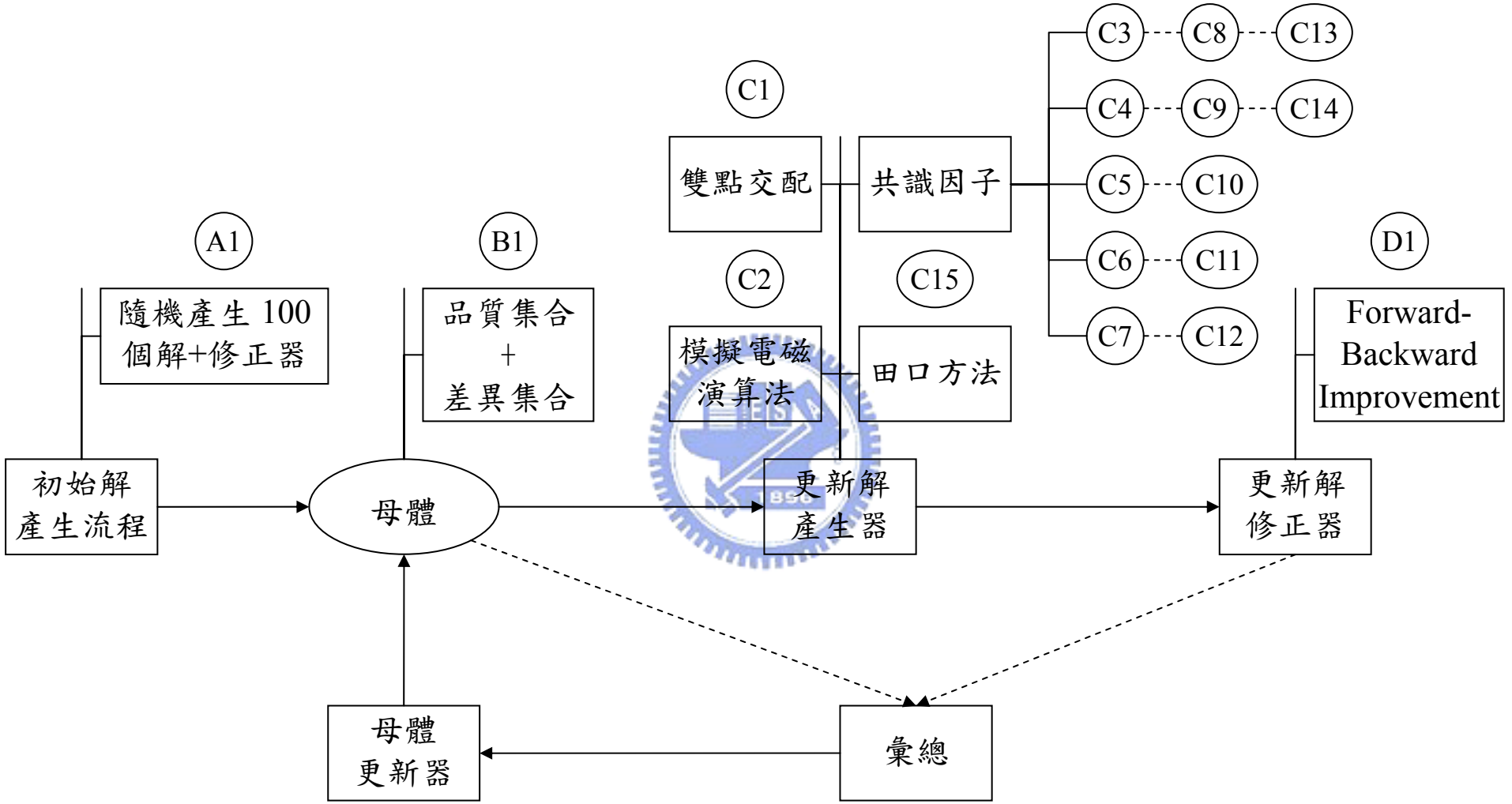

C1
C3
C8
C13
C4
C9
C14
C5
C10
C6
C11
C7
C12
雙點交配
共識因子
A1
B1
C2
C15
D1
隨機產生 100
個解+修正器
品質集合
+
差異集合
模擬電磁
演算法
田口方法
Forward-
Backward
Improvement
初始解
產生流程
母體
更新解
產生器
更新解
修正器
母體
更新器
彙總

圖 3.1 演算法架構圖

經由對圖 3.1 的架構介紹之後，如果將更新解產生器以兩兩配對的方式，可以將演算法整理出 3 + (3 * 4) = 39 種版本(version)如表 3.2，其中 V0 這個版本是 Debels et al. (2006)的演算法，為了比較其他版本是否能優於 V0，所以基本上必須對 V1 到 V38 等 38 個版本進行實驗。然而，由於本問題的 J30、J60 和 J120 這 3 種情境總共有 480 + 480 + 600 = 1560 個例子，而每個例子要跑 10 次實驗(run)取平均才能得到一個例子的資料，所以一個版本要跑 15,600 個例子的實驗。若將所有實驗完整的跑完將會花上許多時間，並且在實驗過程中常在沒有完全跑完實驗時，就可以取得相關資訊進一步對實驗方法做調整。為此，本研究將實驗分成導引實驗(pilot study)和綜合實驗(comprehensive study)分別介紹如下。

表 3.1 共識因子參數組合

組合編號	Weighting	Rate (α)
C3	No	1
C4	No	0.8
C5	No	0.8 combined 1.2
C6	No	0.8 and 1.2
C7	Linear	1
C8	Linear	0.8
C9	Linear	0.8 combined 1.2
C10	Linear	0.8 and 1.2
C11	Power	1
C12	Power	0.8
C13	Power	0.8 combined 1.2
C14	Power	0.8 and 1.2

表 3.2 演算法組合版本

初始解	母體	更新解產生器(3 + 3 * 12 組合)		更新解修正器
A1	B1	**V0**：C1 + C2	**V20**：C2 + C9	D1
		V1：C1 + C3	**V21**：C2 + C10	
		V2：C1 + C4	**V22**：C2 + C11	
		V3：C1 + C5	**V23**：C2 + C12	
		V4：C1 + C6	**V24**：C2 + C13	
		V5：C1 + C7	**V25**：C2 + C14	
		V6：C1 + C8	**V26**：C2 + C15	
		V7：C1 + C9	**V27**：C3 + C15	
		V8：C1 + C10	**V28**：C4 + C15	
		V9：C1 + C11	**V29**：C5 + C15	
		V10：C1 + C12	**V30**：C6 + C15	
		V11：C1 + C13	**V31**：C7 + C15	
		V12：C1 + C14	**V32**：C8 + C15	
		V13：C1 + C15	**V33**：C9 + C15	
		V14：C2 + C3	**V34**：C10 + C15	
		V15：C2 + C4	**V35**：C11 + C15	
		V16：C2 + C5	**V36**：C12 + C15	
		V17：C2 + C6	**V37**：C13 + C15	
		V18：C2 + C7	**V38**：C14 + C15	
		V19：C2 + C8		

3.1.1 導引實驗(pilot study)

所謂導引實驗，就是取出部分的實驗項目進行實驗並且加以分析，決定剩餘的實驗項目是否有需要繼續完成，以及後續實驗內容需要改善的部分。在導引實驗中，本研究只選出 J30 和 J60 的問題來實驗，並且只以 1,000 和 5,000 條解(schedule)做為停止條件。

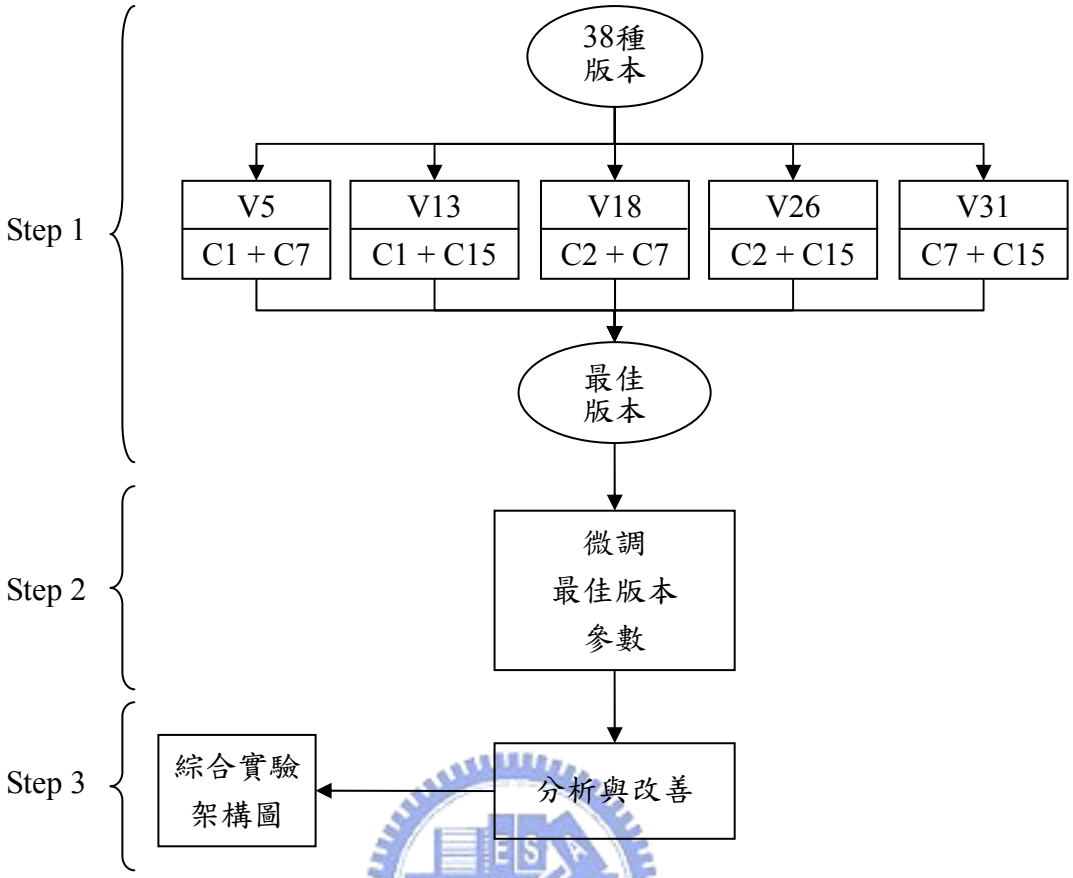


圖 3.2 導引實驗流程

然而要進行導引實驗，也必須決定各種版本的實驗次序，因為共識因子的參數是構成這許多版本的主要原因，並且通常參數的微調是求參數的最佳化，彼此之間的結果不會相差太多，所以在此先將共識因子選出一種參數組合(C7)跟 C1、C2 和 C15 等方法搭配，選 C7 的原因在於 C7 的 $\alpha = 1$ 是共識因子(吳政翰, 2006)的原始參數，因為吳政翰 (2006)沒有使用權重(weighting)的概念，本研究認為加入權重會比原本好，因而選擇 C7。

如圖 3.2 所示，從 38 種版本選出 5 個版本來做實驗，找出最佳組合之後，先由實驗結果考慮最佳版本的參數微調方式，接著分析上述所有實驗的結果，對圖 3.1 的整個實驗架構和內容做改善，最終整理出綜合實驗架構圖(comprehensive study framework)進行下一步的綜合實驗。

3.1.2 綜合實驗(comprehensive study)

綜合實驗是根據導引實驗結果整理的架構圖來進行的，在綜合實驗架構所組合出來的演算法，必須跑完 J30、J60 和 J120 分別在 1,000、5,000 和 50,000 等 3 種停止條件下的實驗。綜合實驗流程分成以下 4 個步驟：

Step 1：與 V0 比較實驗結果，選出最佳版本。

Step 2：將最佳版本跟其他過去文獻比較結果。

Step 3：分析最佳版本的優劣點。

Step 4：改善最佳版本。

3.2 演算法模組介紹

3.1 節介紹了本研究架構和實驗方法，本節介紹研究架構中各模組的內容，包括初始解產生流程、母體、更新解產生器、更新解修正器以及參數設定。

3.2.1 解的表達法

每條解表達了一個專案排程 S 裡，各項作業的排名，可表示為 $X = [v_1; v_2; \ldots; v_n]$，$n$ 代表所有作業的個數，$v_i (1 \leqq i \leqq n)$表示作業 A_i 的排名(rank value)，而一個有效的排程(valid schedule)不可以違反專案的執行順序限制。

例如在一個 $G(V, E)$中可能產生如圖 3.3 的排程，根據各項作業執行的開始時間，由前到後各指定一個排名，就可以得到表 3.3 的染色體。其中 A_1 和 A_{11} 為空節點，所以不會出現在圖 3.3。

表 3.3 有效的排程結果(SRK)

作業	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}
排名	1	2	4	5	7	2	8	9	5	10	11

由此可知，一個有效的排程必須滿足相依關係的限制(precedence constraint)，同時也能表達專案中各項作業的排序(sequence)，進而求得總完工時間。這個表達法由 Debels et al. (2006)提出，稱為 Standardize random Key，SRK。為了方便敘述，本文統一將合理的專案排程稱為「SRK」。

