

國立臺灣大學工學院工業工程學所

碩士論文

Graduate Institute of Industrial Engineering

Collage of Engineering

National Taiwan University

Master Thesis

求解一般性排序優化問題的仿水流優化演算法

Water Flow-like Algorithm for Sequencing Problems



潘嘉琪

Chia-Chi Pan

指導教授：楊烽正 博士

Advisor: Feng-Cheng Yeng, Ph.D.

中華民國 97 年 7 月

July, 2008

誌謝

椰林樹影，傳鐘聲迴，我這裡到達一座求學的里程碑。

沒有楊烽正老師的指導，這本論文無法誕生。老師治學態度的專注和嚴謹，每每讓我感到自己的渺小。在學習的過程中，我看到老師在教學上的熱忱；在完成論文的過程中，我看到老師對做研究的要求。在一次又一次的論文修正中，多虧老師不厭其煩地耐心指導，才能讓這本論文完整而豐富。同時也感謝徐旭昇老師、翁偉泰老師、黃遵鉅老師、陳啟明老師在百忙中審閱這本論文，並以自身淵博的學識給予建議與指教，讓這本論文更加完備。

感謝 YY Group 一路以來一同打拚的夥伴們。于惠、秀惠兩位學姊的帶領，讓我迅速融入新的求學環境，並且有了可以學習的榜樣。從一開始就一起奮鬥到最後的哲杰、俊男、青杉，總是不吝惜給我鼓勵與協助。還有幫忙準備口試的旭芳、傑翔、顯德學弟，有你們在才讓我無後顧之憂。三不五時會來串門子的周 G 成員和陳 G 成員們，在口試前特地為我打氣的、在口試後和我握手道恭喜的，每一個人，都非常謝謝你。

這是一條不算短的旅程。將近七百個日子裡，有辛苦、有挫折，但也有充實、有成長。也許多年後回首今日，會覺得這些成就不足掛齒。但我們都曾是廣大學海中的一滴水珠；這片海洋孕育了我們，而我們所能回報的，就是盡我們的棉薄造就海洋的廣大。在這一刻，我期望這本論文能回饋給這片海洋一股新的生命力。

僅以此論文獻給無條件一直支持我的家人。

潘嘉琪 謹謝

2008·夏

中文摘要

仿水流優化演算法(Water Flow-like Algorithm, WFA)是新創的尋優演算法。WFA 係仿效水流在地理空間逐步流向最低點的特性，將代理人模擬成水流，在解空間中搜尋最佳解。WFA 模仿水流的分流和匯流特性，動態調整代理人的數量，有效地進行集中或分散搜尋。此外 WFA 亦模擬水流蒸發和降水的特性，使搜尋能有機會跳脫區域最佳解。WFA 最先提出是用於求解物件分群優化問題，本研究提出求解一般性物件排序優化問題的仿水流優化演算法(Water Flow-like Algorithm for Sequencing Problems, WFA4SP)。在 WFA 的演算程序規範下，規劃符合物件排序問題限制的仿水流搜尋機制。本研究提出仿連續空間位置指定法、子序列變動法、和鏈結關係繼承法三種求解排序問題的分流移步法，以設定每一演化代次水流移動的位置。接著提出循環比對法以計算物件排序解的相似度，供匯流作業使用。為驗證本研究所提的 WFA4SP 演算機制，本研究以旅行推銷員標竿問題為測試對象。比較三種分流移步法的成效，並與典型的遺傳演算法比較求解結果，探究本研究所提方法可行性。結果顯示在共同目標函式求算次數下，WFA4SP 表現較遺傳演算法優異。因匯流作業的循環比對作業會耗費大量計算資源，在共同求解時間下求解大維度問題時仍有改善空間。

關鍵字：啟發式演算法、仿水流優化演算法、物件排序優化問題、旅行推銷員問題

Abstract

Water Flow-like Algorithm, WFA, is a newly developed optimization algorithm that simulates a solution searching agent as a water flow traversing the lowest point of a terrain. The number of water flows is dynamically changed while water flows split into subflows against rough terrain and merge several flows into one single flow. Flow splitting and merging are mimicked by the WAF to conduct efficient optimum search in the solution space. In addition, water evaporation and precipitation are simulated in WFA to jump out of local optima or to broaden the searching area. WFA was originally proposed for grouping problems. This paper presents a WFA for solving object sequencing problems, namely WFA for Sequencing Problem, WFA4SP. Based on the original WFA computation structure, WFA4SP conducts the water flows to stand for feasible solutions of a sequencing problem, where the coordinates of the position of a water flow must be mutual exclusive. This paper presents three water flow methods: continuous space-like position assignment (CPA), subtour shuffle (SS), and link relationship inheritance (LRI), to assign feasible flow positions in water splitting operation of the WFA4SP. In addition, a cyclic similarity computation procedure is developed to evaluate the closeness of two flow positions to facilitate the water flow merging operation of the WFA4SP. Several TSP benchmarks were used to test the WFA4SP and results were compared with those from the best GA methods that uses GSX and TSP dedicated GX crossover operators. Numerical tests showed that the LRI position assignment method generated the best results than those from the CPA and SS methods. Result comparison showed that based on the same limit of the number of objective function calls, the WFA4SP outperforms the GA methods with GSX and GX crossover operators. Nevertheless, the high computation complexity of the flow merging

operation results in costly computation overhead than other operations of WFA4SP.

Keywords: heuristic algorithm, water flow-like algorithm, sequencing problems,
traveling salesman problems



目錄

誌謝	i
中文摘要	ii
Abstract.....	iii
目錄	v
圖目錄	vii
表目錄	viii
中英文名詞對照表	ix
符號列表	x
第 1 章 緒論	1
1.1 研究背景	1
1.2 研究目的	2
1.3 研究方法	2
1.4 章節概要	4
第 2 章 啟發式演算法及排序優化問題文獻探討	5
2.1 啟發式演算法	5
2.1.1 禁忌搜尋法	5
2.1.2 粒子團優化演算法	5
2.1.3 仿電磁吸斥優化演算法	6
2.1.4 遺傳演算法	7
2.1.5 蟻拓尋優法	9
2.1.6 仿水流優化演算法	9
2.2 物件排序優化問題	10
2.2.1 TSP 類型	10
2.2.2 GA 求解方式	11
2.3 文獻小結	14
第 3 章 排序優化問題的仿水流優化演算法	16
3.1 WFA 設計概念及成效	16

3.2 WFA4SP 的演算流程	18
3.3 初始設定	23
3.4 分流移步作業	23
3.4.1 決定分流數	23
3.4.2 分流移步	25
3.5 匯流作業	43
3.6 蒸發作業	46
3.7 降水作業	47
3.8 WFA4SP 整體演算程序	50
3.9 小結	51
第 4 章 仿水流優化演算法求解系統及範例驗證	53
4.1 仿水流離散優化演算法求解系統	53
4.2 系統驗證分析	55
4.2.1 WFA4SP 三種分流移步法比較	55
4.2.2 WFA4SP 演算成效分析	58
4.3 小結	65
第 5 章 結論與未來研究建議	66
5.1 結論	66
5.2 未來研究建議	67
參考文獻	68

圖目錄

圖 1-1.	研究方法流程圖	4
圖 2-1.	遺傳演算法資料結構	8
圖 2-2.	GSX 示意	13
圖 3-1.	仿水流優化演算法演化流程圖	20
圖 3-2.	分流數和目標函數值改善率的趨勢	24
圖 3-3.	CPA 法定義第一股子水流位置示意	28
圖 3-4.	CPA 法定義第二股以上之子水流位置示意	32
圖 3-5.	第二股以上子水流位置可能發生的極端狀況	33
圖 3-6.	水流 X_i 分出一股子水流 U_{il}	36
圖 3-7.	水流 i 位置向量 X_i 與母水流位置向量 \tilde{X}_i 內物件鏈結關係分佈示意	37
圖 3-8.	LRI 法產生水流 i 的子水流位置向量 U_{il} 範例	41
圖 3-9.	降水水流 i 位置向量偏移演算範例	49
圖 4-1.	WFA4SPDOS 介面	54
圖 4-2.	WFA4SPDOS 求解過程數值變化折線圖	54
圖 4-3.	批次求解介面	55
圖 4-4.	WFA4SP 求解 att48 求解結果比較	57
圖 4-5.	WFA4SP 三種分流模式求解 att48 收斂狀況	57
圖 4-6.	WFA4SP 和 GA 在相同目標函式呼叫次數下的求解結果比較	60
圖 4-7.	WFA4SP 和 GA 求解 att48 的收斂狀況	60
圖 4-8.	WFA4SP 及 GA 在相同求解時間下的求解結果	62
圖 4-9.	WFA4SP 和 GA 求解 att48 以求解時間為基的收斂狀況	62

表目錄

表 4-1.	WFA4SP 三種分流模式在相同目標函式呼叫次數下的求解結果	56
表 4-2.	WFA4SP 及 GA 在相同目標函式呼叫次數下的求解結果	59
表 4-3.	WFA4SP 及 GA 在相同求解時間下的求解結果	61
表 4-4.	WFA4SP 各作業流程所耗時間	63
表 4-5.	匯流作業耗時分析	63
表 4-6.	改進後的相似度計算法和原計算法的求解結果	64
表 4-7.	改進後的相似比對法和原計算法所耗時間比較	65



中英文名詞對照表

中文	英文	英文簡稱
仿水流優化演算法	Water Flow-like Algorithm	WFA
啟發式演算法	Heuristic Algorithm	
禁忌演算法	Tabu Search	TS
粒子群優化演算法	Particle Swarm Optimization	PSO
仿電磁吸斥優化演算法	Electromagnetic-like Algorithm	EM
遺傳演算法	Genetic Algorithm	GA
蟻拓尋優法	Ant Colony Optimization	ACO
螞蟻系統	Ant System	AS
搜尋代理人	Agent	
全域最佳解	global optima	
區域最佳解	local optima	
物件分群優化問題	Grouping Problem	
物件排序優化問題	Sequencing Problem	
旅行推銷員問題	Traveling Salesman Problem	TSP
標竿問題	Benchmark	
交配	crossover	

符號列表

問題定義

q ：變數個數，即問題的維度(dimension)。

$b_h^{(u)}$ ：變數 h 的上界值，在排序問題中此值為常數 q ， $h=1,2,\dots,q$ 。

$b_h^{(l)}$ ：變數 h 的下界值，在排序問題中此值為常數 1 ， $h=1,2,\dots,q$ 。

WFA4SP 參數設定

M_0 ：水流總質量。

\bar{n} ：分流的子水流數上限。

η ：總水流數上限。

γ ：每代次水氣蒸發比例， $0 < \gamma \leq 1$ 。

β ：水流合併作業依據的兩位置向量相似度；兩水流的位置向量相似度達此值時可合併， $0 < \beta \leq 1$ 。

ρ ：水氣飽和值，是水氣質量佔總質量的比值，達此值時須執行降水作業， $0 < \rho \leq 1$ 。

\bar{t} ：演化代次上限。

\bar{F} ：目標函式呼叫次數上限。

\bar{T} ：求解時間上限。

WFA4SP 變數資料

t ：當下水流演化的代次。

$^t\alpha$ ：在代次 t 時，水流的不分流門檻值，是一個以目標函數值改善率衡量的門檻， $0 < ^t\alpha \leq 1$ 。

N_t ：代次 t 的水流股數。

σ ：當代次大氣中的水氣含量， $0 \leq \sigma \leq M_0$ 。

水流演化

Ω ：當代次所有水流的編號集合， $\Omega = \{1, 2, \dots, N_t\}$ 。

\mathbf{X} ：當代次所有水流所代表的解集合(位置向量集合)， $\mathbf{X} = \{X_1, X_2, \dots, X_{N_t}\}$ 。