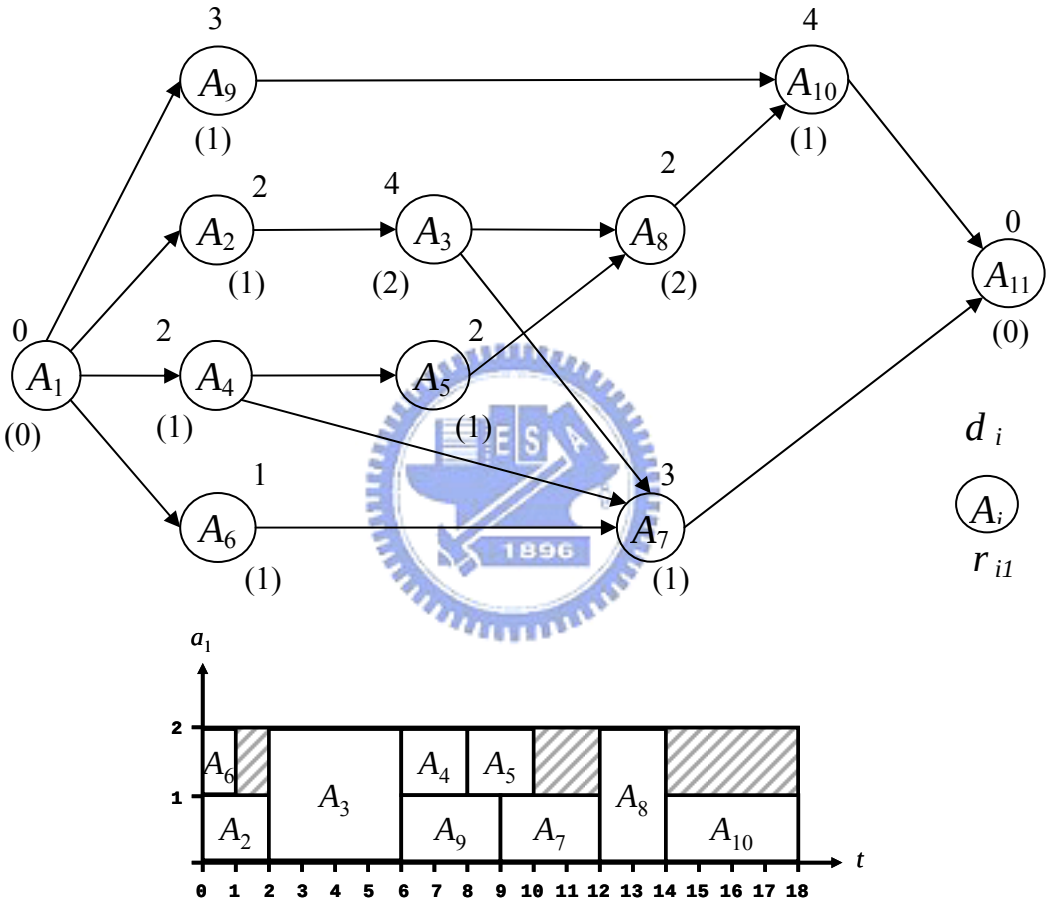


圖 3.3 RCPSP 排程釋例

3.2.2 初始解產生流程

如圖 3.4，初始解產生流程共分為三個步驟：首先，由 SRK 產生器產生 N_i : 100 條初始解放入初始解集合 S_i；接著使用初始解修正器改善；最後，從 S_i 中選出品質最好的解形成母體 $P(t)$。

由於母體是以分散搜尋法的方法建構，結構和產生方法不同於一般，因此放到 3.2.3 介紹；另外，初始解修正器和更新解修正器所使用的方法相同，因此一併留到 3.2.4 節介紹。接著介紹 SRK 產生器如何產生出染色體。

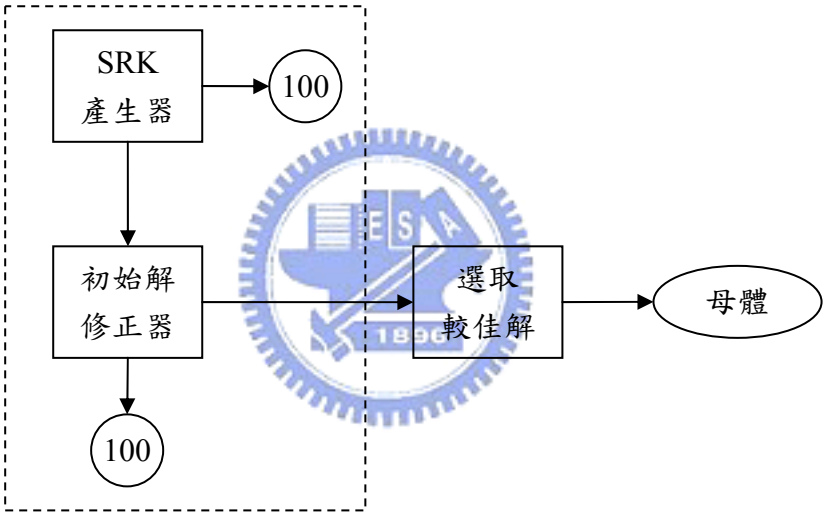


圖 3.4 初始解產生流程

(1) SRK 產生器

在介紹 SRK 的產生方法之前，首先要介紹 random key(RK)，從字面上的意思可以了解到，RK 就是未標準化的 SRK。如表 3.4 所示，RK 是使用隨機方式所產生出的非有效排程(invalid chromosome)，排序內容所代表的並非作業的排名(rank)，而是優先值(priority)；換言之，RK 並未符合作業的相依關係限制，因此不能表達作業在排程中的排序，也無法推算出合理的總完工時間 m_s。

表 3.4 非有效的排程結果(RK)

作業	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}
排名	1	10	14	30	6	25	7	11	20	35	100

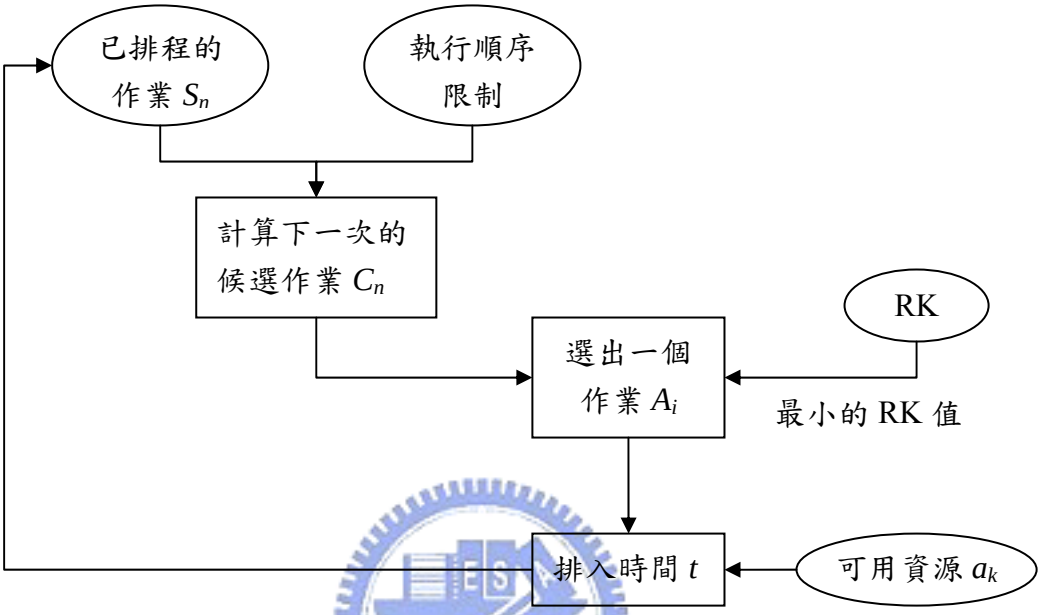


圖 3.5 逐步排程法基本架構

為此，需要使用一種轉換方法將 RK 解碼為 SRK。本研究使用啟發式解法 schedule generation scheme，簡稱 SGS(Kelley, 1963 和 Kolisch, 1996)，本研究使用其中的逐步排程法(serial schedule scheme)，求解流程如圖 3.5 所示。

欲將 n 個作業排程必須經過 n 次的遞迴。首先判斷執行順序限制和已排程(schedule set，S_n)的作業，計算下一次的候選作業有哪些(Candidate set C_n)；接著根據 RK 值，從候選作業中選出一個 RK 值最小的作業 A_i，並計算哪個時段有足夠的資源 a_k 可以分派給 A_i，最後排入時間 t ($\boldsymbol{s_i} = t$；$\boldsymbol{f_i} = t + d_i$)。

以圖 3.3 為例，假設 A_2 已排入排程中，下一次的候選作業為 A_9、A_6、A_4 和 A_3，因為 A_2 已排程，令 A_3 滿足執行順序限制，所以在下一次遞迴時，A_3 就能一併列入候選作業。所以逐步排程法必須在每一次遞迴計算 C_n。

3.2.3 母體

本研究母體選取的方式使用分散搜尋法(Scatter Search)的母體結構，有關分散搜尋法的詳細介紹在 Glover (1998)、Glover et al. (2000, forthcoming)和 Martí et al. (2006)等文獻中。

如圖 3.6，母體被稱為參考集合(reference set)，可以分成品質集合(quality set)和差異集合(diversity set)兩個部分。品質集合 B_1 是從 100 個初始解中選出 b_1 個最佳解放入 B_1 中，為了讓 B_1 集合裡解的排序有一定差異性，每個解之間要至少保持 t_1 的距離；差異集合 B_2 是將原 100 個初始解減去 b_1 後選出 b_2 個數的最佳解放入 B_2 中，為了保有排序的差異性，B_2 的每個解和 B_1 的每個解之間最短要維持 t_2 的距離。距離的計算如公式 1：

$$\text{Distance} = \frac{\sum_{n}^{i=1} |x_i - y_i|}{n} \tag{1}$$

參數設定方面，t_1 為 1.1，t_2 設定在 2.0。母體大小 b_1 和 b_2 則是以停止條件來設定，當停止條件為產生 1,000SRK 時，$b_1 = 5$，$b_2 = 3$；5,000SRK 時，$b_1 = 10$，$b_2 = 5$；50,000SRK 時，$b_1 = 28$，$b_2 = 16$。

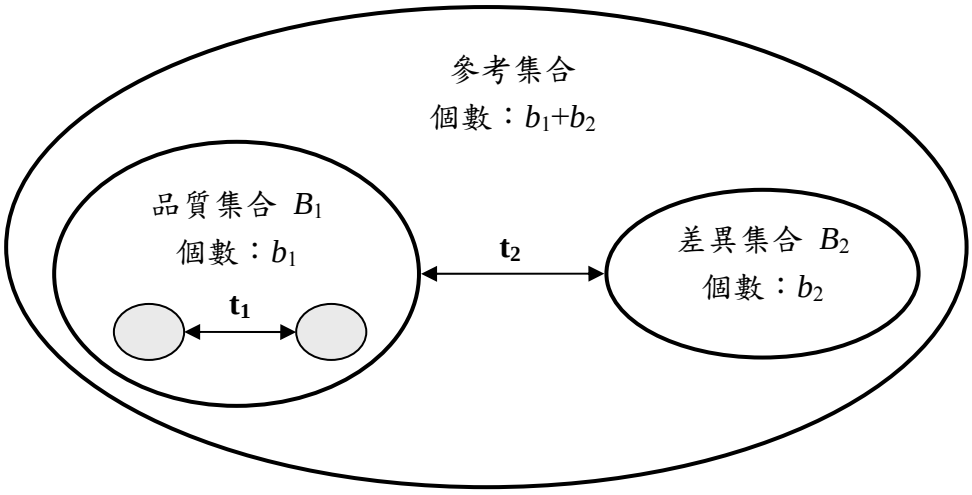


圖 3.6 分散搜尋法母體結構

3.2.4 更新解產生器

更新解產生器包含了 4 種運算子：雙點交配(standard two-point crossover)、雙點式模擬電磁演算法(two-point electromagnetism)、共識因子(consensus operator)和田口方法(Taguchi methods)，茲將分別介紹如下。

(1) 雙點交配

雙點交配是 Debels et al. (2006)的更新方法之一，交配的概念來自於基因演算法。如圖 3.7，輸入(input)為 2 條母代 SRK，隨機選取 2 個交配點，將各自介於 point 1 和 point 2 之間的 SRK 跟對方交換，可以得到 2 條子代(新的解)，藉由交換 SRK 參考不同解的排序，以產生多樣化的更新解。

圖 3.7 雙點交配示意圖

以下為雙點交配的釋例。首先隨機選取 2 個交配點 $P_{\min} \in [1, n\text{-}1]$和 $P_{\max} \in [2, n]$，其中 n 表示作業數目；D 表示$[P_{\min}, P_{\max}]$之間應有的最小距離。假設母代為 X_1 和 X_2，產生第 1 條子代的方法如下：

1. SRK＜$P_{\min}$：將 X_1 小於 $P_{\min}$ 的 SRK 減去一個很大的常數(i.e. $n = 11$)。
2. $P_{\min}$≦SRK≦$P_{\max}$：將 X_1 介於$[P_{\min}, P_{\max}]$的 SRK 換成 X_2 的 SRK。
3. $P_{\max}$＜SRK：將 X_1 大於 $P_{\max}$ 的 SRK 加上一個很大的常數。

使上述方法就能產生出第一條子代染色體 Y_1，如表 3.5 所示($P_{\min} = 2$ & $P_{\max} = 7$)，而只要將 X_1 和 X_2 的角色互換就能生出第二條染色體 Y_2。

表 3.5 產生第一條子代釋例

作業	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}
X_1	1	2	4	2	5	7	9	8	5	9	11
X_2	1	3	6	2	7	5	8	7	3	10	11
X'_2	-10						20	19		20	22
X'_2		3	6	2	7	5			3		
Y_1	-10	3	6	2	7	5	20	19	3	20	22

另外需要注意的是，使用雙點交配產生的更新解 Y_1 和 Y_2 並不保證為 SRK，所以須要使用逐步排程法將 Y_1 和 Y_2 (RK)解碼為 SRK。

近年以來，多數的演算法都已漸漸使用導引(guide)的方式，具有目標性的產生新解，但 Debels et al. (2006)卻使用交配這種非導引的更新方式，原因在於更新的架構中，母代是使用品質集合 B_1 的最佳解，以 C_2^{b1} 的方式配對並進行雙點交配，同時因為 B_1 裡的解是保有排序差異(diversity)的最佳解(總完工時間佳)，所以使用交配的方式能讓好的解互相交換排序，以期能產生近似最佳解。

(2) 雙點式模擬電磁演算法

雙點式模擬電磁演算法是 Debels et al. (2006)的另一個更新方法，起初是由 Birbil & Fang (2003)所提出的方法，其概念來自於庫倫定律(Coulomb's Law)同電相吸、異電相斥的原理。如圖 3.8 所示，將庫倫定律的概念想像在一個多維的歐幾理德空間(Euclidian space)中有 2 條 SRK $X_i \in B_1$ 和 $X_j \in B_2$，SRK 裡每一個 A_i 的排名代表 SRK 在每一個維度上的位置，而為了將較差的 X_j 吸引到好的 X_i 的位置，必須經由下列 2 個公式的計算：

$$q_{ji} = \frac{f(X_j) - f(X_i)}{f(X^{\text{worst}}) - f(X^{\text{best}})} \tag{2}$$

其中 $f(X)$表示解 X 的總完工時間

$$F_{ji} = (X_i - X_j) \cdot q_{ji} \tag{3}$$

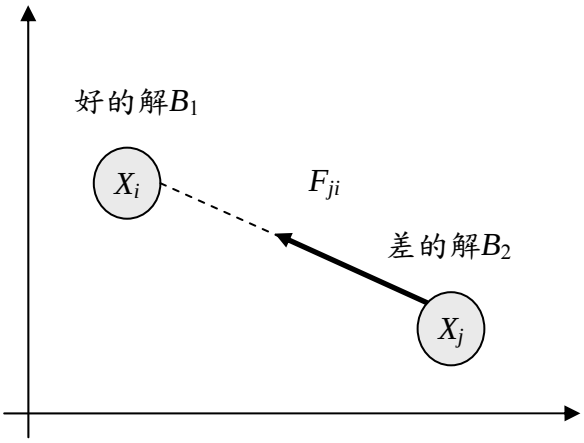


圖 3.8 模擬電磁演算法示意圖

q_{ji} 能計算出 X_i 和 X_j 之間的電荷量，也就是吸引的強度有多強，而 $f(X^{worst})$ 代表在該空間(B_1 & B_2)裡的最差解，$f(X^{best})$則是最佳解。F_{ji} 則表示 X_i 施加在的 X_j 作用力，公式的內容是將 X_i 的位置減去 X_j 的位置，乘以電荷量 q_{ji}，此時可以將 q_{ji} 視為權重。計算出 F_{ji} 之後跟 X_j 的 SRK 相加就可以得到 X_j 的新位置。

表 3.6 為釋例，首先隨機選取 2 個交配點 $P_{min} \in [1, n\text{-}1]$和 $P_{max} \in [2, n]$ (i.e. $P_{min} = 2$；$P_{max} = 7$)，其中 n 表示作業數目；D 表示$[P_{min}, P_{max}]$之間應有的最小距離。SRK 為 X_1 和 X_2，產生新解的方法如下：

1. SRK＜P_{min}：將 X_2 小於 P_{min} 的 SRK 減去一個很大的常數(i.e. $n = 11$)。
2. $P_{min} \leqq$SRK$\leqq P_{max}$：將 X_2 介於$[P_{min}, P_{max}]$的 SRK 計算出 F_{21}。
3. P_{max}＜SRK：將 X_2 大於 P_{max} 的 SRK 加上一個很大的常數。
4. $X_2 + F_{21} + X'_2 = X''_2$
5. X''_2 以逐步排程法解碼為 X'''_2 (SRK)。

舉例來說，X_2 在 A_1 的 SRK 小於 $P_{min} = 2$，所以 SRK 減去 $n = 11$，等於 X'_2 的 -10；接著 A_2 的 SRK 介於$[P_{min}, P_{max}]$之間，因此可以用公式 2 & 3 求得 F_{21} 如下(假設 $X^{worst} = 19$；$X^{best} = 15$)：

表 3.6 產生第一條子代釋例

作業	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	m_s
X_1	1	2	4	2	5	7	9	8	5	9	11	15
X_2	1	3	6	2	7	5	8	7	3	10	11	16
F_{12}		-0.25	-0.5	0	-0.5	0.5		0.25	0.50			
X'_2	-10						19			21	22	
X''_2	-10	2.75	5.5	2	6.5	5.5	19	7.25	5.50	21	22	
X'''_2	1	2	4	2	5	7	9	8	5	9	11	15

$$q_{21} = \frac{16-15}{19-15} = 0.25$$

$$F_{21} = (2-3)\cdot 0.25 = -0.25$$

求得 F_{21} 等於 -0.25 之後，代表 X_2 的 A_2 要向 X_1 的 A_2 移動 0.25 的距離，得到 2.75 的位置，以此類推可以求出新的 RK(X''_2)，最後用逐步排程法得到新的 SRK(X'''_2)。

如上述釋例所示，雙點式模擬電磁演算法的求解數目為，將 B_1 和 B_2 集合的解進行配對，可以有 $b_1 \times b_2$ 對，並且每一對只產生一個解。

(3) 共識因子

所謂的共識因子，其概念為一個解集合 S_c 中，所有解$\{X_1, \ldots, X_P\}$對作業 A_i 排序的意見。如表 3.7 為一個 $n \times n$ 的機率性矩陣 M，稱為共識矩陣(consensus matrix)，其中每一格 m_{ij} 表示 S_c 裡的解認為作業 A_i 應該在作業 A_j 前執行的共識比率。例如，m_{42} 表示百分之 70 的解認為 A_4 應該在 A_2 前執行比較好。

使用共識因子(吳政翰, 2006)來產生新的解，即是希望可以從上一代好的解尋找資訊，建立一個導引(guided)機制，讓好的排序保留，不好的排序則自然淘汰，最終能產生近似最佳解。產生新染色體的方法依序如下：

表 3.7 共識矩陣 M

M		作業 (j)										
		1	2	3	4	5	6	7	8	9	10	11
作業 (i)	1		1	1	1	1	1	1	1	1	1	1
	2	0		1	0.3	0.9	1	1	1	0.6	1	1
	3	0	0		0	0.3	0.3	1	1	0.3	1	1
	4	0	0.7	1		0	1	1	1	0.8	1	1
	5	0	0.1	0.7	1		1	1	1	0.3	1	1
	6	0	0	0.7	0	0		1	1	0	1	1
	7	0	0	0	0	0	0		0	0	0.5	1
	8	0	0	0	0	0	0	1		0	1	1
	9	0	0.4	0.7	0.2	0.7	1	1	1		1	1
	10	0	0	0	0	0	0	0.5	0	0		1
	11	0	0	0	0	0	0	0	0	0	0	

1. 建立共識矩陣

流程如圖 3.9，共識矩陣 $M(n \times n$ matrix)初始為一個 0 矩陣，從 S_c 中依序抽出每一條解，並且計算該解的對偶優先矩陣 Q，形成新矩陣 $M(M = M + Q)$。

計算對偶優先矩陣的方法如下：首先初始一個矩陣 $Q = [q_{ij}]$, $(1 \leqq i, j \leqq n)$為 0 矩陣，從 S_c 中選出一條解 $X_p(1 \leqq p \leqq P)$，例如表 3.1，X_1 = {1; 2; 4; 5; 7; 2; 8; 9; 5; 10; 11}。其中 X_1 在 A_1 的排名為 1，小於 A_2 的排名 2，所以 q_{12} 加 1；A_2 的排名為 2，小於 A_3 的排名 4，所以 q_{23} 加 1…依此類推直到計算完矩陣 Q (表 3.8)，如果有排名相同的情形就改成加 0.5。

計算完所有矩陣 Q 之後，接著將矩陣 M 標準化為 $M = M / N(S_c)$，$N(S_c)$代表解集合 S_c 裡解的總個數，如表 3.9 是以 30 個解為例。由於共識矩陣有對偶的特性，所以用公式(5) $M_{ij} + M_{ji} = 1$ 可以計算出另一半的共識比率，最終結果如表 3.7 的矩陣。

先前提到，計算對偶優先矩陣的方法共有 3 種：No weighting、Linear weighting 和 Power weighting。上述計算方法是 No weighting，也就是累加 q_{ij} 的方式是加 1；Linear weighting 則是考慮總完工時間 m_s 的重要性，經由計算求得累加的數字 w_p，其公式如下：

$$w_p = \frac{f(X^{\text{worst}}) - f(X_p)}{f(X^{\text{worst}}) - f(X^{\text{best}})} \tag{4}$$

上述的公式參考 Debels et al. (2006)修改的模擬電磁演算法公式，$f(X^{\text{worst}})$表示解集合 S_c 裡的最差解，$f(X^{\text{best}})$則是最佳解。如此一來可以依每條解的 m_s 計算出相對重要性。Power weighting 則是將公式 4 取平方，加強權重的效果。

根據共識比率的建議，可以清楚知道 A_i 之間應該如何排序，然而在矩陣中卻可以發現 1 或 0 的機率，造成這樣的原因有兩種：第一，作業本身的執行順序限制某些作業必須排在前面；第二，經由世代演化後形成較佳的共識建議。因此產生更新解時，應以不違背這兩種限制為前題。

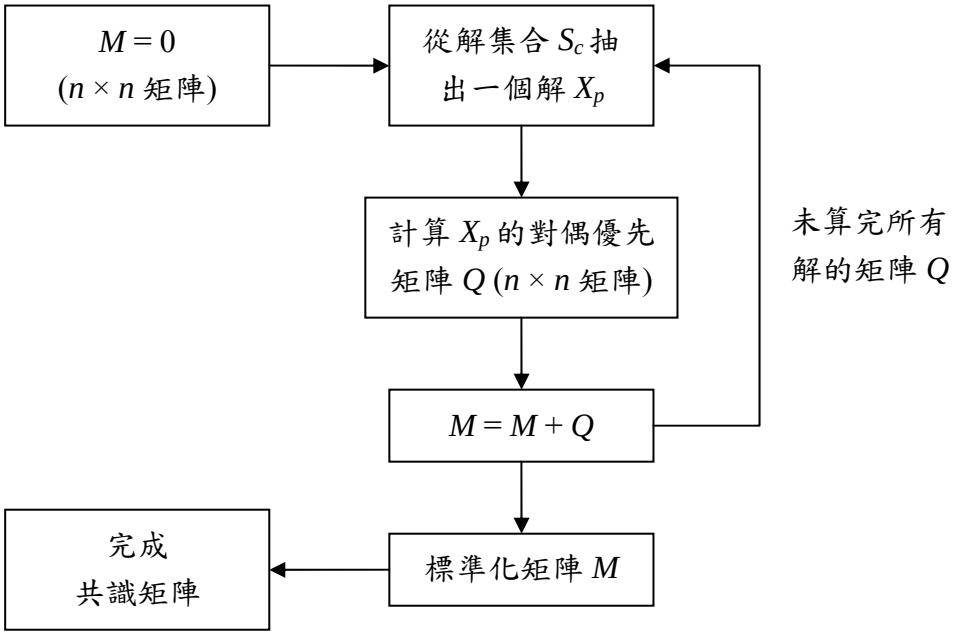


圖 3.9 共識矩陣建立流程

表 3.8 對偶優先矩陣 Q

Q		作業（j）										
		1	2	3	4	5	6	7	8	9	10	11
作業（i）	1		1	1	1	1	1	1	1	1	1	1
	2	0		1	1	1	0.5	1	1	1	1	1
	3	0	0		1	1	0	1	1	1	1	1
	4	0	0	0		1	0	1	1	0	1	1
	5	0	0	0	0		0	1	1	0	1	1
	6	0	0	0	0	0		1	1	1	1	1
	7	0	0	0	0	0	0		1	0	1	1
	8	0	0	0	0	0	0	0		0	1	1
	9	0	0	0	0	0	0	0	0		1	1
	10	0	0	0	0	0	0	0	0	0		1
	11	0	0	0	0	0	0	0	0	0	0	

表 3.9 矩陣 M

M		作業（j）										
		1	2	3	4	5	6	7	8	9	10	11
作業（i）	1		30	30	30	30	30	30	30	30	30	30
	2	0		30	8.5	26	30	30	30	17	30	30
	3	0	0		0	9	9	30	30	9	30	30
	4	0	0	0		30	30	30	30	24	30	30
	5	0	0	0	0		30	30	30	8.5	30	30
	6	0	0	0	0	0		30	30	0	30	30
	7	0	0	0	0	0	0		0	0	15	30
	8	0	0	0	0	0	0	0		0	30	30
	9	0	0	0	0	0	0	0	0		30	30
	10	0	0	0	0	0	0	0	0	0		30
	11	0	0	0	0	0	0	0	0	0	0	

2. 產生新染色體

如先前提到，以共識因子產生新解的方式是參考共識矩陣，以選擇性隨機的方式進行排序，所以共識比率愈高表示形成某一種排序的可能性愈高。

首先建立一個空集合 Y 用來存放已排序的作業，$X = \{A_i ; 1 \leqq i \leqq n\}$ 表示未排序的作業集合。產生染色體 Y 的流程如下：

Step 1：建立合理區

由於任務有先後關係的限制，所以必須以共識矩陣的 0、1 限制建立合理區，將不能插入的位置排除在外。先設定 A_i 的初始合理區域起點 $h_s = 0$，迄點 $h_e = n$，n 表示專案的作業數目，$h(A_i)$表示待決定的 A_i 優先值(priority value)。假設目前有 N 個作業已完成排序，則對於每個 $A_q \in Y\ (1 \leqq q \leqq N)$，如果 $m_{iq} = 0$ 且 $h(A_q) > h_s$，則 $h_s = h(A_q)$；反之若 $m_{iq} = 1$ 且 $h(A_q) < h_e$，則 $h_e = h(A_q)$。更新完後的合理區域，表示 A_i 不論插入這個區域的哪個位置都不會違反順序限制。

For $k = 1$ to N /* for each task in Y_h */

If $m_{iq} = 0$ and $h(A_q) > h_s$，then $h_s = h(A_q)$

If $m_{iq} = 1$ and $h(A_q) < h_e$，then $h_e = h(A_q)$

Endfor

Step 2：停止條件

當合理區建立之後，首先要判斷合理區域(包含 h_s 和 h_e)的作業數 $n(Y_h)$是否少於兩個，如果是則將作業 A_i 插入合理區域 h_s 和 h_e，並決定優先值 $h(A_i) = (h_s + h_e) / 2$；如果 $n(Y_h) > 2$，則進行 Step 3 直到 $n(Y_h) \leqq 2$ 為止。

Step 3：在合理區中選擇參考作業 A_{r*}

已知合理區域所有的作業集合 $Y_h = \{A_q \mid h_s \leqq h(A_q) \leqq h_e, A_q \in Y\}$ 大於 2 個，接著要尋找參考作業 A_{r*}，讓 A_i 決定插入合理區的哪個位置。尋找參考作業 A_{r*} 時，以共識比率最大的為優先，所以有 2 種尋找方式，第一種是 A_i 在 A_q 之前共識比率 m_{iq} 最大的作業 A_k 如公式(6)；第二種是 A_i 在 A_q 之後共識比率 m_{qi} 最大的作業 A_r 如公式(7)。最後將兩者的共識比率比較如公式(8)取最大的成為參考作業 A_{r*}。

$$A_k = \text{Max}\{ m(A_q, A_i) \}, A_q \in Y \quad (6)$$

$$A_r = \text{Max}\{ m(A_i, A_q) \}, A_q \in Y \quad (7)$$

$$A_{r*} = \text{Max}\{ m(A_k, A_i), m(A_i, A_r) \} \quad (8)$$

Step 4：根據參考作業 A_r*執行伯努力(Bernoulii)試驗

尋找到參考作業 A_{r*} 之後，必須決定 A_i 要排在 A_{r*} 之前或之後，而為了讓原本的排序產生變異，根據 A_i 與 A_{r*} 的共識比例 m_{ir*} 做為伯努力試驗的機率。試驗成功代表 A_i 排在 A_{r*} 之前，所以 $h(A_i) < h(A_{r*})$，合理區域更新為 $h_e = h(A_{r*})$；試驗失敗代表 A_i 排在 A_{r*} 之後，所以 $h(A_i) > h(A_{r*})$，合理區域更新為 $h_s = h(A_{r*})$。