X_i ：水流 i 的位置向量， $X_i = [x_{i1} x_{i2} \dots x_{iq}]$ 。

X_{opt} ：目前搜尋到的最佳解，即最佳的水流位置向量。

X'_i ：參考水流 i 即時強制降水的水流位置向量， $X'_i = [x'_{i1} x'_{i2} \dots x'_{iq}]$ 。

\tilde{X}_i ：分流出水流 i 的母水流位置向量， $\tilde{X}_i = [\tilde{x}_{i1} \tilde{x}_{i2} \dots \tilde{x}_{iq}]$ 。

V_i ：水流 i 的速度向量， $V_i = [v_{i1} v_{i2} \dots v_{iq}]$ 。

P_i ：水流 i 的指標向量， $P_i = [p_{i1} p_{i2} \dots p_{iq}]$ 。

M_i ：水流 i 的質量。

$f(X_i)$ ：水流 i 位置向量代表的解的目標函數值。

n_i ：水流 i 分流出的子水流數， $1 \leq n_i \leq \bar{n}$ 。

U_{ik} ：水流 i 分出的水流 k 的位置向量， $U_{ik} = [u_{ik1} u_{ik2} \dots u_{ikq}]$ 。

Y_{ik} ：水流 i 分出的水流 k 的速度向量， $Y_{ik} = [y_{ik1} y_{ik2} \dots y_{ikq}]$ 。

Z_{ik} ：水流 i 分出的水流 k 的指標向量， $Z_{ik} = [z_{ik1} z_{ik2} \dots z_{ikq}]$ 。

w_{ik} ：水流 i 分出的水流 k 的質量。

第1章 緒論

本章說明本論文研究的背景、目的、方法、及研究流程。首先概述啟發式演算法的發展歷程及本研究所承襲的仿水流優化演算法。接著再說明本研究的研究目標、方法並展示研究流程。

1.1 研究背景

為因應優化問題複雜化的需求及電腦科技的快速演進，近數十年來學者致力於啟發式演算技術(heuristic algorithm)的開發。大多數的啟發式演算法都是使用優化解的搜尋代理人(agent)，透過有系統的或智慧型的搜尋方式，亦步亦趨地朝向更好的解搜尋及移動。演化搜尋的優點是避免耗時且成效不彰的地毯式搜尋，能在可接受的時間內找到容許的較佳解。因此啟發式演算法被廣泛應用於求解各類型優化問題。現今常被使用的啟發式演算法包括：禁忌搜尋法(Tabu Search, TS)、模擬退火法(Simulated Annealing, SA)、遺傳演算法(Genetic Algorithms, GA)、粒子團優化演算法(Particle Swarm Optimization, PSO)、仿電磁吸斥優化演算法(Electromagnetism-like Mechanism, EM)、……等。

啟發式演算法依代理人技術的使用情形可分為兩種型態：單一代理人及多代理人。單一代理人的特色是普遍使用歷史資料輔以梯度搜尋技術，確保搜尋方向不會回到較差的解上。缺點是過大型複雜問題時，求解過程過於保守且緩慢。對於高維度的問題容易落入區域最佳解中而無法全面性地搜尋到全域最佳解。為改善此一缺點，多代理人的啟發式演算法因應而生。多代理人的搜尋方式可提高搜尋廣度，提高求得全域最佳解的機率。缺點是當解趨近於收斂時，多代理人的解相似度極高，會產生許多重覆的搜尋資源浪費。

仿水流離散優化演算法(Water Flow-like Algorithm, WFA)是一種動態調整代理人數量的優化演算法。此法是以單一代理人為起始，視演化過程分為一個以上的代理人進行廣域搜尋。當兩個以上的代理人搜尋到鄰近區域時即合併成單一代理人，以避免重覆搜尋造成效率不彰。

WFA 是模擬自然界水流現象而設計的啟發式演算法。水流受地心引力吸引而流動，且受地理空間規範而漸趨流向地形上的最低點。WFA 演算過程反覆進行分流移步、匯流、蒸發、和降水四個作業。每個解由一股水流代理人代表，進行數量變動的多代理人搜尋。水流具有質量和速度，形成的動能足夠時即可依地形變化分支出多股子水流進行搜尋。當多股水流匯聚至同一位置時，即結合成單股水流，並且匯聚新的速度與質量。水流在流動過程中會逐漸散失質量於大氣中，當大氣中的水氣到達飽和程度時即會落下，產生新水流或添加質量到既有的水流中。

1.2 研究目的

WFA 提出時主要是用於求解物件分群優化問題，也嘗試求解物件排序優化問題(王元鵬, 2006; Yang and Wang, 2007)。經實驗驗證，WFA 求解物件分群優化問題的成果較 GA、EM、及 PSO 等啟發式演算法較佳。但在求解物件排序問題時，因物件排序不能重覆的限制，以及 WFA 的搜尋機制較為保守，無法得到較好的結果。本研究目的在研擬專門求解一般物件排序問題的仿水流優化演算法。在考量物件排序問題的特性下設計水流移步方法，使仿水流的優化解搜尋能有效地朝向最佳解前進。

1.3 研究方法

本研究承襲 Yang 及 Wang (2007) 提出的仿水流離散優化演算法(WFA)演化概念，提出專為求解物件排序優化問題特性的 WFA 演算法。WFA 著重區域性搜尋，遇物件排序問題時容易落入區域最佳解，難以求得全域最佳解。此外，WFA 也須倚靠降水的機制使水流重新分配移動，才有機會跳脫陷落的區域最佳解。

本研究將開發求解物件排序優化問題的 WFA 方法，名為 WFA4SP (Water Flow-like Algorithm for Sequencing Problems)。數值測試將使用 TSPLIB 內的標竿問題。求解效能及解品質將與遺傳演算法(GA)進行比較。為驗證所提演算法之可行性，本研究也將實作一套專解物件排序問題的仿水流優化演算求解系統。範例測試結果除了以數據展示外，也將以圖形內容展示。

本研究流程概述如下：

- (1) 構思研究主題：探討啟發式演算法並由 WFA 的功能不足衍生研究動機，據以規劃研究方向、範圍、及目標。
- (2) 蒐集並研讀相關文獻：蒐集各類啟發式演算法及相關應用的文獻，評斷各優化機制之優缺點及思索改善方案。
- (3) 蒐集標竿問題：蒐集物件排序優化問題之標竿問題，以供數值驗證及成果比較使用。
- (4) 思索並建構演算機制：探究水流流動之特性，輔以過往文獻各演算法之優劣特性，改善 WFA 演算機制並即求解驗證。反覆思索、實作，以研擬更佳的演算模式。
- (5) 開發求解系統：以建構好的演算機制開發優化離型系統。研擬親切易與的使用者介面及實作標竿問題的運行讀取功能。此外並提供可反覆多次執行求解的功能，以支援學術研究驗證演算法實作的成效。
- (6) 範例測試：以開發的優化系統測試 TSPLIB 內各種標竿問題。分析結果並探討本研究所提的仿水流優化演算法成效。
- (7) 論文撰寫：將研究成果有系統地整理成論文，供後續研究者參考。

各研究步驟如圖 1-1 所示。

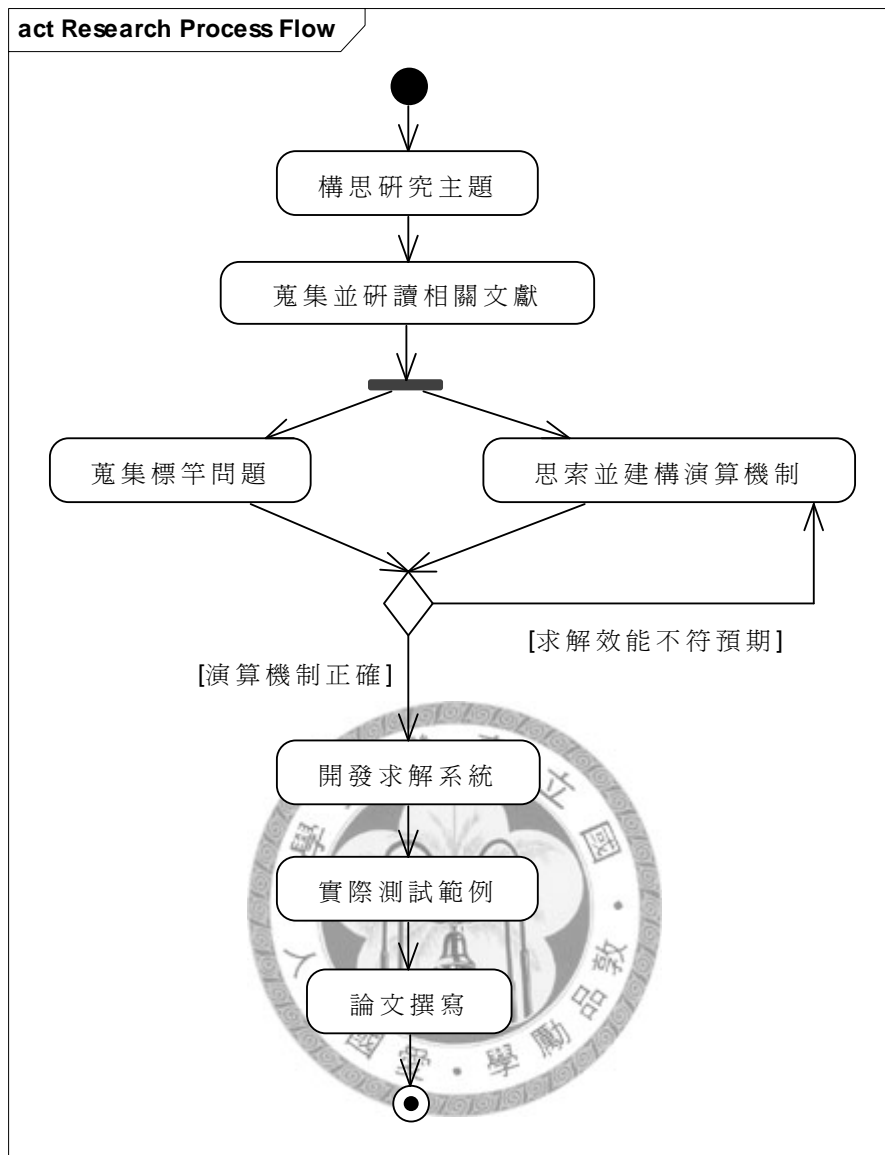


圖1-1. 研究方法流程圖

1.4 章節概要

本章揭示本研究之研究背景、研究目的，及研究方法及研究流程。第2章綜覽各啟發式演算法及這些演算法求解物件排序問題的特性和優缺點。第3章展示本研究研擬的求解一般性排序優化問題的仿水流優化演算法(WFA4SP, Water Flow-like Algorithm for Sequencing Problems)，內容詳細說明仿水流各演算作業的詳細流程。第4章驗證本研究提出的演算方法，展示開發的求解系統並求解 TSP 標竿問題。同時分析實驗結果評斷及演算效能。第5章總結本研究成果並提出對未來研究方向的建議。

第2章 啟發式演算法及排序優化問題文獻探討

本章將介紹現有的啟發式演算法，說明其演算特性及應用。接著介紹物件排序優化問題及其啟發式求解法。

2.1 啟發式演算法

2.1.1 禁忌搜尋法

禁忌搜尋法(Tabu Search, TS)的基本概念最早由 Fred Glover 於 1977 年提出，而後提出完整的演算架構(Glover and Laguna, 1997)。禁忌搜尋法是一種著重區域搜尋的啟發式演算法，在搜尋機制上配合記憶功能，使得搜尋不致落入區域最佳解(local optima)無法跳脫。

禁忌搜尋法是由一初始可行解為始，經由移步法則(move)進行微小變動，在鄰域(neighborhood)間有向搜尋最佳解。每個搜尋代次以上一次所搜尋到的最佳解經由移步產生鄰近解組成鄰域。接著在鄰域中搜尋出鄰域中的最佳解，並將形成此解的移步存入禁忌清單(Tabu list)中。為了避免重覆搜尋到相同的解，列為禁忌的移步將被禁止進行。除了禁忌清單外，尚有一個重要的準則就是破禁準則(aspiration criteria)，是用以破除禁忌清單中的限制。當所有鄰近解都列於禁忌清單無法移動時，依破禁準則清除所有禁忌清單，使搜尋可以繼續。另外，搜尋的停止條件，則依問題的差異而有不同的準則設定。

禁忌搜尋法僅使用單一代理人進行移動搜尋，面對較龐大的解空間時難以掌握全域的搜尋範圍。因此禁忌演算法不適合求解維度較高或解空間較龐大的優化問題。

2.1.2 粒子團優化演算法

粒子團優化演算法(Particle Swarm Optimization, PSO)是 Kennedy 和 Eberhart (1995)所提出，其演算概念是師法自自然界的生物群體行為。群居的生物如鳥群或魚群在移動時能透過群體間特定的訊息傳達方式，使得團體的行動方向一致。粒子團優化演算法即是仿照此種方式，將優化問題的一個解視為一個粒子，各粒

子會參考自己解搜尋歷程中最佳的經驗以及群體的最佳經驗決定移動方向。如此在群體彼此的影響和資訊參照下逐步接近最佳解。

以鳥群的覓食行為而言，鳥即是 PSO 中定義的粒子。一開始所有的鳥都不知道食物的位置，只是隨機地選擇方向和移動速度做為下次移動的依據。單隻鳥雖然不知道食物所在的正確位置，但是卻可以透過像嗅覺的方式來判斷附近是否有食物以及目前自己距離食物的遠近。在搜尋過程中，個別的鳥會記憶過往經驗中靠食物最近的位置，在食物搜尋行為中，會儘量朝該位置搜尋。除此之外，單隻鳥和鳥群間有資訊溝通的能力，知道整個鳥群的搜尋歷程中離食物最近的位置。因此，單隻鳥的搜尋趨向，除了自身的經驗記憶外，也會參考群體經驗，增加找到食物的可能性。粒子團優化演算法將一群本質是解代理人的粒子視為群居的生物，他們的覓食行為有兩個行為模式：

- (1) 認知模式(Cognition-Only Model)：生物個體有著自我最佳經驗的記憶；PSO 個體粒子會記憶它迄今曾經搜尋過的最佳解資訊。
- (2) 社會模式(Social-Only Model)：群居生物具有社會性，生物個體間的經驗會彼此交換並傳承下去。PSO 中的群體有記錄整個群體迄今最佳解的資訊。

個體粒子的位置即代表一個解，目標函式值用來衡量解的好壞。問題的全域最佳解即如鳥群尋覓的食物所在。藉由個體對過往搜尋經驗的記憶和分享，群體將群策群力朝最佳解移動。

2.1.3 仿電磁吸斥優化演算法

仿電磁吸斥優化演算法(Electromagnetism-like Mechanism, EM)是由 Birbil 和 Fang (2003)提出的一種新啟發式演算法。該演算法屬於多個智慧型代理人(multiple intelligent agents)的協同合作模式，亦是群體智慧(swarm intelligence)的一種。仿電磁吸斥優化演算法是模仿自然界電磁吸斥的物理現象，使一群代理人(agents)彼此互動且有向性地朝全域最佳解移動。

仿電磁吸斥優化演算法將各個解視為帶電荷(charge)的粒子(particle)。透過目標函數值的計算，比較各個解的目標函數值，優者賦予較高的電荷，劣者賦予較

低的電荷。粒子以其在群體中所帶電荷及在解空間中的相對位置建立電磁磁力場。此磁力場的建立係模仿或模擬目標函數值在解集空間的分佈狀況。接著仿照電磁磁力場中帶電粒子相互吸、斥的物理現象，訂定每個粒子下一代次的基本移動方向。由兩粒子的電荷和距離計算吸力和斥力。求算單一粒子受其他粒子吸的合力方向，設定該粒子的移動方向。粒子的移動方向是高電荷者遠離低電荷者，低電荷者追向高電荷者。合力方向係在規範移動的方向是正向或逆向。粒子真正的移動距離是由一個隨機值和基本移動距離決定。粒子群中的粒子全部移動到新的位置後，由所有解或只限最佳解的粒子進行（或不進行）區域搜尋，目的在於探究鄰近區域的解品質。區域搜尋若得到較佳的解，則逕行位置取代。透過粒子群中的粒子不斷地模擬電磁吸斥的物理現象，使劣解朝向優解移動、優解也能趨避較差解。每一個代次透過目標函數值的計算重新建立電磁磁力場，展示出朝向全域最佳解移動的群體智慧。

PSO 和 EM 都使用多代理人進行搜尋，且所有代理人的移動是受目前最佳解驅策而具有方向性。然而當所有代理人皆朝同一方向前進，過程中也未因搜尋到更好的解而改變整體方向時，代理人往往會收斂在一個區域內。在無類似 GA 的突變機制跳脫區域另闢新境下，當群體的初始位置不夠廣布於整個解空間時，搜尋結果常會落入區域最佳解。

2.1.4 遺傳演算法

遺傳演算法(Genetic Algorithm, GA)是由 Holland 於 1975 年提出，其觀念源自於達爾文演化論中「物競天擇，適者生存」的原理(Goldberg, 1989; Holland, 1992)。生物繁衍後代時，親代的基因會藉由遺傳複製給子代。在自然界的汰弱存強機制下，存留下的個體有高度的適應力，也會有較好的競爭優勢。演化過程基因會有突變的可能，會導致個體衰亡或是擁有適應環境的新能力。遺傳演算法簡化了生物演化的複雜過程，成為具體可行的最佳化問題求解步驟，是目前學術及業界廣泛運用的一種求解優化問題法。

遺傳演算法的演化群是由一個以上的獨立機體(organism)集合而成；每個機體由一條或多條染色體(chromosome)構成。染色體則由一序列的基因(gene)所組成。

基因值通常代表問題中的變數值。群體組成如圖 2-1 所示。

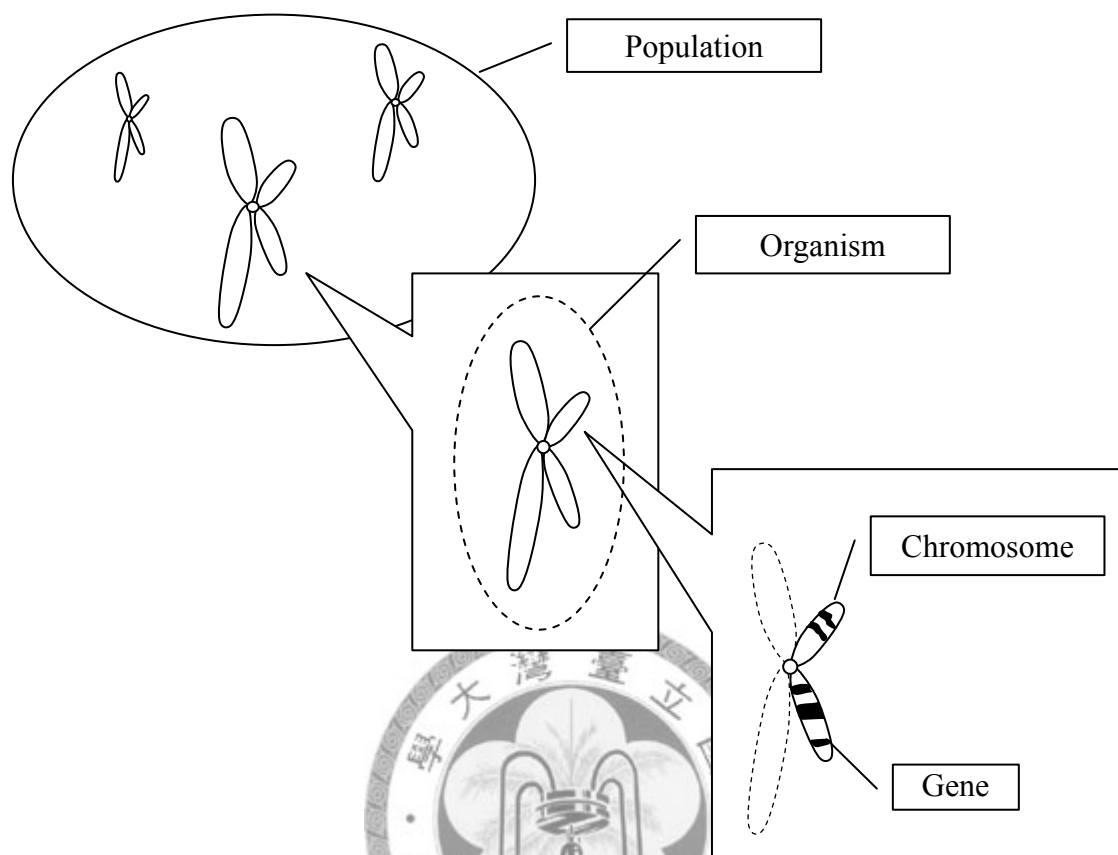


圖2-1. 遺傳演算法資料結構

遺傳演算法的演化機制是由機體內的染色體經由複製(reproduction)、交配(crossover)、突變(mutation)等運算過程產生子代(offspring)。由每個機體內的染色體值計算出該機體的適存值(fitness)。適存值決定該機體存活到下一代的機率。進行 GA 演化前，需將優化問題的變數編碼(encoding)成基因序列組成染色體。計算適存值時需將染色體解碼(decoding)成問題的解以估算解品質。編碼方式依問題型式不同，有實數編碼、整數編碼，及二元編碼等。不同編碼方式須配合不同的交配與突變演算方式，以適用於求解各類型優化問題。

GA 的演化機制可應用於求解各類型的優化問題，因此被學、業界廣泛研究改良並應用，至今已逐漸成熟。GA 的搜尋過程主要是倚靠保存優質染色體繼續下一代次的演化，交配過程並不能保證會產生較好的解。此外，演化趨於穩定時，群體內的染色體（代理人）內的基因值會趨向一致性，而不易跳脫區域最佳解。

2.1.5 蟻拓尋優法

蟻拓最佳化技術(Ant Colony Optimization, ACO)是晚近出現的一種啟發式演算法。ACO 是具有類似蟻拓優化演算的演算法統稱。第一個發展出來的 ACO 是螞蟻系統(Ant System, AS)。AS 是由 Dorigo 等人(1996)提出，演算法設計的靈感來自於自然界中螞蟻的覓食行為的觀察。

蟻拓優化技術師法自然界中螞蟻尋覓食物時的行為。首先，螞蟻會以隨機搜尋的方式離開巢穴尋找食物。移動時螞蟻會沿路釋放名為「費洛蒙(pheromone)」的分泌物，以引導螞蟻返回巢穴。若是出來覓食的螞蟻有找到食物，這些找到食物的螞蟻就會在返回巢穴的路徑上，添加額外的費洛蒙作為記號，供其他螞蟻依循。由於剛開始找到食物的螞蟻可能不只一隻，所以能到達食物的路線也不只一條。因往返頻率的密集或稀疏，在較短的路徑上會累積較多的費洛蒙。隨後的螞蟻處於隨機移動的狀態下，能辨別先前螞蟻所釋放的費洛蒙，跟隨費洛蒙強度最強的路徑行走，並繼續釋放自身的費洛蒙，更強化此路徑的標記。此程序有正向循環的回饋特色，當選擇相同路徑的螞蟻數愈多，後來的螞蟻選擇此路徑行走的機率也相對地提高。