Step 5：更新合理區域

實驗成功：合理區域更新為 $h_e = h(A_{r*})$。

實驗失敗：合理區域更新為 $h_s = h(A_{r*})$。

更新完畢後回到 Step 2。

以圖 3.10 為例，要重新排序 11 個作業，A_1 和 A_{11} 為開始和結束節點，所以 A_2 直接放入 A_1 和 A_{11} 之間，接著要插入 A_3。首先在表 3.7 的共識矩陣中，A_3 在 A_1 和 A_2 前的共識比率為 0，所以將合理區的起點 h_s 設在 A_2，接著 A_3 在 A_{11} 前的共識比率為 1，便將合理區的迄點 h_e 設在 A_{11}。由於合理區域的作業數 $n(Y_h)$ 已小於 2，所以直接將 A_3 插入 A_2 和 A_{11} 之間的位置。

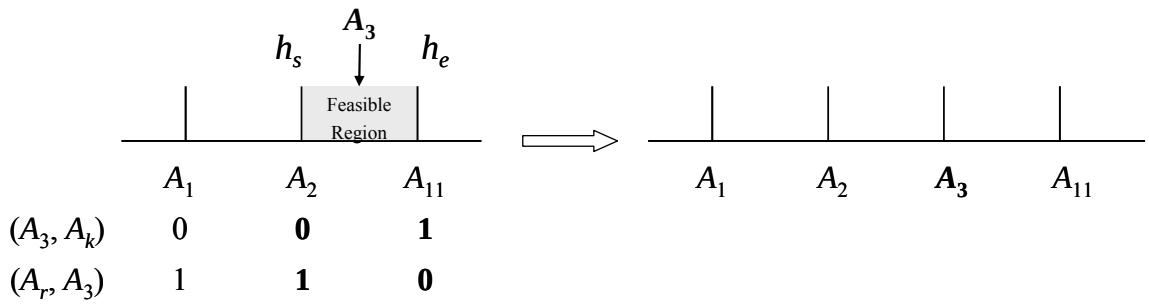


圖 3.10 插入 A_3 釋例

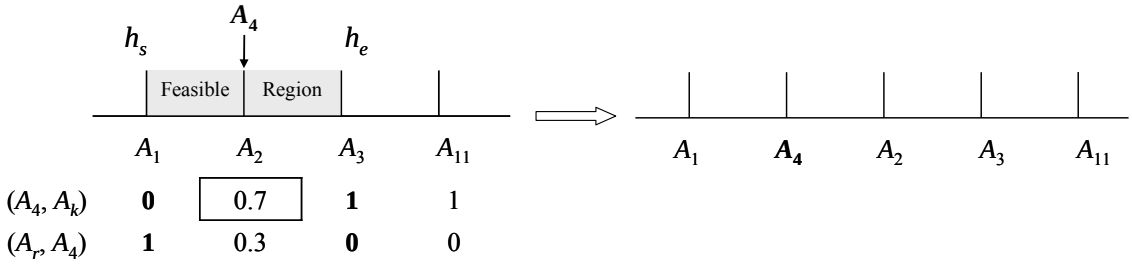


圖 3.11 插入 A_4 釋例

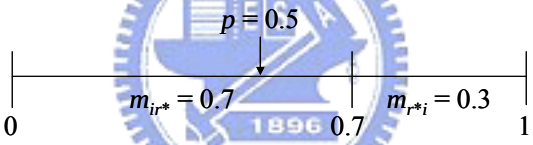


圖 3.12 伯努力實驗釋例

接著的圖 3.11，已知合理區域在 A_1 和 A_3 之間，而 $n(Y_h) > 2$，所以要尋找參考作業 A_r*。已知 $m_{ik} = 0.7$，$m_{ri} = 0.3$，所以 $m_{ir^*} = 0.7$，$A_r^* = A_2$ 進行伯努力試驗。如圖 3.12，若隨機數值 p 落在[0, 0.7]區間表示實驗成功，合理區域更新為 $h_e = h(A_2)$，此時因為 $n(Y_h) \leqq 2$，所以決定將 A_4 插在 A_1 和 A_2 之間。假設最終作業排序為 $A_1 \rightarrow A_4 \rightarrow A_2 \rightarrow A_6 \rightarrow A_3 \rightarrow A_9 \rightarrow A_5 \rightarrow A_7 \rightarrow A_8 \rightarrow A_{10} \rightarrow A_{11}$，可以經由逐步排程法將此排序轉換為 SRK 格式，$Y = \{1; 3; 5; 2; 6; 4; 8; 9; 6; 10; 11\}$。

在實際應用共識因子時，有時為了讓產生的解排序有更大的變化，會在進行伯努力實驗時將共識比率 m_{ir^*}做調整，若以 α 做為調整因子，以 $m_{ir^*} \times \alpha$ 做為調整方式，α 有 4 種設定方法：第一種，$\alpha = 1$(先前介紹的方法)。第二種，$\alpha = 0.8$ 稱為降級因子，為吳政翰 (2006)所提出。第三種，$\alpha = 0.8$ combined 1.2，由於使

用降級因子只會單純讓所有的共識降低，為此本研究結合 1.2。當 $m_{ir^*} > 0.5$ 時以 0.8 降級；$m_{ir^*} < 0.5$ 時以 1.2 升級；$m_{ir^*} = 0.5$ 不調整；第四種，$\alpha = 0.8$ and 1.2，和 combined 的差別在於這種調整方式並非綜合指標，而是降級因子和升級因子獨立產生各自的更新解。

建立共識矩陣的抽樣方法有 2 種：第一種，以母體建構($b_1 + b_2$)；第二種，母體加上雙點交配產生的新解，取 28 個最佳解做為樣本。共識因子每一代產生的新解個數為 $b_1 \times b_2$ (跟雙點式模擬電磁演算法相同)。

(4) 田口方法

田口方法(Taguchi, 1986 & 1987；蘇朝墩, 2002；黎正中, 2000)的起源，是由品管大師田口玄一(Genichi Taguchi) 博士於 1949 年間，發現傳統的實驗設計方法並不適合實驗應用，因而逐漸發展「田口式品質工程」，簡稱為田口方法(Taguchi methods)。

田口方法的基本概念，是一種經由系統化的參數設計方法，使得產品對外在因素的敏感度降至最低，所以重點在於如何最佳化產品的參數組合，方法是使用直交表，目的是希望以較少的實驗次數達到最佳化的目標。田口直交表的符號說明(Taguchi, 1987；蘇朝墩, 2002)如下：

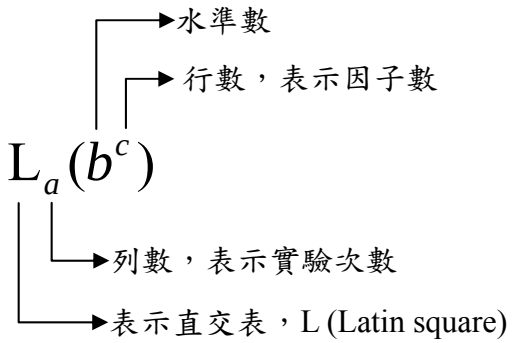


圖 3.13 田口直交表符號說明

表 3.10 $L_8(2^7)$直交表

L8		行(i)						
		A_1	A_2	A_3	A_4	A_5	A_6	A_7
列(a)	L_1	1	1	1	1	1	1	1
	L_2	1	1	2	1	2	2	2
	L_3	1	2	1	2	1	2	2
	L_4	1	2	2	2	2	1	1
	L_5	2	1	1	2	2	1	2
	L_6	2	1	2	2	1	2	1
	L_7	2	2	1	1	2	2	1
	L_8	2	2	2	1	1	1	2

田口博士一共設計了 18 種直交表，以 $L_8(2^7)$直交表為例可以描述如表 3.10，表中共有 7 個因子，每個因子 2 水準，有 8 次實驗要進行。若以專案問題來模擬，可以將 7 個因子想像成有 7 個作業，2 水準想像成有 2 條 SRK，而行列交叉欄位 L_{ai} 的內容所代表的意義為：”1”是第 1 條 SRK 的排名(rank)；”2”是第 2 條 SRK 的排名。但是為了要使用田口博士所設計的直交表，必須實驗內容的因子和水準組合都相同；然而，本研究欲探討的 3 種情境 J30、J60 和 J120 並沒有相符的直交表可以使用。為此，本研究以判斷式 $2^k \geqq n$ 來建構適合的直交表，以 J30(30 個作業)為例，n 為 30，所以至少需要 $2^5 = 32$ 的直交表(實驗次數為 32，因子數為 31)。

介紹完田口方法的概念後，以下介紹本研究如何使用田口方法產生新解：

Step 1：建構 *n* 因子 2 水準直交表

Step 2：產生更新解

輸入(input)為 2 條 SRK (S_1, S_2)，輸出(output)為 1 條 RK(S_b)，依直交表的實驗次數求得每一筆實驗組合 L_a 的總完工時間 $m_s(L_a)$，為了計算出每一個作業 A_i

應使用 S_1 或 S_2 的 SRK 才能形成最佳參數組合 S_b，必須累計 A_i 在不同 L_{ai} 的數值等於 1 或 2 的 $1 / m_s(L_a)$總和(令 Sum1 為 $L_{ai} = 1$ 時的總和；令 Sum2 為 $L_{ai} = 2$ 時的總和)，流程如下：

For $i = 1$ to n

For $a = 1$ to 2^k

If $L_{ai} = 1$，then Sum1 = Sum1 + $(1 / m_s(L_a))$

If $L_{ai} = 2$，then Sum2 = Sum2 + $(1 / m_s(L_a))$

Endfor

If (Sum1 >= Sum2)

$S_b(A_i) = S_1(A_i)$

else

$S_b(A_i) = S_2(A_i)$

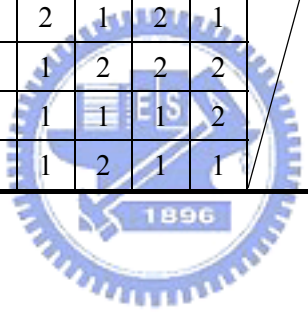
Endfor

以圖 3.3 為例，假設 2 條 SRK 為 S_1 = {1; 2; 4; 2; 5; 7; 9; 8; 5; 9; 11}，S_2 = {1; 3; 6; 2; 7; 5; 8; 7; 3; 10; 11}，因為有 11 個作業(因子)，所以至少需要 $2^4 = 16$ 的直交表。如表 3.11 所示，2 條 SRK 可以透過直交表產生 16 種實驗組合，由於只有 11 個作業，所以 A_{12} 到 A_{15} 的資料便省略不看。

首先計算每個實驗組合的總完工時間 $m_s(L_a)$，接著取倒數，最後依序從 A_1 到 A_{11} 個別計算 Sum1 和 Sum2 的值，並比較以決定最佳組合 S_b，由於總完工時間為望小特性，取倒數以後變望大特性，所以 Sum1 和 Sum2 的比較應該選較大者。最後可得到 S_b = {1; 2; 4; 2; 5; 5; 8; 8; 3; 9; 11}，因為這個組合是 RK，所以要用逐步排程法轉換為 SRK。

表 3.11 田口方法釋例

No.	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	$m_s(L_a)$	$1 / m_s(L_a)$
L_1	1	1	1	1	1	1	1	1	1	1	1					15	1/15
L_2	1	1	1	2	1	1	2	1	2	2	1					17	1/17
L_3	1	1	2	1	2	2	1	2	1	2	2					16	1/16
L_4	1	1	2	2	2	2	2	2	2	1	2					18	1/18
L_5	1	2	1	1	2	1	1	2	2	1	2					15	1/15
L_6	1	2	1	2	2	1	2	2	1	2	2					20	1/20
L_7	1	2	2	1	1	2	1	1	2	2	1					19	1/19
L_8	1	2	2	2	1	2	2	1	1	1	1					15	1/15
L_9	2	1	1	1	1	2	2	1	1	1	2					15	1/15
L_{10}	2	1	1	2	1	2	1	1	2	2	2					15	1/15
L_{11}	2	1	2	1	2	1	2	2	1	2	1					17	1/17
L_{12}	2	1	2	2	2	1	1	2	2	1	1					15	1/15
L_{13}	2	2	1	1	2	2	2	2	2	1	1					15	1/15
L_{14}	2	2	1	2	2	2	1	2	1	2	1					16	1/16
L_{15}	2	2	2	1	1	1	2	1	2	2	2					15	1/15
L_{16}	2	2	2	2	1	1	1	1	1	1	2					19	1/19
$\boldsymbol{S_b}$	2	1	1	1	1	2	2	1	2	1	1					15	



參數設定方面，必須決定田口方法每個世代應該選多少對來更新，因為田口方法分別跟其他 3 種更新方法組合，所以參數設定也有所變化。若田口方法跟雙點交配組合，每一個世代產生 C_2^{b1+b2} 條解；若是跟雙點式模擬電磁演算法或共識因子搭配，則產生 C_2^{b1} 條解。

3.2.5 修正器模組

修正器(Li & Willis, 1992；Őzadamar & Ulusoy, 1996；Debels et al., 2006)用於改善逐步排程法產生的排程 S，概念為減少專案排程期間資源的閒置，名為 Forward–Backward Improvement，FBI。如圖 3.14 為圖 3.3 釋例的排程結果，修正過程有 2 步驟：

Step 1：排程向後移動(backward–pass)

根據排程(a)的 SRK，可以得到各項作業 A_i 的完工時間，藉此由大到小決定出一個排序 A_{10}→A_8→A_7→A_5→A_9→A_4→A_3→A_2→A_6，以 $m_s = 18$ 為底限，並且不違反順序 DAG 限制為前題，依序將作業的排程向後移動至如圖(b)的 SRK'。

Step 2：排程向前移動(forward–pass)

同理，根據排程(b)的 SRK'，可以得到 A_i 的開工時間，藉此由小到大決定出一個排序 A_2→A_3→A_4→A_9→A_5→A_8→A_6→A_{10}→A_7，依序將作業的排程向前移動至如圖(c)所示，即可得到修正後的結果(SRK*)。