典型的蟻拓優化應用是求解旅行推銷員問題(Traveling Salesman Problem, TSP)。螞蟻由其中一個城市出發，藉由費洛蒙的濃度分佈而尋找到鄰近的城市，再由搜尋到的城市出發，搜尋下一個鄰近的城市。如此逐步建構出最短的路徑繞行所有城市。經研究證實，ACO 應用於求解 TSP 有不錯的驗證成效。但螞蟻在搜尋過程中可能會發生過早收斂或停滯現象。過早收斂是由於螞蟻每次僅選擇費洛蒙濃度高且成本低的路徑，可能因此限於區域最佳解中而無法跳脫。停滯現象是由於螞蟻搜尋下一個鄰近城市的過程中，有兩條以上距離成本差異性不大的路徑，因此費洛蒙濃度沒有明顯差異，使螞蟻需耗費較長時間才能決定行走方向，同時也可能走到較長的路徑上。

2.1.6 仿水流優化演算法

仿水流優化演算法(Water Flow-like Algorithm, WFA)是由 Yang 及 Wang 提出，是一仿效水流在物理空間中流動的特性而設計的優化演算法(王元鵬, 2006; Yang

and Wang, 2007)。WFA 的演算機制是將解空間視為地理空間，每股水流的位置代表一個解。當水流在空間中流動時，會依地理特性改變流向、速度，逐步向地理最低處移動。水流具有分流的特性，會分成多股支流。當流經同一區域時，水流也可合併。水流流動過程，質量會逐漸散逸到大氣中，此時水氣可跳脫地形的限制，隨降水重新落到新的地域再進行流動。WFA 是藉由模仿水流的物理特性，動態調整搜尋代理人的數量（分流及匯流結果），朝解空間最佳解演化。

WFA 演化求解的流程有四個作業程序：分流移步、匯流、蒸發，及降水。Yang 及 Wang 的 WFA 以動能大小作為水流分支數的門檻，當動能越高時可分出較多的水流進行搜尋。決定分流數後搜尋水流的鄰近解以設定子水流的位置。WFA 以最好的前幾個移步位置指定成子水流的位置。當分流移步完成後，檢查是否有水流聚在同一區域，若有則將水流合併。匯流後執行蒸發作業將部份水流質量逸散到大氣中。當水氣質量達到飽和時即進行降水作業。

WFA 的水流移步演化機制有引用 TS 的禁忌作法，防止水流移步回流形成重覆搜尋。Yang 和 Wang (2007)發表的 WFA 只以裝箱問題(Bin Packing Problem, BPP)為求解對象。王元鵬(2006)曾以 WFA 嘗試求解旅行推銷員問題。結果顯示 WFA 求解物件分群問題表現較佳。求解物件排序問題時，因問題基本限制不易產生合理解，導致搜尋方式過於保守而形成區域搜尋。

2.2 物件排序優化問題

在物件排序優化問題中，當物件的排列不同時，即是一個不同的解且有不同的目標值。此種類型問題以旅行推銷員問題(TSP)最具一般性及代表性。TSP 最早是由 Hassler Whitney 於 1934 年所提出。推銷員需拜訪若干個城市，且每個城市僅能經過一次，最後回到原出發點(Flood, 1956)。TSP 的求解目標是能求得最短路徑。TSP 已被證明屬於 NP-complete 的問題(Karp, 1972)。問題定義雖簡單，但解空間大小隨城市數呈冪次增長，無法在多項式時間的複雜度下搜尋所有可行解以獲得全域最佳解(global optima)。以 20 個城市的 TSP 為例，其解空間將超過 10^{18} 。

2.2.1 TSP 類型

TSP 的研究發展迄今，基於原有的問題定義下延伸出數種不同類型的題型：

(1) 對稱型旅行推銷員問題(Symmetric traveling salesman problem, TSP)

是最為一般所探討的 TSP。兩城市間往返的距離成本皆相同，因此路徑迴圈反轉後的總距離成本亦和原路徑相同。

(2) 漢彌爾頓迴圈問題(Hamiltonian cycle problem, HCP)

漢彌爾頓迴圈問題屬於數學中的圖論問題。在已知的路徑圖(Graph)上找出一條迴圈經過每個頂點(Vertex)一次並回到起點。此問題為 NP-complete 型的問題。HCP 是 TSP 的特殊案例。

(3) 非對稱型旅行推銷員問題(Asymmetric traveling salesman problem, ATSP)

非對稱型 TSP 由 TSP 延伸而來，不同處在於兩城市間考量實際距離之外的成本。往返的距離成本不同，使距離成本矩陣呈不對稱關係。

(4) 優先順序旅行推銷員問題(Sequential ordering problem, SOP)

此問題是 ATSP 的延伸問題，由 ATSP 中加入優先順序的限制。亦即當兩城市間有優先順序限制時，則必定須先拜訪特定一個城市才能拜訪下一個。

(5) 具載貨量限制的車輛途程問題(Capacitated vehicle routing problem, CVRP)

在車輛有裝載限制下及貨運點需求量已知情況下，數輛車從運輸網路上同一個物流中心(depot)出發，通過其他每個貨運點恰好一次，再回到原出發點，使得總運送成本最小。

2.2.2 GA 求解方式

使用 GA 求解典型物件排序問題的 TSP 時，常採用整數排序編碼法(Permutation)。並使用針對排序問題而設計의 交配法來進行演化，使得新產生的解不會違反排序問題物件不得重覆出現的基本限制。常用的交配法有部份相映交配法(Partial-mapped Crossover, PMX)、順序對映交配法(Order Crossover, OX)、循環對映交配法(Cycle Crossover)、位置為基交配法(Position-based Crossover)、順序為基

交配法(Order-based Crossover)……等。近年來亦有學者針對 TSP 特性開發出新的啟發式交配法，例如 GSX 和 GX：

(1) Greedy Subtour Crossover (GSX)

GSX 交配法是 Sengoku 和 Yoshihara (1998) 所提出，此種交配法是截取兩條親代染色體中一部份的子序列合併成子代。舉例來說，兩條親代染色體分別是 $p_1 = [48213675]$ 和 $p_2 = [23478651]$ ，如圖 2-2 所示。演算初始先由 p_1 中隨機選定一個物件，假設是位置 $a_1 = 5$ 的物件 3。將物件 3 指定為子代 c_1 的染色體中同一位置(位置 a_1)的物件。找出物件 3 在親代 p_2 的位置 $a_2 = 2$ 。找出親代 p_1 位置 $a_1 - 1$ 的物件 1，如果物件 1 尚未排入子代染色體的話，則排入子代染色體的物件 3 之前；也就是位置 $a_1 - 1$ 處。若物件 1 已排入子代則停止往前搜尋。由親代 p_2 的位置 $a_2 + 1$ 找到物件 4。若物件 4 尚未排入子代染色體，則排入子代染色體的物件 3 之後，也就是位置 $a_2 + 1$ 處。若物件 4 已排入則往後搜尋。如此由親代 p_1 先往前再由親代 p_2 往後依序指定二親代的部份序列到子代，直到無法排入為止。往前後指定都停止後，將剩下未排入的物件隨機排入未指定的基因值。第二條子代染色體仍以最初選定的物件，依親代 p_1 往後、 p_2 往前方式指定染色體的基因值。圖 2-2(b)展示子代 c_2 的基因指定結果。

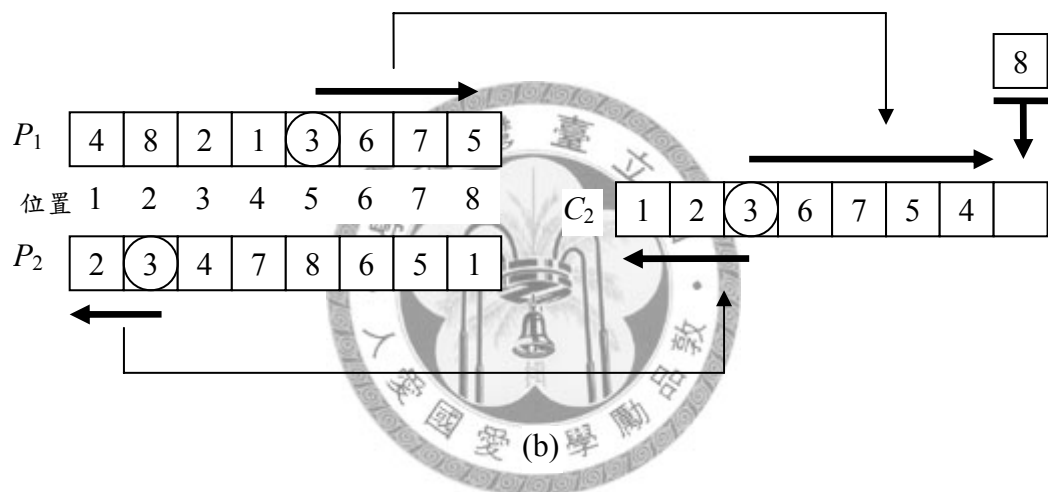
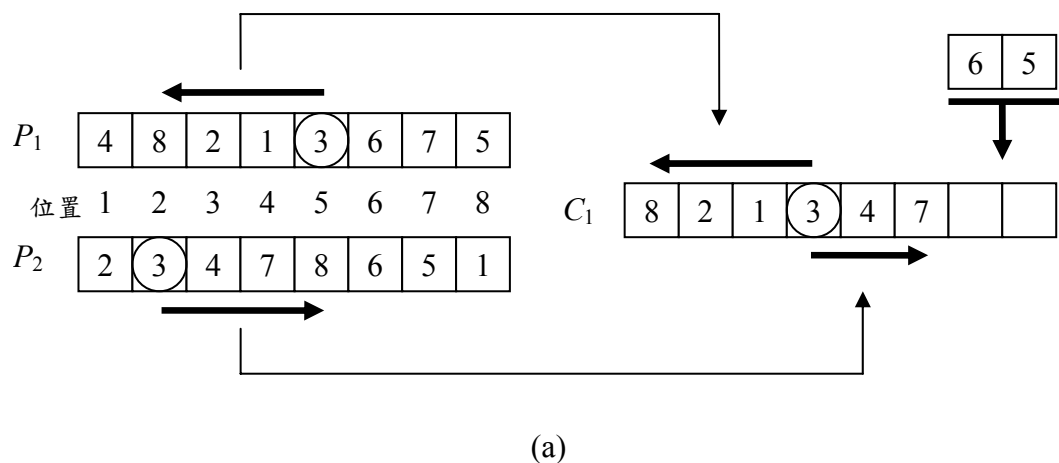


圖2-2. GSX 示意

(2) Greedy Crossover (GX)

GX 交配法是 Yang (1997) 專為求解 TSP 而提出的啟發式交配法。GX 法在交配時會比對兩條親代染色體中是否有相同的城市鏈結，並且會參照 TSP 中的距離成本矩陣，找出距離成本較短的城市鏈結建構出子代。例如兩條親代染色體分別為 $p_1 = [14253876]$ 和 $p_2 = [17826435]$ 。決定第一條子代染色體的演算初始先隨機選定一個物件，假設此例為物件 4。先檢查選定的物件 4 在兩條親代染色體中前後接鄰的物件，亦即 p_1 的 $\{1,2\}$ 和 p_2 的 $\{6,3\}$ ，是否有相同物件。若沒有相同物件則從兩染色體物件 4 起始向右的兩鏈結（此例是 $\langle 42 \rangle$ 和 $\langle 43 \rangle$ ）中選擇距離較短的鏈結

設定子代染色體的基因值。假設此例 $\langle 43 \rangle$ 是距離較短者，則第一條子代染色體第一和第二基因值設成 4 和 3。接著比對物件 3 在兩條染色體中前後接鄰的物件，亦即比對 $\{5,8\}$ 和 $\{4,5\}$ 。因兩條親代染色體中物件 5 都和物件 3 接鄰，因此將物件 5 排入物件 3 之後。再比對物件 5 在兩條染色體中前後接鄰的物件。因物件 3 已排入子代中所以必須從接鄰的物件集合中剔除，亦即比對 $\{2\}$ 和 $\{1\}$ 。兩集合中沒有相同的物件，因此由物件 5 起始向右的兩鏈結 $\langle 53 \rangle$ 和 $\langle 51 \rangle$ 中先剔除已排入子代的物件所形成的鏈結，亦即剔除 $\langle 53 \rangle$ ，再將剩下的鏈結所對應的未排入物件，此為物件 1，排入子代中。若兩鏈結皆被剔除，則由剩下尚未排入子代的物件中任選一個物件排入子代染色體序列。

第二條子代染色體和第一條染色體產生方式大致相同。唯一不同處在於當目前檢查的物件在兩條親代染色體中沒有相同的接鄰物件時，是從兩親代染色體中目前檢查物件起始向左的兩鏈結裡選擇距離較短的鏈結，再將鏈結所對應的未排入物件排入子代序列中。交配完成後的染色體依序列循環特性，將序列第一順位調整為物件 1。

2.3 文獻小結

本章介紹了現有的啟發式演算法及物序排序優化問題。啟發式演算法在求解行為上可依區分為單一代理人及多代理人；搜尋行為可分為區域搜尋及全域搜尋。單一代理人的搜尋成本雖然較低，但對於較複雜或維度較高的問題時，因搜尋過於保守而無法有效求解。多代理人的演算方式雖可廣泛搜尋解空間，但代理人數量過多，或是搜尋到的解已趨收斂時會造成搜尋成本的浪費。

若啟發式演算法的代理人移動演化是具有方向性時，如：PSO、EM 等，容易朝同方向搜尋而落入區域最佳解不易跳脫。相對地，當演化作業不具明確方向性時，如：GA，雖可廣泛搜尋整個解空間，但搜尋跳動性過大，無法精確向全域最佳解收斂。

基於上述缺點的考量，本研究承襲原有之仿水流演算概念，在動態調整搜尋代理人數量且能兼顧區域搜尋及全域搜尋下，開發改進式的仿水流優化演算法以

求解物件排序優化問題。下一章將詳細說明本研究開發的物件排序優化問題之仿水流優化演算法，WFA4SP，的演化細節，以及求解 TSP 的仿水流演化求解程序。



第3章 排序優化問題的仿水流優化演算法

WFA 係仿照水流流動時的物理特性，以初始單股繼而多股分流的概念設計搜尋機制以求解物件分群問題。本研究承襲 WFA 的概念開發求解一般性排序優化問題的仿水流優化演算法，簡稱 WFA4SP(Water Flow-like Algorithm for Sequencing Problems)。WFA4SP 模擬水流流動時的物理行為，動態調整代理人數量在排序問題的解空間中進行最佳排序解的搜尋。本章先說明 WFA 的演算內容，再展示 WFA4SP 求解一般排序優化問題的演算程序及相關的資料架構。

3.1 WFA 設計概念及成效

水在自然環境中有固、液、氣體三態存在，其中以液態佔大多數。當水以液態存在於地理空間時，受到地心引力的影響，會依循承載介面（地表）的延伸移動它的位置，且由高處往低處流動，但當衝量足夠時也能逆勢上移。當水以氣態存在時，則不受地理空間影響，可移動至不同領空再凝為液體落下地表。因此水流在地理空間的流動中具有趨低性及分散性，有極大的機率會流經地理上的最低點。此種趨低的特性是仿水流優化演算尋優的精神。

WFA 將水流的特性模擬為啟發式演算法中代理人搜尋最佳解的演化機制。將地理空間的最低點視為優化問題中的搜尋目標，依據水流特性可逐步移動水流至最佳解。水流在地理空間中的流動與代理人在優化問題中的搜尋行為兩者為的對應關係是：

1. 水受到地心引力影響，流動時會沿地理空間之高低起伏，從高處流入低窪地區。
當水流具備速度及衝量時，則可能從低處移動至較高的位置。代理人在搜尋時，將較差的解視為解空間中的高點，較佳者視為低點，搜尋時逐漸朝佳者（低點）推進。
2. 水流流動時會因地形的分歧，分出多支子水流往不同方向流動，質量分配則由動量守衡公式規範。代理人可視情況機動分出多個代理人同時搜尋，延伸搜尋的廣度。

3. 水流流動具有方向性，除非受到地形及外力影響，否則會依循原本的速度方向前進。代理人在解的搜尋上，若搜尋到較好的解，應維持原有方向持續搜尋。
4. 不同水流在同一時間流經同一地區時，會匯流成單一水流，其質量及速度的變動仍會服從能量守恆法則。匯合後的水流有較大的質量保持流動。兩個以上的代理人在同一代次搜尋到相同位置時，將代理人合併，以增加搜尋的動能，同時避免重覆搜尋浪費資源。
5. 水流在流動時其質量會持續蒸散到大氣中。當大氣中的水氣達到飽和，即會降水落回地表，併入地表水流或另起水流。降水可將水流帶至不易搜尋到的地理空間再度流動。當代理人在解搜尋過程中陷於區域最佳解時透過降水的程序，可增加水量脫離原本停滯的位置。

迄今廣為使用的多代理人啟發式演算法的代理人數量在演化過程都保持固定。單一代理人在搜尋上較為緩慢，而多代理人雖然可改善此一缺點，但在目標值趨於收斂時會產生許多重覆搜尋而降低演算效率。WEA 不同於其他多代理人啟發式演算法，最明顯的是代理人數量會隨搜尋狀況而變化。其採用的水流物理特性有：

1. 分流特性

水流動量夠大時面對分歧的地形會分流流動。當動量小時則維持一股水流繼續流動。

2. 匯流特性

當兩股以上的水流流經同一區域時，會合併為單一水流。質量合併增加，流速也匯聚增強。如此可增加搜尋的能量，也避免多股水流同時在同一區域重覆搜尋。

3. 蒸發及降水特性

水流在流動時會逐漸散失質量到大氣中。搜尋能力較差的水流會蒸發所有質量而消失，以減少無效的搜尋。大氣中的水氣達飽和狀態時則降水匯聚到流動中的水流，或另起水流流動。降水可使質量少的水流獲得挹注，使搜尋得以持續。

而當水流速度趨於停滯時，水流會持續蒸發直到消失。

WFA 提出時主在處理物件分群優化問題。每個代理人在每代搜尋中會選擇最適合變動的變數進行調整，即持續找出最適合變動的物件以改變其所屬群別。因此 WFA 搜尋機制看似保守，但卻可持續地往最佳方向收斂。經實驗驗證，WFA 較其他多點搜尋的啟發式演算法，如 GA、PSO、及 EM，更適於求解物件分群優化問題。王元鵬(2006)也使用 WFA 對分群問題的搜尋法嘗試求解 TSP 的物件排序優化問題。結果因 TSP 限制太大，WFA 於搜尋變動上較為保守，因此求解效能較 GA 和 PSO 都差。本研究承襲 WFA 的精神建構一般性的排序優化問題專向的 WFA 演算架構，名為 WFA4SP。WFA4SP 以仿水流的尋優演化程序為基礎，進一步規劃符合排序問題限制的演算策略以求解一般性的物件排序離散優化問題。

3.2 WFA4SP 的演算流程

TSP 是最典型的排序優化問題。推銷員必須拜訪一定數目的城市，每個城市僅能經過一次。而城市間的距離各有不同，求解目標在找出拜訪各個城市的最佳順序使繞行的距離最短。排序問題要求所有物件皆必須排入序列中，且物件不可重覆出現在不同順位上，且一個順位只能有一個物件。在 N 個物件的排序問題中，整個解空間有 $N!$ 個可行解。WFA4SP 的演算程序須確保解在合理解區域內。

WFA4SP 的演化程序如圖 3-1 所示。初始設定包含水流總質量、分流股數上限、水流合併相似度、水流質量蒸發率、總水流數上限、和水氣飽和度等參數。水流演化以一股水流開始，並以隨機方式決定其初始位置 X_1 ，和其母水流位置 \tilde{X}_1 後開始流動。

水流流動時依解空間的變化，可分出數股支流進行搜尋。分流支數依水流演化過程解品質的變化率來決定。當目前水流位置的解較母水流好，就保持原方向流動搜尋。若解較母水流差，則分出較多子水流進行多方搜尋。分流時會依目標函數值優劣分配子水流的質量。目標值越好的子水流可分到的質量較多，以供後續持續搜尋；反之目標值較差的水流分到較少質量，但可在後續演化過程和其他水流合併，或是蒸發至大氣中經由降水重啟新的搜尋。

水流移步或分流後，須檢查水流位置是否鄰近，當兩條以上的水流位置鄰接時保留目標值較好的一股水流，合併其餘水流質量進入下一代次水流演化。匯流演算後各水流進行蒸發作業。地面所有水流都將散失一部份的質量於大氣中，若水流蒸發後的質量低於所設定的下限，則該水流全數蒸散後消失。大氣聚積水氣至飽和時，即進行降水作業，將水氣依現有水流的質量比例作分配，以供下一代次的水流演化使用。