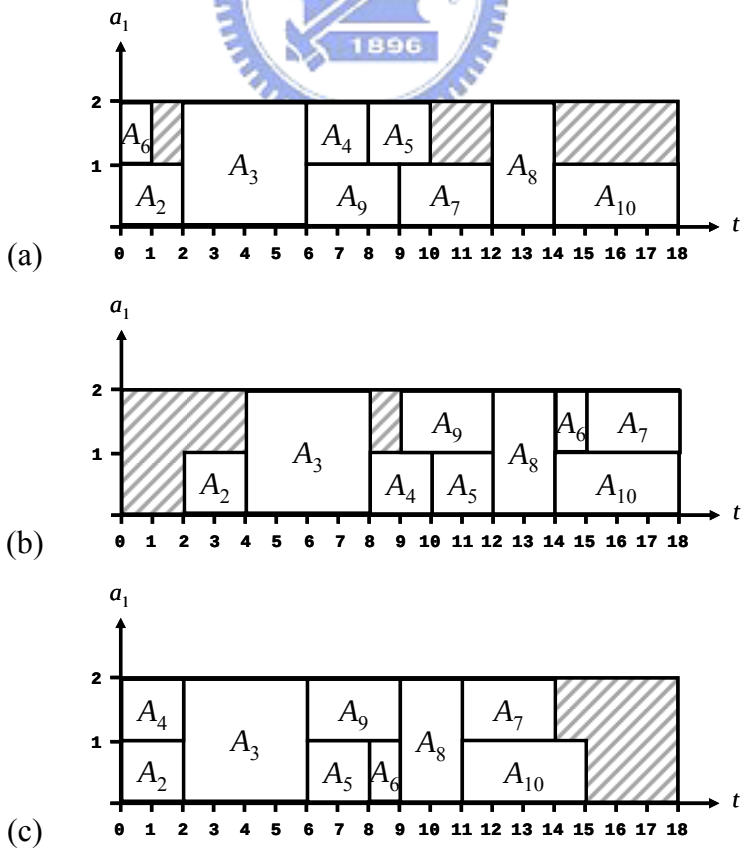


圖 3.14 修正器釋例

3.2.6 彙總與母體更新器

本研究所使用的母體更新方法，稱為靜態更新(static update)。靜態更新(Martí et al., 2006)是將"母體∪更新解"後，取出最佳的 $b_1 + b_2$ 個解形成新母體。本研究以相同的方式進行彙總和更新母體，同樣的，更新母體時仍然要讓 B_1 和 B_2 分別維持 t_1 和 t_2 的位置。

3.2.7 停止條件

第二章 2.4 節提到，本研究是以 PSPLIB 做為測試的範本，參考 Kolisch & Hartmann (2006)在文獻的比較方式，是以限制 SRK 產生的數目 N_s 做為比較標準，其上限 N_L 有 1,000、5,000 和 50,000 三種。因此演化過程(不包括初始解流程)每產生一個 SRK 就將 N_s 加 1，當 N_s 的數目達到上限 N_L 時，則停止演化，並且當時最好的解為最佳解。

第四章　實驗結果

在第三章概述本研究的實驗方法，分成導引實驗和綜合實驗，本章則詳細說明 2 種實驗的內容和分析結果。

4.1 導引實驗(pilot study)

圖 3.2 闡述了導引實驗的流程，首先找出最佳版本，其次是參數微調，最後是分析實驗結果，建立新的架構圖。本節將以上述流程介紹導引實驗的內容。

4.1.1 選出最佳版本(Step 1)

Step 1 主要將 C1(雙點交配)、C2(雙點式模擬電磁演算法)、C7(共識因子)和 C15(田口方法)等 4 種方法兩兩配對進行實驗，V0 為 Debels et al.(2006)的方法。從實驗數據(合計所有例子的總完工時間)可以看出，V5 的實驗結果最為接近 V0，並且在 1,000 SRK 收斂的品質跟 V0 沒有差異，表示 C7 或許可以取代 C2；另外，從 V5 比 V18 和 V13 比 V18 的比較中可以發現，C1 的求解表現比 C2 的好。而根據 V13、V26 和 V31 的結果，使用田口方法的組合，求解品質都相當的差。所以從表 4.1 的結果得知，更新解產生器的最佳組合為：雙點式交配(C1)和共識因子(C7)。詳細的實驗數據放在附錄中。

表 4.1 更新解產生方法實驗結果

版本	更新解產生器組合				合計總完工時間			
					J30 (480 個例子)		J60 (480 個例子)	
					1,000 SRK	5,000 SRK	1,000 SRK	5,000 SRK
V0	A1	B1	C1 + C2	D1	***28,410***	***28,355***	***38,765***	***38,554***
V5			C1 + C7		28,404	28,360	38,765	38,614
V13			C1 + C15		28,470	28,413	39,100	38,964
V18			C2 + C7		28,408	28,364	38,803	38,660
V26			C2 + C15		28,479	28,414	39,130	38,982
V31			C7 + C15		28,493	28,415	39,163	38,988

4.1.2 參數微調(Step 2)

Step 1 選出的最佳版本是 V5 的 C1 + C7，但因為共識因子(C7)只選出一種參數來實驗，所以為了瞭解調整參數對求解品質的影響，在 Step 2 調整 C7 參數後得到的實驗結果如表 4.2 的數據，並從整體表現能力和部分表現能力來比較。

整體表現能力來看，以 V1 和 V7 的表現最為平均，跟 V5 的差異不大；V6 和 V9 雖然看來跟 V5 也差不多，但是在 1,000 SRK 的收斂能力比較差。以部分表現能力來看，V4 在 J30 的表現很好，甚至比 V0 還好，但在 J60 卻是所有參數組合表現最差的；V2、V3、V8、V10、V11 和 V12 雖然在 J30 表現跟 V5 差不多，但是在 J60 就完全比不上 V5 的結果。

選擇整體表現較平均的 V1 和 V7，部分表現好的 V4，以及原本的 V5 進行下一步的測試。

表 4.2 參數調整實驗結果

版本	更新解產生器組合	合計總完工時間			
		J30 (480 個例子)		J60 (480 個例子)	
		1,000 SRK	5,000 SRK	1,000 SRK	5,000 SRK
V5	C7 (Linear + 1)	***28,404***	***28,360***	***38,765***	***38,614***
V1	C3 (No + 1)	28,402	28,358	38,785	38,614
V2	C4 (No + 0.8)	28,409	28,362	38,799	38,635
V3	C5 (No + 0.8 combined 0.2)	28,407	28,358	38,788	38,623
V4	C6 (No + 0.8 and 1.2)	28,396	28,353	38,810	38,636
V6	C8 (Linear + 0.8)	28,413	28,363	38,797	38,618
V7	C9 (Linear + 0.8 combined 1.2)	28,410	28,362	38,772	38,619
V8	C10 (Linear + 0.8 and 1.2)	28,407	28,358	38,792	38,630
V9	C11 (Power + 1)	28,420	28,360	38,776	38,614
V10	C12 (Power + 0.8)	28,412	28,357	38,788	38,623
V11	C13 (Power + 0.8 combined 1.2)	28,406	28,358	38,789	38,620
V12	C14 (Power + 0.8 and 1.2)	28,413	28,357	38,792	38,628

4.1.3 分析與改善(Step 3)

根據表 4.1 和表 4.2 的結果，選出 V1、V4、V5 和 V7 繼續測試，雖然有其他方法表現不佳，但仍需要分析求解能力差的原因，在此將分析內容分成以下 3 個部分來探討：

(1) 雙點交配與雙點式模擬電磁演算法

在比較兩者之前，先要再次說明 Debels et al. (2006)使用的母體結構。分散搜尋法是藉由 t_1 和 t_2 同時讓母體擁有品質(quality)和差異(diversity)2 種特性。而雙點交配與雙點式模擬電磁演算法正是在這種結構下運行，雙點交配是將品質集合 B_1 的解進行交配，所以常常能求得近似最佳解；雙點式模擬電磁演算法則是將差異集合 B_2 的解，應用庫倫定律的原理讓 B_2 的解能參考 B_1 好的排序，在每一世代的演化過程能為 B_1 產生多樣且較佳的解。

所以綜合上述，雙點交配的目的是為了產生近似最佳解；而雙點式模擬電磁演算法雖然也可能在更新的過程中搜尋到近似最佳解，但主要目的是盡可能產生多樣的較佳解，所以從求解的方向來看，可以看出兩者的優劣之處，但共同使用的時候可以有互補的效果(如 Debels et al., 2006)，因而能有很好的求解效益。

(2) 田口方法

在實驗過程中，田口方法的績效明顯過差，首先從方法的結構來分析。田口方法是將 2 條解以直交表進行實驗來產生 1 條新的解，而問題正出在以直交表進行實驗的部分。由於本研究問題的停止條件是為 SRK 產生數量設上限，並非像過去以世代數做停止條件，所以進行田口方法的過程中極有可能因為直交表進行的實驗組合太多，使得以田口方法需要產生許多 SRK 才能得到 1 條新解。

若以 J30-1,000 SRK 的停止條件為例，母體的 $b_1 = 5$，$b_2 = 3$，使用 V13 (雙點交配+田口方法)的組合，田口方法每世代要以 C_2^{5+3} 選出 28 對解來更新，一對需做 2^5 次實驗，才產生 1 個新解，完成一次田口方法需 28 * (32 + 1) = 924 個

SRKs，加上 C1 的新解等於 924 + 20 = 944 個 SRK，最後再加上修正器修正(每修正一次多 2 條解)的數量，等於 944 * 3 = 2832 個 SRK。也就是使用 V13 這個組合，在 1,000 SRK 的停止條件下只能跑 0.35 世代，即使以田口方法產生的新解績效會比一般的新解好，但無法在不到一個世代的演化內彌補這個差距。

反觀 V0 (雙點交配+雙點式模擬電磁演算法)，使用一次更新方法只產生 20 + 5 * 3 = 35 個新解，所以演算法至少可以跑 1000 / 35 ≒ 9.5 個世代，因此可以從這個例子發現田口方法的績效較差的原因。

(3) 共識因子

本導引實驗選出的最佳版本，是以共識因子取代雙點式模擬電磁演算法，而為了在相同的基準下比較績效，共識因子產生的更新解數目也是 $b_1 \times b_2$ 個解。若希望產生近似最佳解，必須用「有差異(diversity)的最佳解」做為母代，而分析共識因子能產生好解的原因，主要是因為共識矩陣集合了品質集合與差異集合，利用矩陣的結構維持好的序(0 和 1 的限制)，將非 0 或 1 的關係重組排序，得到更好更多樣的解。所以要有好的共識因子，必須把重點放在共識矩陣上。

然而，就目前的共識矩陣來看，抽樣的數量有很大的問題。因為共識矩陣是使用母體($b_1 + b_2$)來形成的，當停止條件為 1,000 SRK 時母體只有 8 個，所以共識矩陣的樣本可能會太少，但到 50,000 SRK 母體增加到 44 個。為了改善這個缺點，考慮到以共識因子產生新解之前，會先使用雙點式交配(C1)，所以將雙點式交配產生的新解(最少 20 個)加上母體(最少 8 個)，共識矩陣就可以有 28 個樣本。若使用較大的停止條件，母體+新解的數量大於 28 個，就取 28 個最佳解為樣本。

總結以上論點，在導引實驗中整理出共識因子的 4 種參數組合：No weighting + 1、No weighting + 0.8 and 1.2、Linear weighing + 1 和 Linear weighting + 0.8 combined 1.2，同時共識矩陣的抽樣方法變成”共識因子 A(母體)”和”共識因子 B(母體+新解)”兩種，所以共有 8 個版本在綜合實驗中測試。

4.2 綜合實驗(comprehensive study)

在導引實驗的結論中，決定有 8 個版本進行綜合實驗(如圖 4.1)，其中只有共識因子 B 是新的版本，本節依照 3.1.2 小節的流程：找出最佳版本→分析與比較→後續建議與改善，介紹綜合實驗的內容。

4.2.1 選出最佳版本(Step 1)

如圖 4.1 為綜合實驗的架構，根據架構的內容可以整理成表 4.3，版本的編號接續從 V39 到 V46 進行綜合實驗。

表 4.3 演算法組合版本

初始解	母體	更新解產生器		更新解修正器
A1	B1	C1	**V39**：D1 (No + 1)	E1
			V40：D2 (No + 0.8 and 1.2)	
			V41：D3 (Linear + 1)	
			V42：D4 (Linear + 0.8 combined 1.2)	
			V43：D5 (No + 1)	
			V44：D6 (No + 0.8 and 1.2)	
			V45：D7 (Linear + 1)	
			V46：D8 (Linear + 0.8 combined 1.2)	

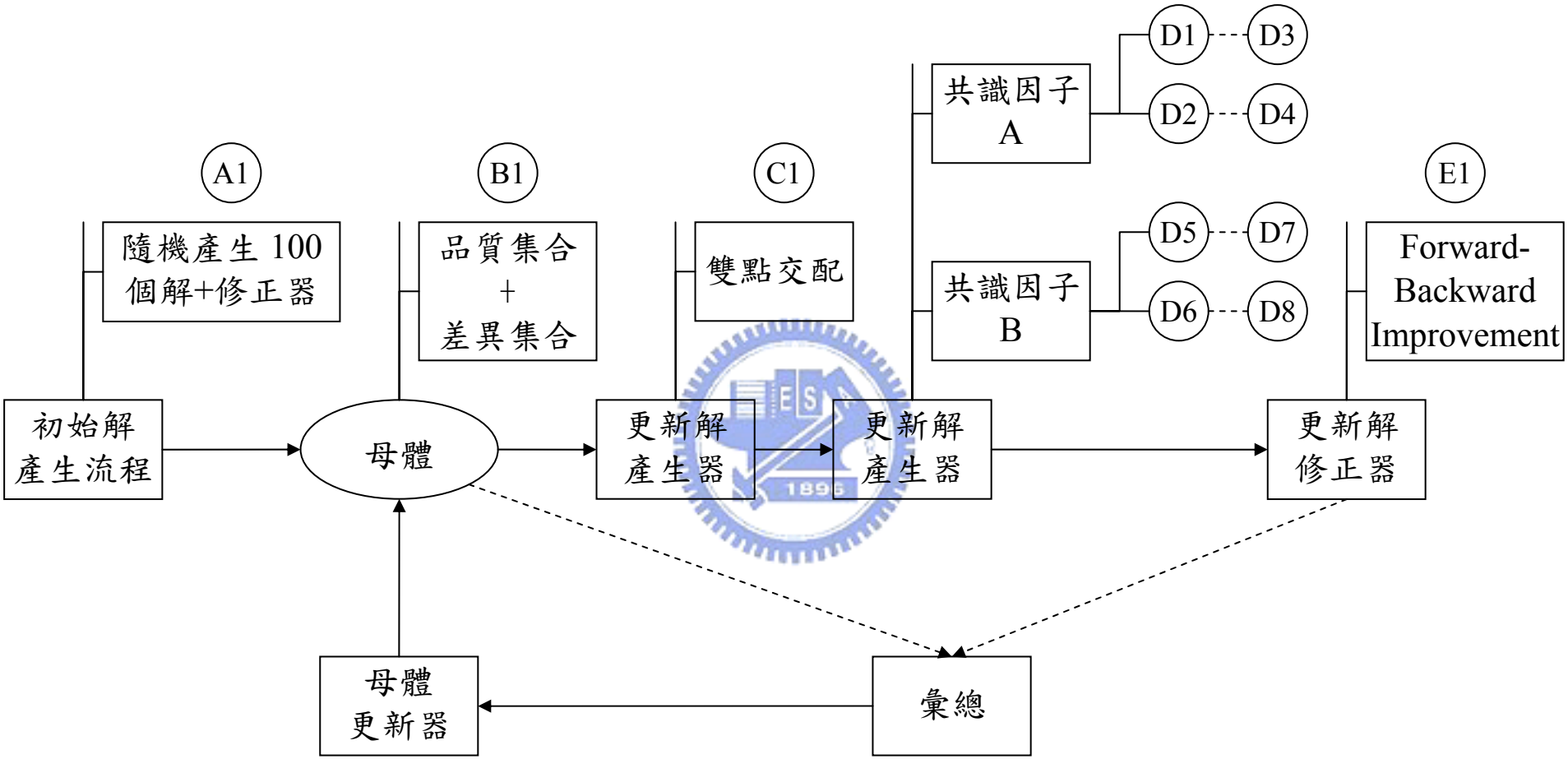

A1
隨機產生 100
個解+修正器
B1
品質集合
+
差異集合
C1
雙點交配
共識因子
A
D1
D3
D2
D4
共識因子
B
D5
D7
D6
D8
E1
Forward-
Backward
Improvement
初始解
產生流程
母體
更新解
產生器
更新解
產生器
更新解
修正器
母體
更新器
彙總