執行一代次水流演化作業後，檢查是否符合停止條件。若是則完成水流演化，以迄今最佳解作為求解最終結果。否則以現有水流再次由分流作業開始進行下一代次水流演化程序。



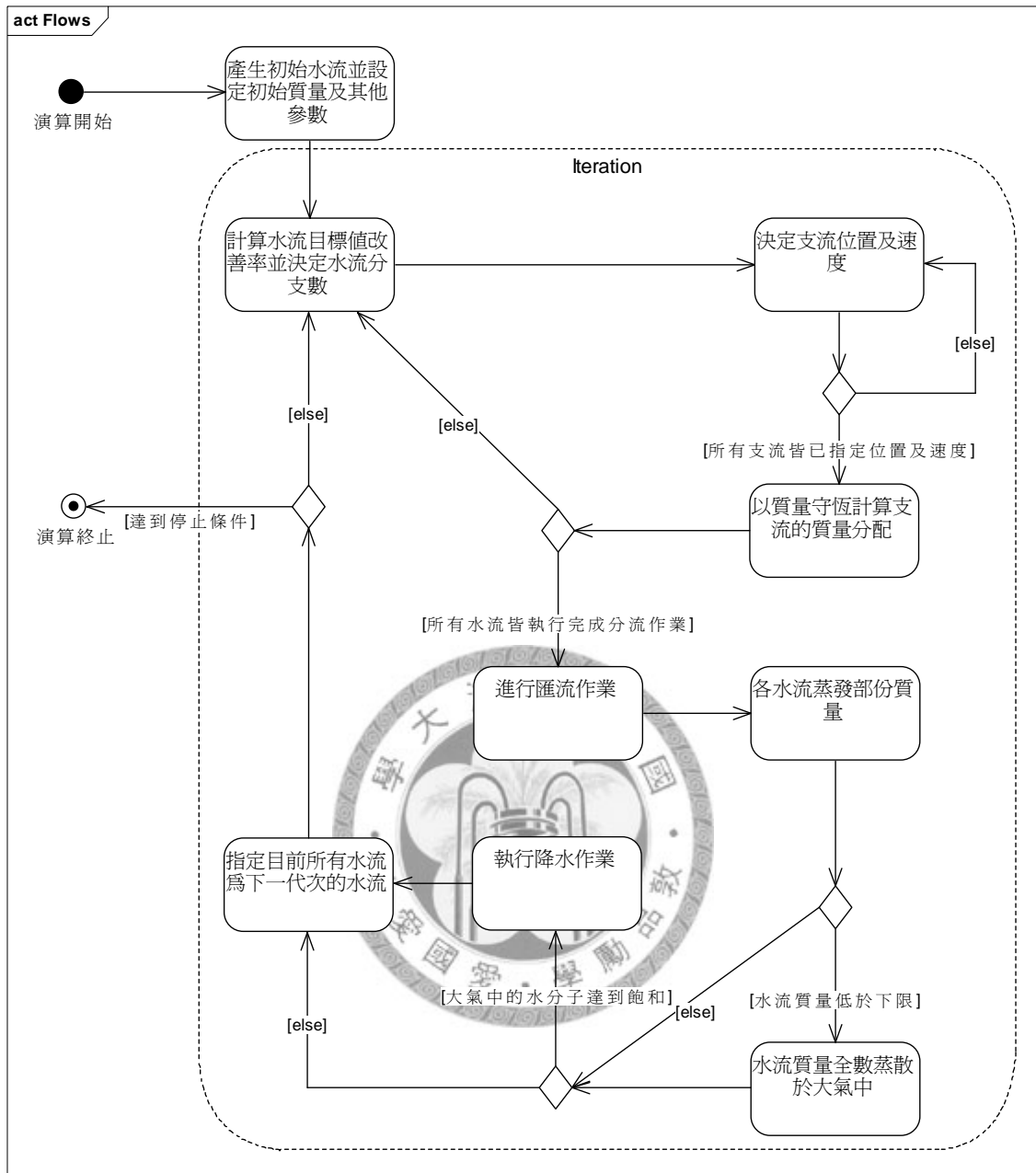


圖3-1. 仿水流優化演算法演化流程圖

基於上述求解的流程規劃，WFA4SP 演化流程的四項主要作業是：

1. 分流移步作業

水流可依原有方向維持單一水流或分流成多股子水流流動，因此分流移步作業前須先決定分流數。先依目前水流的目標函數值相對於其母水流的改善率決定分流數後，依目前水流速度設定各子水流的位置。若分流數是 1 則維持單一水流流動。計算各子水流位置的目標函數值後依比例分配各子水流質量，更新水流集

合。

2. 匯流作業

檢驗水流集合的各水流位置，將位置鄰近的水流進行合併。多股水流合併時質量守恆，保留目標值最好的一股水流，將其餘水流的質量併入該水流。

3. 蒸發作業

將所有水流蒸發部份質量到大氣中，使水能跳脫地理空間的限制，以接續的降水作業移動到不同區域。蒸發後若水流質量小於設定的下限，則將該水流質量全數散發於大氣中。

4. 降水作業

大氣中的水氣含量達到飽和時，即啟動降水作業。降水可使轉為氣態的水份在新的區域落下形成水流，進行不同區域的流動搜尋。

接下來說明本研究研擬的 WFA4SP 的演算細節，內容包含初始參數設定程序以及分流、匯流、蒸發、降水等四個作業流程細節。首先定義使用的數學符號：

q ：變數個數，即問題的維度(dimension)。

$b_h^{(u)}$ ：變數 h 的上界值，在排序問題中此值為常數 q ， $h=1,2,\dots,q$ 。

$b_h^{(l)}$ ：變數 h 的下界值，在排序問題中此值為常數 1 ， $h=1,2,\dots,q$ 。

t ：當下水流演化的代次。

${}^t\alpha$ ：在代次 t 時，水流的不分流門檻值，是一個以目標函數值改善率衡量的門檻， $0 < {}^t\alpha \leq 1$ 。

M_0 ：水流總質量。

N_t ：代次 t 的水流股數。

\bar{n} ：分流的子水流數上限。

η ：總水流數上限。

γ ：每代次水氣蒸發比例， $0 < \gamma \leq 1$ 。

β ：水流合併作業依據的兩位置向量相似度；兩水流的位置向量相似度達此值時可合併， $0 < \beta \leq 1$ 。

ρ ：水氣飽和值，是水氣質量佔總質量的比值，達此值時須執行降水作業， $0 < \rho \leq 1$ 。

σ ：當代次大氣中的水氣含量， $0 \leq \sigma \leq M_0$ 。

Ω ：當代次所有水流的編號集合， $\Omega = \{1, 2, \dots, N_t\}$ 。

\mathbf{X} ：當代次所有水流所代表的解集合(位置向量集合)， $\mathbf{X} = \{X_1, X_2, \dots, X_{N_t}\}$ 。

X_i ：水流 i 的位置向量， $X_i = [x_{i1} x_{i2} \dots x_{iq}]$ 。

X_{opt} ：目前搜尋到的最佳解，即最佳的水流位置向量。

X'_i ：參考水流 i 即時強制降水的水流位置向量， $X'_i = [x'_{i1} x'_{i2} \dots x'_{iq}]$ 。

\tilde{X}_i ：分流出水流 i 的母水流位置向量， $\tilde{X}_i = [\tilde{x}_{i1} \tilde{x}_{i2} \dots \tilde{x}_{iq}]$ 。

V_i ：水流 i 的速度向量， $V_i = [v_{i1} v_{i2} \dots v_{iq}]$ 。

M_i ：水流 i 的質量。

$f(X_i)$ ：水流 i 位置向量代表的解的目標函數值。

n_i ：水流 i 分流出的子水流數， $1 \leq n_i \leq \bar{n}$ 。

U_{ik} ：水流 i 分出的子水流 k 的位置向量， $U_{ik} = [u_{ik1} u_{ik2} \dots u_{ikq}]$ 。

Y_{ik} ：水流 i 分出的子水流 k 的速度向量， $Y_{ik} = [y_{ik1} y_{ik2} \dots y_{ikq}]$ 。

w_{ik} ：水流 i 分出的子水流 k 的質量。

以下各節將詳述 WFA4SP 演算中各作業的運算程序。

3.3 初始設定

WFA4SP 演化求解初始須設定初始單股水流質量 M_0 、水流分流數上限 \bar{n} 、水流合併相似度 β 、每代次水流質量蒸發率 γ 、總水流數上限 η 、和水氣飽和度 ρ 。 M_0 用以設定初始單股水流的質量，亦即水流總質量。 \bar{n} 限定子水流的分支數，數值不宜過大；建議應小於 5 防止浪費搜尋資源。 β 是兩水流位置相似程度的門檻，用以決定是否合併。 γ 用以求算水流的蒸發質量。 η 用以限定總水流數，不使水流數量無限擴張。 ρ 設定大氣中水氣含量的飽和臨界值，超過此值即視為達飽和狀態。變數 h 的上界值 $b_h^{(u)}$ 和下界值 $b_h^{(l)}$ 即在排序問題中是物件 h 的排列順位，因此分別是 1 和變數個數 q 。

排序問題解限定物件不可重複出現在不同順位上，初始解可以隨機方式設定，但須符合限制。初始水流的位置向量隨機設定 $X_1 = [x_{11} x_{12} \dots x_{1q}]$ ， $x_{1j} \in \{1, 2, \dots, q\}$ ，且 $x_{1i} \neq x_{1k}, \forall i \neq k$ 。同時設定初始水流的母水流位置向量 $\tilde{X}_1 = [\tilde{x}_{11} \tilde{x}_{12} \dots \tilde{x}_{1q}]$ ， $\tilde{x}_{1j} \in \{1, 2, \dots, q\}$ ，且 $\tilde{x}_{1i} \neq \tilde{x}_{1k}, \forall i \neq k$ 。

3.4 分流移步作業

分流移步作業分為兩部份：首先決定當下各股水流可分出的分流數，再決定各支流的位置及質量，並更新其他屬性。

3.4.1 決定分流數

進行分流作業時，須先決定當下各股水流欲分出的支流數。Yang 和 Wang (2007) 提出的分流數係由當下水流所隱含的動量決定。水流在地理空間的移動遵守能量守恆原則，由高處落至低處時會將位能轉換為動能表現於速度上。地形落差越大時，水流速度越快，能維持方向集中，有助於快速流向地理低處。當地形較平緩時，水流容易擴散至四周，搜尋鄰近區域。

WFA4SP 的分流數量規劃成水流落差由大到小呈指數縮減。當目標函數值改

善率大時，分流數少；改善率小則分出的水流多。WFA4SP 水流 i 的分流數

$$n_i = \begin{cases} 1, & \text{if } \dot{f}_i > {}^t\alpha \\ \text{RoundOff}\left(\bar{n}\left(1-\frac{\dot{f}_i}{{}^t\alpha}\right)\right), & \text{otherwise} \end{cases} \quad (3-1)$$

$$\dot{f}_i = \frac{|f(\tilde{X}_i) - f(X_i)|}{f(\tilde{X}_i)},$$

X_i 是水流 i 的位置向量， \tilde{X}_i 是水流 i 的母水流位置向量，而 ${}^t\alpha$ 是不分流的一個門檻值，其值隨著代次演進而更新。超過此門檻值時，維持該股水流逕行移位演化；否則分出二股以上子水流作廣域搜尋。圖 3-2 說明分流數 n_i 和目標函數值改善率 \dot{f}_i 的關係。此例假定分流的子水流上限 $\bar{n} = 5$ ，不分流門檻值 ${}^t\alpha = 0.4$ 。當 \dot{f}_i 超過 ${}^t\alpha$ 時，維持一股不分流。低於 ${}^t\alpha$ 時以指數趨勢縮減並取整數為分流數。

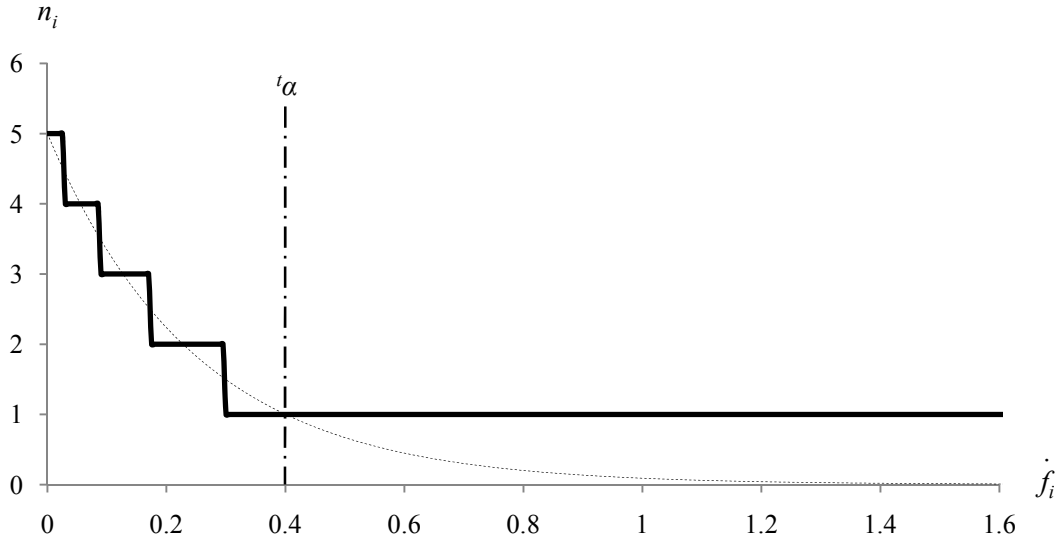


圖3-2. 分流數和目標函數值改善率的趨勢

為因應不同問題的目標函數值刻劃不同，及不同的目標函數值改善率，本研究設計三種門檻值設定法：

(1) 固定成初始目標函數值改善率

以初始水流相對其母水流的目標函數值改善率設為所有代次的 ${}^t\alpha$ ，亦即

$${}^t\alpha = {}^1\alpha = \dot{f}_1 = \frac{|f(\tilde{X}_1) - f(X_1)|}{f(\tilde{X}_1)}, \forall t. \quad (3-2)$$