圖 4.1 綜合實驗架構圖

為了方便檢閱，將實驗結果分成表 4.4 和表 4.5，J30 和 J60 是合計 480 個例子的總完工時間，J120 則是 600 個例子。從表 4.4 的數據上來看，共識因子 A 在 1,000 SRK 都有接近或稍勝 V0 的收斂能力，但是到最後的 5,000 和 50,000 SRK 都無法再收斂至更好的解。而 V40(原 V4)放在綜合實驗的用意，是希望測試能否在 J30-50,000 SRK 表現的比 V0 好，但結果無法再繼續收斂，因為在 J60 和 J120 表現太差，所以在 50,000 SRK 的部分便不再進行實驗(同表 4.5)。

表 4.4 共識因子 A 實驗結果(合計總完工時間)

情境和停止條件		V0	V39 (V1)	V40 (V4)	V41 (V5)	V42 (V7)
J30	1,000 SRK	***28,410***	28,402	28,396	28,404	28,410
	5,000 SRK	***28,355***	28,358	28,353	28,360	28,362
	50,000 SRK	***28,319***	28,327	28,326	28,325	28,327
J60	1,000 SRK	***38,765***	38,785	38,810	38,765	38,772
	5,000 SRK	***38,554***	38,641	38,636	38,614	38,619
	50,000 SRK	***38,420***	38,427	—	38,467	38,470
J120	1,000 SRK	***76,736***	76,756	76,946	76,720	76,766
	5,000 SRK	***75,537***	75,968	76,217	75,926	75,977
	50,000 SRK	***74,671***	75,128	—	75,088	75,143

表 4.5 共識因子 B 實驗結果(合計總完工時間)

情境和停止條件		V0	V43	V44	V45	V46
J30	1,000 SRK	***28,410***	28,399	28,402	28,407	28,408
	5,000 SRK	***28,355***	28,367	28,361	28,361	28,367
	50,000 SRK	***28,319***	28,332	28,332	28,331	28,331
J60	1,000 SRK	***38,765***	38,795	38,804	38,776	38,793
	5,000 SRK	***38,554***	38,605	38,622	38,610	38,611
	50,000 SRK	***38,420***	38,465	—	38,475	38,412
J120	1,000 SRK	***76,736***	76,763	76,947	76,725	76,824
	5,000 SRK	***75,537***	75,834	76,078	75,788	75,884
	50,000 SRK	***74,671***	74,986	—	74,951	75,020

表 4.5 是以共識因子 B，從數據上可以明顯的看出，共識因子 B 在 J30-1,000 SRK 大部分的表現都比 V0 好一些，而 V45 也能在 J120-1,000 SRK 有稍好的數據，但仍然在 5,000 和 50,000 SRK 無法比 V0 更好。

總結實驗結果，共識矩陣的抽樣方法在不同的"情境"和"停止條件"下會有不同的求解表現，參數設定似乎對抽樣方法和情境的變化較不敏感，跟 V0 比較起來，共識因子似乎在 J30 和 J120-1,000 SRK 能較快收斂到不錯的解，並且在 J30 和 J60 的其他停止條件也不會差太多。最後共識因子 A 以 V41 為最佳版本；共識因子 B 以 V45 為最佳版本。最佳參數則是 Linear weighting 和 $\alpha = 1$。

4.2.2 與過去文獻比較結果(Step 2)

本節將 V41 和 V45 跟 Debels et al. (2006)裡的其他文獻比較求解績效(如表 4.6)，其中 Hartmann (1998)的實驗資料在 Valls et al. (2003)文獻中。

跟 Hartmann (1998)和 Nonobe & Ibaraki (2002)的比較結果，J30 情境中，V41 和 V45 在 50,000 SRK 的品質都比較好，求解時間則是只有 V45 最好，V41 比 Hartmann (1998)差一些；J60 情境的結果，V41 和 V45 在 50,000 SRK 的品質都比較好，求解時間則是 V41 最好，接著是 Hartmann (1998)，最後是 V45；最後在 J120 情境，V41 和 V45 不論在 5,000 或 50,000 SRK 的求解品質都比較好，求解時間在 5,000 或 50,000 SRK 也同樣的快上許多。

跟 Valls et al. (2003b)的比較結果，J30 情境中，V41 和 V45 在 50,000 SRK 的品質都比較好，求解時間則是比較慢；J60 和 J120 情境的結果相同，V41 和 V45 不論在 5,000 或 50,000 SRK 的求解品質都好很多，求解時間則只有 5,000 SRK 比較好。

跟 Valls et al. (2004)的結果是過去文獻中表現最好的一個。J30 情境中，V41 和 V45 在 50,000 SRK 的品質都比較好(5,000 SRK 無差異)，求解時間則是比較慢；J60 情境中，V41 和 V45 在 50,000 SRK 的求解品質都好很多，求解時間則

比較差；最後的 J120 情境，只有 V45 在 50,000 SRK 的求解績效比較好，V41 則比較差，排名第 3，求解時間則是 Valls et al. (2004)比較快。

綜合以上實驗結果，V41 和 V45 幾乎在每個情境中的表現都是最好的，但從整體來看，V45 是唯一在 J120 也有最好的表現(比其他文獻都好)，V41 的績效則只有在 J30 和 J60 的情境裡有較佳的表現。

表 4.6 與過去文獻比較結果

情境	作者	合計總完工時間	平均 CPU 時間	排名
J30	Hartmann (1998)	28,349	3.95	5
	Nonobe & Ibaraki (2002)	28,337	9.07	4
	Valls et al. (2003b)	28,335	1.61	3
	Valls et al. (2004)	28,361	0.38	7
	本研究 V41 [5,000 SRK]	28,360	0.42	6
	本研究 V41 [50,000 SRK]	28,325	4.75	1
	本研究 V45 [5,000 SRK]	28,361	0.22	7
	本研究 V45 [50,000 SRK]	28,331	2.42	2
J60	Hartmann (1998)	38,660	8.06	6
	Nonobe & Ibaraki (2002)	38,697	26.49	8
	Valls et al. (2003b)	38,671	2.76	7
	Valls et al. (2004)	38,572	1.14	3
	本研究 V41 [5,000 SRK]	38,614	0.56	5
	本研究 V41 [50,000 SRK]	38,467	5.64	1
	本研究 V45 [5,000 SRK]	38,610	1.30	4
	本研究 V45 [50,000 SRK]	38,475	11.35	2
J120	Hartmann (1998)	76,924	48.86	8
	Nonobe & Ibaraki (2002)	76,600	245.33	7
	Valls et al. (2003b)	76,356	17.00	6
	Valls et al. (2004)	75,009	14.52	2
	本研究 V41 [5,000 SRK]	75,926	2.73	5
	本研究 V41 [50,000 SRK]	75,088	22.73	3
	本研究 V45 [5,000 SRK]	75,788	2.27	4
	本研究 V45 [50,000 SRK]	74,951	23.79	1

4.2.3 分析最佳版本(Step 3)

從 4.2.2 的實驗結果得知，共識因子 A 只有在 J30 和 J60 比過去文獻好，共 B 則反而能在 J120 有最好的表現，為此要探討兩者在結構上的差異。

共識樣本應結合"品質"、"差異化"和"適量樣本數"這 3 種特性，才能產生好的解。所謂"品質"代表總完工時間的好壞、"差異化"的目的是希望高品質的樣本，其相互之間的排序是有差異的、而"抽樣數量"是希望抽取適量的樣本，當中能包含許多有差異化的高品質樣本，所以抽樣數量會影響差異化，而差異化會影響抽樣的品質好壞。

表 4.7 為表 4.4 和 4.5 實驗數據的平均值與抽樣數目對照表。J30 情境因為是小問題，很可能在初始解形成母體後就找到最佳解，所以就算共識因子 A 在 1,000 SRK 和 5,000 SRK 的抽樣少，但母體包含了品質和差異化，所以求解品質依然很好；但在 50,000 SRK 中，因為演化世代增加後容易收斂，所以共識因子 B 則可能因為抽樣數目較少，並且因為只抽取"母體+新解"中品質最好的解，使得樣本差異化較低而變成局部搜尋；反觀共識因子 A 的樣本增加後，差異化也變高，較不容易變成局部搜尋，使得解比較好。

J60 情境是中小問題，問題較 J30 複雜也稍難找到最佳解，共識因子 B 在 1,000 SRK 因為抽樣數目較多，所以可能包含了不好的排序結果，短時間內收斂不到好的解，但隨著停止條件上升演化世代變多，加上 J60 也不如 J30 可以輕易找到最佳解，所以共識因子 B 就可以從"母體+新解"中挑出有品質的樣本，再加上差異化的特性就可隨著世代演化使求解品質提升。

最後在 J120 情境的大問題中，由於產生不好的解的可能性更大，所以在 1,000 SRK 使用較大的樣本就會將不好的排序放入共識矩陣中，但同樣隨著停止條件上升至 5,000 或 50,000 SRK 後，共識因子 B 樣本結構的優勢就更明顯了。

表 4.7 共識因子實驗數據和抽樣數目對照表

情境和停止條件		共識因子 A		共識因子 B	
		表 4.4 平均值	抽樣數目	表 4.5 平均值	抽樣數目
J30	1,000 SRK	28,403	8	28,404	28
	5,000 SRK	28,358	15	28,364	28
	50,000 SRK	28,326	44	28,332	28
J60	1,000 SRK	38,783	8	38,792	28
	5,000 SRK	38,628	15	38,612	28
	50,000 SRK	38,455	44	38,451	28
J120	1,000 SRK	76,797	8	76,815	28
	5,000 SRK	76,022	15	75,896	28
	50,000 SRK	75,120	44	74,986	28

從兩者在 3 種情境的比較可以發現，共識因子 A 的抽樣隨著停止條件的增加而變大(8→44)，共識因子 B 則是一直維持 28 個，但樣本品質較好。因為 J30 的問題複雜度還不夠高，大部分的最佳解都還是可以被找到，共識因子 B 的抽樣結構優勢還不明顯；但是在 J60 和 J120 情境中，因為問題變得複雜，要找到好的解成為樣本不容易，所以共識因子 B 抽樣結構反而變成優點。因此共識矩陣在抽樣時應該視情境和演化世代，除了盡可能抽取品質高的樣本以外，也應該考慮”差異化”和”抽樣數量”的參數調整。

4.2.4 改善最佳版本(Step 4)

依上述的分析結果將共識因子 B 的 V45 加以修改，同樣以母體+新解為樣本，抽樣數量改成母體大小，並且在抽樣時加入分散搜尋法 t_1 和 t_2 的概念，成為新版本「V47」。V45 和 V47 的實驗結果如表 4.8 所示，考慮”差異化”和”抽樣數量”之後，V47 的整體績效比 V45 來得好，以結構來判斷 V47 較差的部分如 J30-1,000SRK，原因在於使用雖然是抽品質最好的解，但因為抽樣數量不夠多，相形之下 t_1 和 t_2 的差異化就太大，所以不能形成好的共識；而 J120-1,000SRK 則是因為抽樣數量較多，所以包含了較差的排序，不容易產生很好的解。

表 4.8 共識因子 B 版本 V45 和 V47 結果對照

情境和停止條件		V45	抽樣數量	V47	抽樣數量
J30	1,000 SRK	***28,407***	28	28,417	8
	5,000 SRK	***28,361***	28	28,360	15
	50,000 SRK	***28,331***	28	28,327	44
J60	1,000 SRK	***38,776***	28	38,773	8
	5,000 SRK	***38,610***	28	38,603	15
	50,000 SRK	***38,475***	28	38,464	44
J120	1,000 SRK	***76,725***	28	76,605	8
	5,000 SRK	***75,788***	28	75,783	15
	50,000 SRK	***74,951***	28	74,993	44

表 4.9 共識樣本於不同情境及停止條件的參數調整

情境	1,000SRK	5,000SRK	50,000SRK
J30	抽樣少	**控制**差異化 **提升**抽樣數量	**控制**差異化 **提升**抽樣數量
J60	**控制**差異化 抽樣少	**控制**差異化 **提升**抽樣數量	**控制**差異化 **提升**抽樣數量
J120	**控制**差異化 抽樣少	**控制**差異化 **提升**抽樣數量	**控制**差異化 **“微”升**抽樣數量

經過了上述的分析與改善結果得知，共識樣本品質好，不一定能找到近似最佳解，還需配合差異化及抽樣數量的參數調整，所以以 V47 的抽樣結構為範本，表 4.9 針對上述的分析結果整理出在不同情境和停止條件之下，應如何調整共識因子的樣本。其中，因為抽樣時都是選取品質最好的樣本，所以不列在表中，而提升差異化的方式有兩種：第一，以 t_1 和 t_2 來控制可得到較大的差異化；第二，提升抽樣數量也可增加一些差異化。

然而，表 4.9 只針對 PSPLIB 測試調整共識因子的參數，但應用演算法於實際的問題時，可能無法對問題的情境做適當的判斷，所以需要找出一種真正的規則，但是在尋找規則時會碰上一些困難，舉例來說：在差異化和抽樣數量固定下，品質愈高愈好；在品質和抽樣數量固定下，差異化在本研究使用 Debels et al. (2006)設定 t_1 和 t_2 的參數較好；但是在品質和差異化固定時，卻無法衡量出抽樣數量的設定，因為先前提到當抽樣數量改變時，差異化和品質都會被改變。因此，若需要為演算法找出一種抽樣的規則，仍需要做相關的實驗和分析才能得到結果。

另外，分析產生新解的部分，共識因子雖然可以產生出平均品質較高的新解，但共識比率決定一切，樣本結構經演化後容易變成局部搜尋，無法像雙點交配能有突破性的排序變化(無法跳出局部搜尋的窘境)。並且即使在演化過程中尋找到品質很好的解，也會因為跟其他解的共識合併後，使品質好的解的排序意見被中和，無法針對好的排序進行區域性的搜尋，因此不容易有更好的求解品質。

第五章　結論和未來研究方向

5.1 結論

本研究目的，主要是應用共識因子和田口方法在有限資源的專案排程問題，並且跟 Debels et al. (2006)的演算法做組合，與過去的演算法比較。

實驗結果方面，V41 和 V45 只能在 22%的情境(J30 和 J120-1,000SRK)下表現較好比 Debels et al. (2006)好，而 V41 也能在 22%的情境(J30-5,000SRK 和 J60-1,000SRK)有相近的結果，其他 56%情境的比較結果卻比較差。而 V45 和 V47 跟 Debels et al. (2006)文獻內其他標竿方法：Valls et al. (2004)、Valls et al. (2003b)、Nonobe & Ibaraki (2002)和 Hartmann (1998)，比較的結果則是在各種情境下都表現最好。

分析結果發現，分散搜尋法設計的母體結構對求解品質有相當好的表現，因為透過 t_1 和 t_2 兩種指標的比較，就可以讓母體同時擁有品質和差異化兩種特性，並且計算所需的 CPU 時間幾乎可以忽略。更新解的部分，田口方法因為在更新過程會產生許多新解，所以如果不改善這個缺點就不適合使用在本問題的演算法架構中。

在共識因子的實驗結果發現，和 Debels et al. (2006)比較之下，共識因子能在 1,000SRK 有較好的收斂品質，並且整體績效比其他過去著名演算法還好，產生新解平均的品質也較高；並且在分析結果發現抽樣結構是提升求解品質的關鍵，只有好的樣本品質不一定能求得近似最佳解。因此共識矩陣在抽樣時應該視情境和演化世代，除了盡可能抽取品質高的樣本以外，也應該考慮”差異化”和”抽樣數量”的參數調整。

5.2 未來研究方向

經過了導引實驗和綜合實驗的分析後，在此整理分析結果對各種方法提出改善方向如下：

(1) 初始解

原本的初始解是隨機產生RK(不合理解)後以逐步排程法求解得到SRK(合理解)，若能先行考慮專案問題的執行順序限制，只針對沒有執行順序限制的任務來指定RK，就能一開始就產生出合理解，可能使初始解品質更好。

(3) 共識因子

依4.2.3和4.2.4的分析結果來看，本研究在控制差異化的部分只使用 t_1 和 t_2 做設定，還可以再做其他種方法的嘗試；抽樣數量的部分只使用母體大小或是抽固定數量，然而抽樣數量對差異化和品質的影響這部分還未做詳細的實驗和分析，是未來需加以研究的部分。

另外，由於共識矩陣是更新的關鍵，所以可以在連續幾代沒有更新最佳解時，重新調整共識矩陣的樣本。產生新解時，也可以結合田口方法的概念，以田口直交表結合共識矩陣中非0或1的比率來更新，以提高求解績效。

(4) 更新解修正器

Alcaraz & Maroto (2001)在使用修正器時，在染色體中多加入一個基因來判斷該SRK是否進行修正，或是只修正一半，同樣是本研究可以考慮的改善方向。

參考文獻

吳政翰，“DHC 系統之任務指派與排程的新型基因算法”，國立交通大學工業工程與管理學系碩士論文，2006。

蘇朝墩，“品質工程”，中華民國品質學會，2002。

黎正中，李家琪，“田口方法的應用與發展”，品質學報，第 7 卷第 2 期，頁 117-137，2000。

Alcaraz J. and Maroto C., “A robust genetic algorithm for resource allocation in project scheduling,” *Annals of Operations Research*, 102, pp. 83-109, 2001.

Alcaraz J., Maroto C. and Ruiz R., “Improving the performance of genetic algorithms for the RCPS problem,” in Proceedings of the Ninth International Workshop on Project Management and Scheduling, Nancy, pp. 40-43, 2004.

Birbil S.I. and Fang S.C., “An electromagnetism-like mechanism for global optimization,” *Journal of Global Optimization*, Vol. 25, pp. 263-282, 2003.

Bouleimen K., Lecocq H., “A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple modes version,” *European Journal of the Operational Research*, Vol. 149, pp. 268-281, 2006.

Brucker P., Drexl A., Möhring R., Neumann K. and Pesch E., “Resource-constrained project scheduling: Notation, classification, models and methods,” *European Journal of the Operational Research*, Vol. 112, pp. 3-41, 1999.

Brucker P., Knust S., Schoo A. and Thiele O., “A branch & bound algorithm for the resource-constrained project scheduling problem,” *European Journal of the Operational Research*, Vol. 107, pp. 272-288, 1998.

Cho J.H. and Kim Y.D., “A simulated annealing algorithm for resource constrained project scheduling problems,” *Journal of the Operational Research Society*, Vol. 48, pp. 736-744, 1997.

Debels D., Reych B.D., Leus R. and Vanhoucke M., “A hybrid scatter search/electromagnetism meta-heuristic for project scheduling,” *European Journal of the Operational Research*, Vol. 169, pp. 638-653, 2006.

Demeulemeester E. and Herroelen W., “A branch-and-bound procedure for the multiple resource-constrained project scheduling problem,” *Management Science*, Vol. 38, pp. 1803-1818, 1992.

Demeulemeester E. and Herroelen W., "New benchmark results for the resource-constrained project scheduling problem," *Management Science*, Vol. 43, pp. 1458-1492, 1997.

Glover F., "A template for scatter search and path relinking," Artificial Evolution Lecture Notes In Computer Science, Vol. 1363, pp. 3-51, 1998.

Glover F., Laguna M. and Martí R., "Fundamental of scatter search and path relinking," *Control and Cybernetics*, Vol. 39, pp. 653-684, 2000.

Glover F., Laguna M. and Martí R., forthcoming, "Scatter search," in: Ghosh, A., Tsutsui S. (Eds.), *Theory and Applications of Evolutionary Computation: Recent Trends*, Springer-Verlag, New York.

Hartmann S., "A competitive genetic algorithm for the resource-constrained project scheduling," *Naval Research Logistics*, Vol. 45, pp. 733-750, 1998.

Hartmann S., "A self-adapting genetic algorithm for project scheduling under resource constraints," *Naval Research Logistics*, Vol. 49, pp. 433-448, 2002.

Hartmann S. and Kolisch R., "Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem," *European Journal of the Operational Research*, Vol. 127, pp. 394-407, 2000.

Herroelen W., "Project scheduling-Theory and Practice," *Productions and Operations Management*, Vol. 14, No. 4, pp. 413-432, 2005.

Herroelen W., Demeulemeester E. and De Reyck B., "A classification scheme for project scheduling," in Weglarz J.(Ed.), *Project Scheduling — Recent Model, Algorithms and Applications*, *International Series in Operations Research and Management Science*, Vol 14, Kluwer Academic Publishers, Dordrecht, pp. 77-106(Chapter 1), 1998a.

Herroelen W., De Reyck B. and Demeulemeester E., "Resource-constrained project scheduling: A survey of recent development," *Computers and Operations Research*, Vol. 25, No. 4, pp. 279-302, 1998b.

Kelley J.E. Jr., "The critical-path method: Resources planning and scheduling," in J.F. Muth and G.L. Thompson (Eds.), *Industrial Scheduling*, Prentice-Hall, New Jersey, pp. 347-365, 1963.

Kochetov Y. and Stolyar A., "Evolutionary local search with variable neighborhood for the resource constrained project scheduling problem," in Proceedings of the 3th International Workshop of Computer Science and Information Technologies, Russia, 2003.

Kolisch R.., "Serial and Parallel resource-constrained project scheduling methods revisited: Theory and computation," *European Journal of the Operational Research*, Vol. 90, pp. 320-333, 1996.

Kolisch R. and Hartmann S., "Experimental investigation of heuristics for resource-constrained project scheduling: An update," *European Journal of the Operational Research*, Vol. 174, pp. 23-27, 2006.

Kolisch R. and Sprecher A., "PSPLIB－A project scheduling library," *European Journal of the Operational Research*, Vol. 96, pp. 250-216, 1996.

Kolisch R., Sprecher A. and Drexl A., "Characterization and Generation of a General Class of Resource-constrained Project Scheduling Problems," *Management Science*, Vol. 41, No. 10, pp. 1693-1703, 1995.

Li K.Y. and Willis R.J., "An iterative scheduling technique for resource-constrained project scheduling," *European Journal of the Operational Research*, Vol. 56, pp. 370-379, 1992.

Martí R., Laguna M. and Glover F., "Principles of scatter search," *European Journal of the Operational Research*, Vol. 169, pp. 359-372, 2006.

Mingozzi A., Maniezzo V., Ricciardelli S. and Bianco L., "An exact algorithm for the resource-constrained project scheduling problem based on a new mathematical formulation," *Management Science*, Vol. 44, pp. 715-729, 1998.

Nonobe K. and Ibaraki T., "Formulation and tabu search algorithm for the resource constrained project scheduling problem (RCPSP)," in: Ribeiro C.C., Hansen P. (Eds.), *Essays and Surveys in Metaheuristics*. Kluwer Academic Publishers, Boston, pp. 557-588, 2002.

Őzdamar L. and Ulusoy G., "A note on an iterative forward / backward scheduling technique with reference to a procedure by Li and Willis," *European Journal of the Operational Research*, Vol. 89, pp. 400-407, 1996.

Sprecher A., "Scheduling resource-constrained projects competitively at modest resource requirements," *Management Science*, Vol. 46, pp. 710-723, 2000.

Taguchi G., "Introduction to Quality Engineering: Designing Quality into products and Processes," Tokyo: Asian Productivity Organization, 1986.

Taguchi G., "System of Experimental Design," White Plains, New York: UNIPUB / Krauss International, 1987.

Valls V., Ballestin F. and Quintanilla M.S., "A hybrid genetic algorithm for RCPSP," Technical report, Department of Statistics and Operations Research, University of Valencia, 2003a.

Valls V., Ballestin F. and Quintanilla M.S., "A population-based approach to the resource-constrained project scheduling problem," *Annals of Operational Research*, Vol. 131, pp. 305-324, 2004.

Valls V., Ballestin F. and Quintanilla M.S., "Justification and RCPSP: A technique that pays," *European Journal of the Operational Research*, Vol. 96, pp. 375-386, 2005.

Valls V., Quintanilla M.S. and Ballestin F., "Resource-constrained project scheduling: Acritical reordering heuristic," *European Journal of the Operational Research*, Vol. 149, pp. 282-301, 2003b.

Zhang H., Heng L. and Tam C.M., "Particle swarm optimization for resource-constrained project scheduling," *International Journal of Project Management*, Vol. 24, pp. 83-92, 2006.

附錄

本附錄附上第四章所有實驗的詳細資料，包括總完工時間 m_s 的加總、Avg. Dev. from best solutions(%)以及 480 或 600 個例子中，可以找到多少個最佳解。每張表的左邊代表 Debels et al. (2006)的實驗數據，右邊是本研究版本的數據，因為 Debels et al. (2006)在 J60 和 J120 比較的是 PSPLIB 在 2003 年提供的最佳解，而本研究是使用 2007 年的最佳解來比，為免混淆，所以 Debels et al. (2006)在 Avg. Dev. Best 和共有多少問題找到最佳解的部分不放入，以”-”標示。

Debels et al., 2006(V0) v.s. V1(V39)							
項目	停止條件	J30 (480 instances)		J60 (480 instances)		J120 (600 instances)	
Makespan 加總	1000	***28,410***	28,402	***38,765***	38,758	***76,736***	76,756
	5000	***28,355***	28,358	***38,554***	38,614	***75,537***	75,968
	50,000	***28,319***	28,327	***38,420***	38,467	***74,671***	75,126
	PSPLIB Best Sum	***28,316***		***38,316***		***73,766***	
Avg. Dev. Best	1000	***0.27%***	0.24%	-	0.96%	-	3.68%
	5000	***0.11%***	0.12%	-	0.61%	-	2.79%
	50,000	***0.01%***	0.03%	-	0.31%	-	1.87%
共有幾個問題找到最佳解	1000	***421***	429	-	357	-	194
	5000	***451***	449	-	367	-	212
	50,000	***477***	468	-	398	-	233

Debels et al., 2006(V0) v.s. V2							
項目	停止條件	J30 (480 instances)		J60 (480 instances)		J120 (600 instances)	
Makespan 加總	1000	***28,410***	28,409	***38,765***	38,799	***76,736***	-
	5000	***28,355***	28,362	***38,554***	38,635	***75,537***	-
	50,000	***28,319***	-	***38,420***	-	***74,671***	-
	PSPLIB Best Sum	***28,316***		***38,316***		***73,766***	
Avg. Dev. Best	1000	***0.27%***	0.25%	-	0.99%	-	-
	5000	***0.11%***	0.13%	-	0.65%	-	-
	50,000	***0.01%***	-	-	-	-	-
共有幾個問題找到最佳解	1000	***421***	429	-	357	-	-
	5000	***451***	445	-	365	-	-
	50,000	***477***	-	-	-	-	-

Debels et al., 2006(V0) v.s. V3							
項目	停止條件	J30 (480 instances)		J60 (480 instances)		J120 (600 instances)	
Makespan 加總	1000	***28,410***	28,407	***38,765***	38,788	***76,736***	-
	5000	***28,355***	28,358	***38,554***	38,623	***75,537***	-
	50,000	***28,319***	-	***38,420***	-	***74,671***	-
	PSPLIB Best Sum	***28,316***		***38,316***		***73,766***	
Avg. Dev. Best	1000	***0.27%***	0.25%	-	0.97%	-	-
	5000	***0.11%***	0.12%	-	0.63%	-	-
	50,000	***0.01%***	-	-	-	-	-
共有幾個問題找到最佳解	1000	***421***	423	-	355	-	-
	5000	***451***	446	-	369	-	-
	50,000	***477***	-	-	-	-	-

Debels et al., 2006(V0) v.s. V4(V40)							
項目	停止條件	J30 (480 instances)		J60 (480 instances)		J120 (600 instances)	
Makespan 加總	1000	***28,410***	28,396	***38,765***	38,810	***76,736***	76,946
	5000	***28,355***	28,353	***38,554***	38,363	***75,537***	76,217
	50,000	***28,319***	28,326	***38,420***	-	***74,671***	-
	PSPLIB Best Sum	***28,316***		***38,316***		***73,766***	
Avg. Dev. Best	1000	***0.27%***	0.22%	-	1.00%	-	3.89%
	5000	***0.11%***	0.10%	-	0.65%	-	3.06%
	50,000	***0.01%***	0.03%	-	-	-	-
共有幾個問題找到最佳解	1000	***421***	427	-	354	-	196
	5000	***451***	444	-	358	-	212
	50,000	***477***	461	-	-	-	-