(2) 鄰近代次目標函數值改善率平均

隨著代次演化，目標函數值的改善率會趨緩。若採固定門檻將形成高門檻值，無法不進行分流。因此取當下演化代次 t 的前 r 代次所有水流的平均目標函數值改善率的平均，為門檻值，亦即

$${}^t\alpha = \frac{1}{\min\{r, t-1\}} \sum_{t'=\max\{t-r, 1\}}^{t-1} \frac{1}{N_{t'}} \sum_{i=1}^{N_{t'}} \dot{f}_i. \quad (3-3)$$

(3) 累計目標函數值改善率平均

累算到演化代次 $t-1$ 為止，所有代次中所有水流目標函數值改善率的平均，設定為當下演化代次 t 的門檻值，亦即

$${}^t\alpha = \frac{1}{t-1} \sum_{t'=1}^{t-1} \frac{1}{N_{t'}} \sum_{i=1}^{N_{t'}} \dot{f}_i. \quad (3-4)$$

3.4.2 分流移步

決定分流數後接著決定分出的各水流移步後的位置，並且適當分配水流質量到子水流中。

決定分流位置

由於物件排序優化問題的解是以整數序列來表示，分流移步時無法沿襲地表水流在連續空間般，由位置加上速度決定支流移步的位置而不會違反排列限制。本研究提出的 WFA4SP 針對一般性的物件排序優化問題，開發出三種分流移步法，確保分流後的支流位置皆代表排序問題的一個合理解。以下分別詳述三種分流移步法：

a. 仿連續空間位置指定法(continuous-space-like position assignment method, CPA)

本法將離散的排序問題的解空間視為變動大且不合理解散置的連續整數解空

間，再依循位置向量加上速度向量的方式設定子水流的分流位置。仿速度向量決定子水流前進的方向。由於分流移步時可能不分流而維持單一股水流前進，因此本法採用當下水流的速度向量來決定第一股子水流的分流位置。其他股子水流的位置向量則由第一股子水流的位置向量變動設定之。分流位置向量的座標值代表物件的排列序列，座標值間須符合排序問題不得重覆的限制。分流後子水流的其他屬性的值也須有適當的分配規劃。以下分別說明第一股子水流及其他各子水流的位置和屬性設定法。

第一股子水流

因分流移步的位置向量座標值不能違反排序不得重覆原則，水流 i 分出的第一股子水流向量座標的位置由水流 i 的速度分量大小決定其設定順位。速度分量大者優先設定座標值（排序問題的物件編號）。當座標值已由其他分量佔用時，以未被佔用的最鄰近座標值設定之，且以與速度分量同向的鄰近座標值優先設定，若已被佔用再考量反向者。

令集合 D 是尚未排入序列的候選座標，初始設定 $D = \{1, 2, \dots, q\}$ ，最小值 $d_{\min} = 1$ ，最大值 $d_{\max} = q$ 。當 D 更新時，須更新上下限

$$\begin{aligned} d_{\min} &\leftarrow \min_{\forall k \in D} \{k\} \\ d_{\max} &\leftarrow \max_{\forall k \in D} \{k\} \end{aligned} \quad (3-5)$$

水流 i 的速度向量可由目前和母水流的位置向量求算：

$$V_i = X_i - \tilde{X}_i = [v_{i1} v_{i2} \dots v_{iq}] \quad (3-6)$$

第一股子水流位置向量的設定演算程序如下：

步驟 1：依式(3-6)計算水流 i 的速度向量 V_i 。

步驟 2：依速度向量 V_i 的分量絕對值由大到小排序得到一維度索引序列

$$O = \langle o_1 o_2 \dots o_q \rangle, |v_{io_j}| \geq |v_{io_{j+1}}|, \forall j = 1, 2, \dots, q-1。並建立尚未排入序列的候選$$

座標值集合 $D = \{1, 2, \dots, q\}$ ，且令 $d_{\min} = 1$ 及 $d_{\max} = q$ 。

步驟 3：決定水流 i 分出的第一股子水流的位置向量 U_{i1} 。

步驟 3.1：令 $j \leftarrow 1$ 。

步驟 3.2：令 $\alpha \leftarrow x_{io_j} + v_{io_j}$ ， $\delta \leftarrow \min_{\forall d \in D} \{\alpha - d\}$ 。

步驟 3.3：指定第一股子水流的位置分量

$$u_{i1o_j} \leftarrow \begin{cases} d_{\min}, & \text{if } \alpha \leq d_{\min} \\ d_{\max}, & \text{elseif } \alpha \geq d_{\max} \\ \alpha, & \text{elseif } \alpha \in D \\ \alpha + \delta, & \text{elseif } (v_{io_j} \geq 0 \text{ and } \alpha + \delta \in D) \text{ or } (v_{io_j} < 0 \text{ and } \alpha + \delta \notin D) \\ \alpha - \delta, & \text{otherwise} \end{cases}$$

，更新 $D \leftarrow D - \{u_{i1o_j}\}$ 、 $d_{\min} \leftarrow \min_{\forall k \in D} \{k\}$ 、 $d_{\max} \leftarrow \max_{\forall k \in D} \{k\}$ 、和 $j \leftarrow j + 1$ 。

步驟 3.4：若 $j > q$ 表示所有位置向量的分量皆指定完成，結束演算程序，否則返回步驟 3.2。

如圖 3-3 所示，先以式(3-6)計算水流 i 的速度向量 V_i ，排出 $O = \langle 913564782 \rangle$ ，並建立候選集合為 $D = \{1, 2, \dots, 9\}$ ，下限 $d_{\min} = 1$ ，上限 $d_{\max} = 9$ 。計算維度 $o_1 = 9$ 的位置分量 $x_{i9} = 8$ 加上速度分量 $v_{i9} = 3$ 等於 11。由於 11 大於上限 d_{\max} ，因此指定子水流的位置分量 u_{i19} 為上限值，也就是座標值 9。更新 D 為 $D = \{1, 2, 3, 4, 5, 6, 7, 8\}$ ，下限不變，上限 $d_{\max} = 8$ 。接著計算維度 $o_2 = 1$ 的位置分量 $x_{i1} = 7$ 加上速度分量 $v_{i1} = -2$ 等於 5，不超過上下限的範圍且屬於集合 D 。因此 $u_{i11} = 5$ 。更新 $D = \{1, 2, 3, 4, 6, 7, 8\}$ ，上下限不變。依上述規則可依序指定 $u_{i13} = 4$ 和 $u_{i14} = 8$ ，此時 $D = \{1, 2, 3, 6, 7\}$ ，下限 $d_{\min} = 1$ ，上限 $d_{\max} = 7$ 。計算維度 $o_5 = 6$ 的位置分量 $x_{i5} = 3$ 加上速度分量 $v_{i5} = 2$ 等於 5。由於 $5 \notin D$ ，因此計算 $\delta = 1$ ， $u_{i15} = 5 + 1 = 6$ 。更新 $D = \{1, 2, 3, 7\}$ ，上下限不變。再計算維度 $o_6 = 4$ 的位置分量 $x_{i6} = 2$ 加上速度分量 $v_{i6} = -1$ 等於 1。由於 1 小於下限 d_{\min} ，因此指定子水流的位置分量 u_{i16} 為下限值，也就是座標值 1。更新 D 為 $D = \{2, 3, 7\}$ ，下限 $d_{\min} = 8$ ，上限不變。接著計算維度 $o_7 = 7$ 的位置分量 $x_{i7} = 5$ 加上速度分量 $v_{i7} = -1$ 等於 4。由於 $4 \notin D$ ，因此計算 $\delta = 1$ ，

$u_{i15} = 4 - 1 = 3$ 。更新 $D = \{2, 7\}$ ，上下限不變。如此計算完所有維度，則可得到新的可行解。

Index	1	2	3	4	5	6	7	8	9
\tilde{X}_i	9	4	8	3	7	1	6	2	5
X_i	7	4	6	2	9	3	5	1	8
V_i	-2	0	-2	-1	2	2	-1	-1	3

$O = \langle 9 \ 1 \ 3 \ 5 \ 6 \ 4 \ 7 \ 8 \ 2 \rangle$

(a)

X_i	7	4	6	2	9	3	5	1	8
V_i	-2	0	-2	-1	2	2	-1	-1	3
U_{il}									9

$D = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$
 $d_{\min} = 1$
 $d_{\max} = 9$
 $\alpha = x_{i9} + v_{i9} = 11 \geq d_{\max}$
 $u_{i19} = d_{\max} = 9$

(b)

X_i	7	4	6	2	9	3	5	1	8
V_i	-2	0	-2	-1	2	2	-1	-1	3
U_{il}	5								9

$D = \{1, 2, 3, 4, 5, 6, 7, 8\}$
 $d_{\min} = 1$
 $d_{\max} = 8$
 $\alpha = x_{i1} + v_{i1} = 5 \in D$
 $u_{i11} = \alpha = 5$

(c)

圖3-3. CPA 法定義第一股子水流位置示意

Index	1	2	3	4	5	6	7	8	9
X_i	7	4	6	2	9	3	5	1	8
V_i	-2	0	-2	-1	2	2	-1	-1	3
U_{il}	5		4		8	6			9

(d)

$D = \{1, 2, 3, 6, 7\}$
 $d_{\min} = 1$
 $d_{\max} = 7$
 $\alpha = x_{i6} + v_{i6} = 5 \notin D$
 $\delta = 1$
 $u_{i16} = \alpha + \delta = 6$

X_i	7	4	6	2	9	3	5	1	8
V_i	-2	0	-2	-1	2	2	-1	-1	3
U_{il}	5		4	1	8	6			9

(e)

$D = \{1, 2, 3, 7\}$
 $d_{\min} = 1$
 $d_{\max} = 7$
 $\alpha = x_{i4} + v_{i4} = 1 \leq d_{\min}$
 $u_{i14} = d_{\min} = 1$

X_i	7	4	6	2	9	3	5	1	8
V_i	-2	0	-2	-1	2	2	-1	-1	3
U_{il}	5		4	1	8	6	3		9

(f)

$D = \{2, 3, 7\}$
 $d_{\min} = 2$
 $d_{\max} = 7$
 $\alpha = x_{i7} + v_{i7} = 4 \notin D$
 $\delta = 1$
 $u_{i16} = \alpha - \delta = 3$

X_i	7	4	6	2	9	3	5	1	8
V_i	-2	0	-2	1	2	2	-1	-1	3
U_{il}	5	7	4	1	8	6	3	2	9

(g)

$D = \emptyset$

圖 3-3. CPA 法定義第一股子水流位置示意 (續)

第二股以上之子水流

為了擴展搜尋的廣度及方向性並簡約分流演化的計算，以第一股子水流所設定的位置向量為基準，將其速度分量為正及為負的維度區分成兩個集合。將兩集合內的維度的座標值分別隨機設定其他子水流的位置向量座標值。