Debels et al., 2006(V0) v.s. V5(V41)							
項目	停止條件	J30 (480 instances)		J60 (480 instances)		J120 (600 instances)	
Makespan 加總	1000	***28,410***	28,404	***38,765***	38,365	***76,736***	76,720
	5000	***28,355***	28,360	***38,554***	38,614	***75,537***	75,926
	50,000	***28,319***	28,326	***38,420***	38,467	***74,671***	75,088
	PSPLIB Best Sum	***28,316***		***38,316***		***73,766***	
Avg. Dev. Best	1000	***0.27%***	0.24%	-	0.92%	-	3.24%
	5000	***0.11%***	0.12%	-	0.61%	-	2.74%
	50,000	***0.01%***	0.03%	-	0.32%	-	1.82%
共有幾個問題找到最佳解	1000	***421***	424	-	361	-	193
	5000	***451***	448	-	367	-	212
	50,000	***477***	469	-	397	-	237

Debels et al., 2006(V0) v.s. V6							
項目	停止條件	J30 (480 instances)		J60 (480 instances)		J120 (600 instances)	
Makespan 加總	1000	***28,410***	28,413	***38,765***	38,797	***76,736***	-
	5000	***28,355***	28,363	***38,554***	38,618	***75,537***	-
	50,000	***28,319***	-	***38,420***	-	***74,671***	-
	PSPLIB Best Sum	***28,316***		***38,316***		***73,766***	
Avg. Dev. Best	1000	***0.27%***	0.27%	-	0.99%	-	-
	5000	***0.11%***	0.13%	-	0.62%	-	-
	50,000	***0.01%***	-	-	-	-	-
共有幾個問題找到最佳解	1000	***421***	416	-	348	-	-
	5000	***451***	428	-	357	-	-
	50,000	***477***	-	-	-	-	-

Debels et al., 2006(V0) v.s. V7(V42)							
項目	停止條件	J30 (480 instances)		J60 (480 instances)		J120 (600 instances)	
Makespan 加總	1000	***28,410***	28,410	***38,765***	38,772	***76,736***	76,766
	5000	***28,355***	28,362	***38,554***	38,619	***75,537***	75,977
	50,000	***28,319***	28,327	***38,420***	38,470	***74,671***	75,143
	PSPLIB Best Sum	***28,316***		***38,316***		***73,766***	
Avg. Dev. Best	1000	***0.27%***	0.26%	-	0.93%	-	3.69%
	5000	***0.11%***	0.13%	-	0.62%	-	2.80%
	50,000	***0.01%***	0.03%	-	0.32%	-	1.88%
共有幾個問題找到最佳解	1000	***421***	414	-	356	-	193
	5000	***451***	433	-	358	-	211
	50,000	***477***	457	-	376	-	233

Debels et al., 2006(V0) v.s. V8							
項目	停止條件	J30 (480 instances)		J60 (480 instances)		J120 (600 instances)	
Makespan 加總	1000	***28,410***	28,407	***38,765***	38,792	***76,736***	-
	5000	***28,355***	28,358	***38,554***	38,630	***75,537***	-
	50,000	***28,319***	-	***38,420***	-	***74,671***	-
	PSPLIB Best Sum	***28,316***		***38,316***		***73,766***	
Avg. Dev. Best	1000	***0.27%***	0.25%	-	0.97%	-	-
	5000	***0.11%***	0.11%	-	0.64%	-	-
	50,000	***0.01%***	-	-	-	-	-
共有幾個問題找到最佳解	1000	***421***	416	-	353	-	-
	5000	***451***	431	-	360	-	-
	50,000	***477***	-	-	-	-	-

Debels et al., 2006(V0) v.s. V9							
項目	停止條件	J30 (480 instances)		J60 (480 instances)		J120 (600 instances)	
Makespan 加總	1000	***28,410***	28,420	***38,765***	38,756	***76,736***	-
	5000	***28,355***	28,360	***38,554***	38,614	***75,537***	-
	50,000	***28,319***	-	***38,420***	-	***74,671***	-
	PSPLIB Best Sum	***28,316***		***38,316***		***73,766***	
Avg. Dev. Best	1000	***0.27%***	0.28%	-	0.94%	-	-
	5000	***0.11%***	0.12%	-	0.61%	-	-
	50,000	***0.01%***	-	-	-	-	-
共有幾個問題找到最佳解	1000	***421***	424	-	357	-	-
	5000	***451***	447	-	368	-	-
	50,000	***477***	-	-	-	-	-

Debels et al., 2006(V0) v.s. V10							
項目	停止條件	J30 (480 instances)		J60 (480 instances)		J120 (600 instances)	
Makespan 加總	1000	***28,410***	28,412	***38,765***	38,788	***76,736***	-
	5000	***28,355***	28,357	***38,554***	38,623	***75,537***	-
	50,000	***28,319***	-	***38,420***	-	***74,671***	-
	PSPLIB Best Sum	***28,316***		***38,316***		***73,766***	
Avg. Dev. Best	1000	***0.27%***	0.27%	-	0.97%	-	-
	5000	***0.11%***	0.11%	-	0.63%	-	-
	50,000	***0.01%***	-	-	-	-	-
共有幾個問題找到最佳解	1000	***421***	416	-	354	-	-
	5000	***451***	452	-	370	-	-
	50,000	***477***	-	-	-	-	-

Debels et al., 2006(V0) v.s. V11							
項目	停止條件	J30 (480 instances)		J60 (480 instances)		J120 (600 instances)	
Makespan 加總	1000	***28,410***	28,406	***38,765***	38,789	***76,736***	-
	5000	***28,355***	28,358	***38,554***	38,620	***75,537***	-
	50,000	***28,319***	-	***38,420***	-	***74,671***	-
	PSPLIB Best Sum	***28,316***		***38,316***		***73,766***	
Avg. Dev. Best	1000	***0.27%***	0.25%	-	0.97%	-	-
	5000	***0.11%***	0.11%	-	0.62%	-	-
	50,000	***0.01%***	-	-	-	-	-
共有幾個問題找到最佳解	1000	***421***	422	-	353	-	-
	5000	***451***	448	-	372	-	-
	50,000	***477***	-	-	-	-	-

Debels et al., 2006(V0) v.s. V12							
項目	停止條件	J30 (480 instances)		J60 (480 instances)		J120 (600 instances)	
Makespan 加總	1000	***28,410***	28,413	***38,765***	38,792	***76,736***	-
	5000	***28,355***	28,357	***38,554***	38,628	***75,537***	-
	50,000	***28,319***	-	***38,420***	-	***74,671***	-
	PSPLIB Best Sum	***28,316***		***38,316***		***73,766***	
Avg. Dev. Best	1000	***0.27%***	0.27%	-	0.98%	-	-
	5000	***0.11%***	0.11%	-	0.64%	-	-
	50,000	***0.01%***	-	-	-	-	-
共有幾個問題找到最佳解	1000	***421***	410	-	353	-	-
	5000	***451***	436	-	358	-	-
	50,000	***477***	-	-	-	-	-

Debels et al., 2006(V0) v.s. V13							
項目	停止條件	J30 (480 instances)		J60 (480 instances)		J120 (600 instances)	
Makespan 加總	1000	***28,410***	28,470	***38,765***	39,100	***76,736***	-
	5000	***28,355***	28,413	***38,554***	38,964	***75,537***	-
	50,000	***28,319***	-	***38,420***	-	***74,671***	-
	PSPLIB Best Sum	***28,316***		***38,316***		***73,766***	
Avg. Dev. Best	1000	***0.27%***	0.43%	-	1.61%	-	-
	5000	***0.11%***	0.27%	-	1.32%	-	-
	50,000	***0.01%***	-	-	-	-	-
共有幾個問題找到最佳解	1000	***421***	403	-	349	-	-
	5000	***451***	419	-	354	-	-
	50,000	***477***	-	-	-	-	-

Debels et al., 2006(V0) v.s. V18							
項目	停止條件	J30 (480 instances)		J60 (480 instances)		J120 (600 instances)	
Makespan 加總	1000	***28,410***	28,408	***38,765***	38,803	***76,736***	-
	5000	***28,355***	28,364	***38,554***	38,660	***75,537***	-
	50,000	***28,319***	-	***38,420***	-	***74,671***	-
	PSPLIB Best Sum	***28,316***		***38,316***		***73,766***	
Avg. Dev. Best	1000	***0.27%***	0.26%	-	1.00%	-	-
	5000	***0.11%***	0.13%	-	0.70%	-	-
	50,000	***0.01%***	-	-	-	-	-
共有幾個問題找到最佳解	1000	***421***	423	-	355	-	-
	5000	***451***	444	-	369	-	-
	50,000	***477***	-	-	-	-	-

Debels et al., 2006(V0) v.s. V26							
項目	停止條件	J30 (480 instances)		J60 (480 instances)		J120 (600 instances)	
Makespan 加總	1000	***28,410***	28,479	***38,765***	39,130	***76,736***	-
	5000	***28,355***	28,414	***38,554***	38,982	***75,537***	-
	50,000	***28,319***	-	***38,420***	-	***74,671***	-
	PSPLIB Best Sum	***28,316***		***38,316***		***73,766***	
Avg. Dev. Best	1000	***0.27%***	0.46%	-	1.67%	-	-
	5000	***0.11%***	0.27%	-	1.35%	-	-
	50,000	***0.01%***	-	-	-	-	-
共有幾個問題找到最佳解	1000	***421***	400	-	351	-	-
	5000	***451***	419	-	354	-	-
	50,000	***477***	-	-	-	-	-

Debels et al., 2006(V0) v.s. V31							
項目	停止條件	J30 (480 instances)		J60 (480 instances)		J120 (600 instances)	
Makespan 加總	1000	***28,410***	28,493	***38,765***	39,163	***76,736***	-
	5000	***28,355***	28,415	***38,554***	38,988	***75,537***	-
	50,000	***28,319***	-	***38,420***	-	***74,671***	-
	PSPLIB Best Sum	***28,316***		***38,316***		***73,766***	
Avg. Dev. Best	1000	***0.27%***	0.50%	-	1.73%	-	-
	5000	***0.11%***	0.28%	-	1.37%	-	-
	50,000	***0.01%***	-	-	-	-	-
共有幾個問題找到最佳解	1000	***421***	398	-	347	-	-
	5000	***451***	417	-	353	-	-
	50,000	***477***	-	-	-	-	-

Debels et al., 2006(V0) v.s. V43							
項目	停止條件	J30 (480 instances)		J60 (480 instances)		J120 (600 instances)	
Makespan 加總	1000	***28,410***	28,399	***38,765***	38,795	***76,736***	76,763
	5000	***28,355***	28,367	***38,554***	38,605	***75,537***	75,864
	50,000	***28,319***	28,332	***38,420***	38,465	***74,671***	74,986
	PSPLIB Best Sum	***28,316***		***38,316***		***73,766***	
Avg. Dev. Best	1000	***0.27%***	0.23%	-	0.98%	-	3.69%
	5000	***0.11%***	0.14%	-	0.60%	-	2.68%
	50,000	***0.01%***	0.04%	-	0.31%	-	1.72%
共有幾個問題找到最佳解	1000	***421***	430	-	357	-	193
	5000	***451***	443	-	370	-	210
	50,000	***477***	469	-	391	-	229

Debels et al., 2006(V0) v.s. V44							
項目	停止條件	J30 (480 instances)		J60 (480 instances)		J120 (600 instances)	
Makespan 加總	1000	***28,410***	28,402	***38,765***	38,804	***76,736***	76,947
	5000	***28,355***	28,361	***38,554***	38,622	***75,537***	76,078
	50,000	***28,319***	28,332	***38,420***	-	***74,671***	-
	PSPLIB Best Sum	***28,316***		***38,316***		***73,766***	
Avg. Dev. Best	1000	***0.27%***	0.24%	-	1.00%	-	3.89%
	5000	***0.11%***	0.13%	-	0.63%	-	2.90%
	50,000	***0.01%***	0.05%	-	-	-	-
共有幾個問題找到最佳解	1000	***421***	425	-	358	-	192
	5000	***451***	447	-	369	-	210
	50,000	***477***	468	-	-	-	-

Debels et al., 2006(V0) v.s. V45							
項目	停止條件	J30 (480 instances)		J60 (480 instances)		J120 (600 instances)	
Makespan 加總	1000	***28,410***	28,407	***38,765***	38,776	***76,736***	76,725
	5000	***28,355***	28,361	***38,554***	38,610	***75,537***	75,788
	50,000	***28,319***	28,331	***38,420***	38,475	***74,671***	74,951
	PSPLIB Best Sum	***28,316***		***38,316***		***73,766***	
Avg. Dev. Best	1000	***0.27%***	0.25%	-	0.94%	-	3.64%
	5000	***0.11%***	0.13%	-	0.61%	-	2.60%
	50,000	***0.01%***	0.04%	-	0.33%	-	1.69%
共有幾個問題找到最佳解	1000	***421***	422	-	362	-	193
	5000	***451***	448	-	368	-	209
	50,000	***477***	468	-	394	-	227

Debels et al., 2006(V0) v.s. V46							
項目	停止條件	J30 (480 instances)		J60 (480 instances)		J120 (600 instances)	
Makespan 加總	1000	***28,410***	28,408	***38,765***	38,793	***76,736***	76,824
	5000	***28,355***	28,367	***38,554***	38,611	***75,537***	75,834
	50,000	***28,319***	28,331	***38,420***	38,472	***74,671***	75,020
	PSPLIB Best Sum	***28,316***		***38,316***		***73,766***	
Avg. Dev. Best	1000	***0.27%***	0.26%	-	0.97%	-	3.76%
	5000	***0.11%***	0.14%	-	0.61%	-	2.71%
	50,000	***0.01%***	0.04%	-	0.33%	-	1.79%
共有幾個問題找到最佳解	1000	***421***	421	-	359	-	193
	5000	***451***	440	-	368	-	208
	50,000	***477***	466	-	389	-	230

Debels et al., 2006(V0) v.s. V47							
項目	停止條件	J30 (480 instances)		J60 (480 instances)		J120 (600 instances)	
Makespan 加總	1000	***28,410***	28,417	***38,765***	38,773	***76,736***	76,605
	5000	***28,355***	28,360	***38,554***	38,603	***75,537***	75,783
	50,000	***28,319***	28,327	***38,420***	38,464	***74,671***	74,993
	PSPLIB Best Sum	***28,316***		***38,316***		***73,766***	
Avg. Dev. Best	1000	***0.27%***	0.29%	-	0.94%	-	3.52%
	5000	***0.11%***	0.12%	-	0.59%	-	2.60%
	50,000	***0.01%***	0.03%	-	0.31%	-	1.72%
共有幾個問題找到最佳解	1000	***421***	414	-	357	-	194
	5000	***451***	448	-	367	-	207
	50,000	***477***	470	-	392	-	233