以前述方法設定第一股子水流的移步位置 U_{i1} 後，其速度向量可由其母水流 i 的位置向量 X_i 推算，

$$Y_{i1} = U_{i1} - X_i = [y_{i11} y_{i12} \dots y_{i1q}]。 \quad (3-7)$$

檢視速度分量正負值，區分出分量正值的位置座標集合 A 及其維度索引集合 I^A ，以及速度分量負者的位置座標集合 B 及其維度索引集合 I^B ：

$$\begin{aligned} I^A &= \{j | y_{i1j} \geq 0 \text{ and } j \in \{1, 2, \dots, q\}\} \\ A &= \{u_{i1j} | j \in I^A\} \\ I^B &= \{j | y_{i1j} < 0 \text{ and } j \in \{1, 2, \dots, q\}\} \\ B &= \{u_{i1j} | j \in I^B\}。 \end{aligned} \quad (3-8)$$

再將 A 和 B 內的座標值分別隨機排入子水流位置向量 I^A 和 I^B 的各維度完成子水流的位置向量指定。

第二股以上子水流位置向量座標設定的演算程序如下：

步驟 1：以式(3-7)計算第一股子水流的速度向量 Y_{i1} 。

步驟 2：以式(3-8)定義集合 I^A 、 $A = \{a_1, a_2, \dots, a_{|A|}\}$ 、 I^B 、和 $B = \{b_1, b_2, \dots, b_{|B|}\}$ 。

步驟 3：設定水流 i 分出的第二股以上子水流的位置向量。

步驟3.1：令 $k \leftarrow 2$ ， $j \leftarrow 1$ 。

步驟3.2：若 $j \in I^A$ 則由速度分量正值的位置座標集合 A 中隨機選取一個元素

a_α ， $\alpha = \text{rand}(|A|)$ 是由 1 到 $|A|$ 隨機選取一整數。令 $u_{ikj} \leftarrow a_\alpha$ ，並更

新 $A \leftarrow A - \{a_\alpha\}$ 。否則 $j \in I^B$ ，由 B 中隨機選取一個元素 b_β ，

$$\beta = \text{rand}(|B|), \text{ 令 } u_{ikj} \leftarrow b_\beta, \text{ 並更新 } B \leftarrow B - \{b_\beta\}。$$

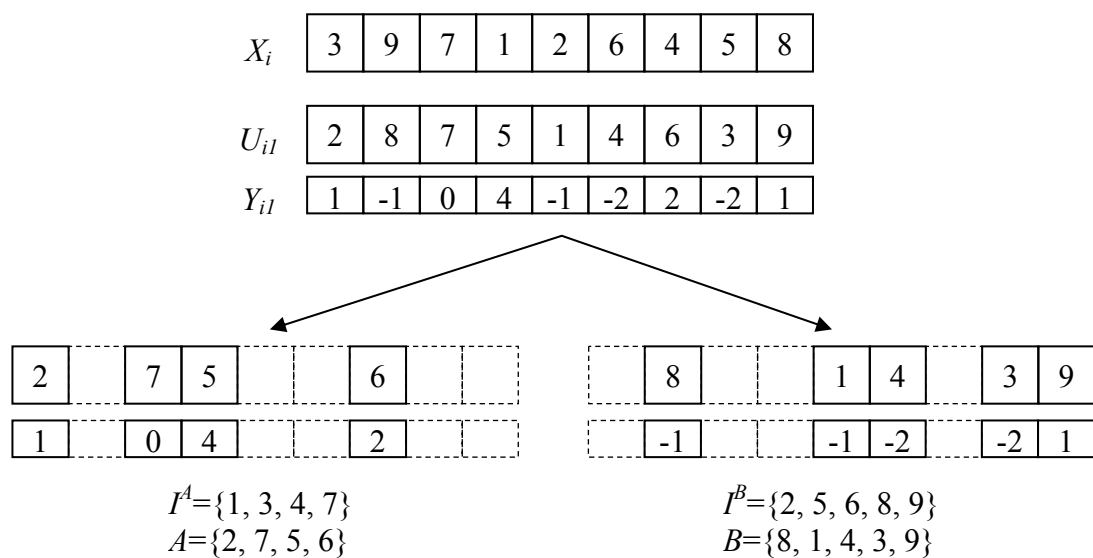
步驟3.3：更新 $j \leftarrow j+1$ 。若 $j > q$ ，則更新 $k \leftarrow k+1$ 到步驟 4；否則回到步驟

3.2 進行下一個座標值的設定。

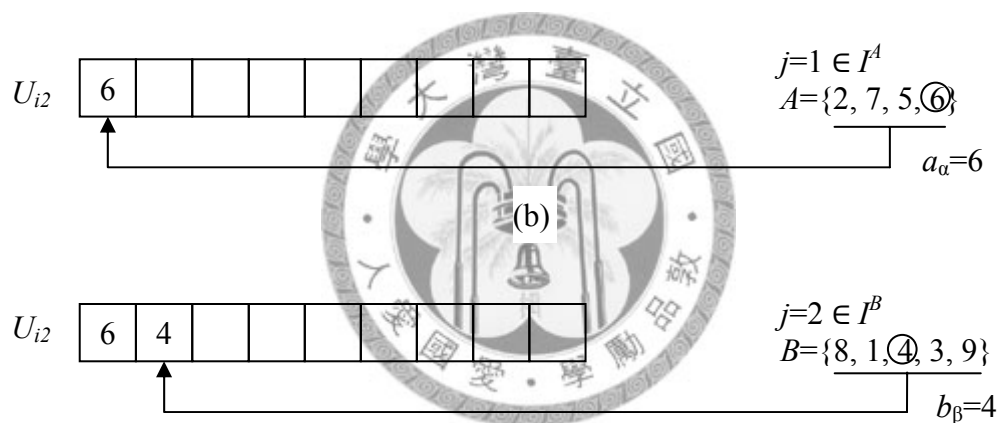
步驟 4：若 $k > n_i$ 則結束演算程序，否則回到步驟 3 進行下一股水流的位置設定。

如圖 3-4 為例，依式(3-8)可得速度分量為正的座標集合 $A = \{2, 5, 7, 6\}$ 和維度索引集合 $I^A = \{1, 3, 4, 7\}$ ，速度分量為負的座標集合 $B = \{8, 1, 4, 3, 9\}$ 和維度索引集合 $I^B = \{2, 5, 6, 8, 9\}$ 。開始依序指定第二股子水流的位置分量，首先維度 $1 \in I^A$ ，因此由 $\{2, 5, 7, 6\}$ 中任意選取一個座標值。此例選取 $a_4 = 6$ 指定給 u_{i21} ，並更新 $A = \{2, 5, 7\}$ 。接著維度 $2 \in I^B$ ，由 $\{8, 1, 4, 3, 9\}$ 中任意選取一個座標值。此例選取 $b_3 = 4$ 指定給 u_{i22} ，更新 $B = \{8, 1, 3, 9\}$ 。再來維度 $3 \in I^A$ ，由 $\{2, 5, 7\}$ 中任意選取一個座標值。此例選取 $a_2 = 5$ 指定至 u_{i23} ，更新 $A = \{2, 7\}$ 。如此依序指定各座標值直到 $A = \emptyset$ 且 $B = \emptyset$ 。依此產生出的子水流與第一股子水流的位置向量會有一定的歧異度，並且大致上能保持原有之方向性。計算時間也較為節省。

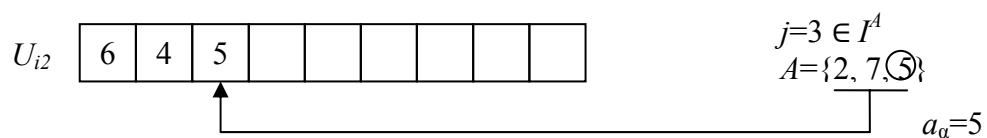
圖 3-5 是一種極端狀況，速度分量為正的座標集合 A 中僅有一個座標值 9 其餘都是負值。因此在決定第二股以上子水流時，座標 9 仍會維持在維度 5 上不會變動。此時速度分量為負的維度集合 I^B 會有較大的變動幅度。這種極端狀況的發生是由於第一股子水流的流動方向偏向單一維度，若依此流動方向進行移步所產生的位置會趨向一致性。第二股以上子水流的設計概念是以第一股子水流方向為主來產生鄰近方向的子水流位置。當第一股子水流的流動方向偏向單一時，延續此方向的移步也會偏向單一性。CPA 法的第二股子水流產生方式會使第二股子水流方向產生較大偏移。



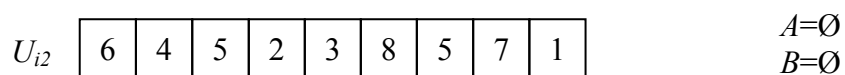
(a)



(c)



(d)



(e)

圖3-4. CPA 法定義第二股以上之子水流位置示意

X_i	2	8	7	5	1	4	6	3	9
U_{i1}	1	7	6	4	9	3	5	2	8
Y_{i1}	-1	-1	-1	-1	8	-1	-1	-1	-1

↓

U_{i2}	3	4	7	8	9	2	1	5	6
Y_{i2}	1	-4	0	3	8	-2	-5	2	-3

圖3-5. 第二股以上子水流位置可能發生的極端狀況

b. 子序列變動法(subtour shuffer method, SS)

物件排序優化問題中，排序解的優劣受物件排序的接續鏈結關係的影響頗大。因此本法在進行分流作業時，會保留母水流 i 部份物件間的鏈結關係。作法是只更動母水流位置向量的一段物件排列序列產生鄰近解，作為子水流的位置向量。本法最大的特色是不使用速度作為子水流位置設定的依據，而改用一指標向量 P_i 作為水流 i 各維度的座標變動是否可尋得較佳解的指引；亦即，若某個維度 j 的值變動可使目標值改善，則增加該維度的指標分量 p_{ij} ，使該維度有較大的機會變動尋得較佳的解。

採用本法時，水流初始指標分量設為 0 到 1 的隨機實數，因此指標向量的分量

$$p_{1j} = U(0,1), \quad j = 1, 2, \dots, q. \quad (3-9)$$

此為初始設定，之後隨水流演化，指標向量的分量值將不限於 0 到 1 之間。分流移步時，先依指標分量由大到小排出維度順位序列

$$O = \langle o_1 o_2 \dots o_q \rangle, p_{io_j} \geq p_{io_{j+1}}, j = 1, 2, \dots, q-1. \quad (3-10)$$

此序列在確認前指標分量排名前 n_i 個的維度索引，供 n_i 個子水流選定母水流水流 i

的位置向量變動中心維度。第 k 股子水流，子水流 k ，的位置座標只更動母水流 i 位置向量以維度 o_k 為中心的一小段序列，此段序列由維度 o_k 向左右隨機延伸一段範圍，此範圍不超過整個序列的一半。範圍

$$l = \text{RoundOff}(0.25 \cdot U(0,1) \cdot q) + 1 \quad (3-11)$$

以維度 o_k 為中心，分別向左右各延伸長度 l 的座標值，組成座標值集合 D 維度及索引集合 I^D 。當延伸的變動範圍超過第一個順位或超過最後一個順位時，依迴圈循環原則分由最後和第一順位接續延伸。因此設定子水流 k 時變動的維度集合

$$\begin{aligned} I^D &= \{j | j \in \{1, 2, \dots, q\} \text{ and } (|j - o_k| \leq l \text{ or } |j - q - o_k| \leq l \text{ or } |j + q - o_k| \leq l)\} \\ D &= \{u_{ikj} | j \in I^D\} \end{aligned} \quad (3-12)$$

將 D 內的座標值隨機排入 I^D 的各維度中，其餘維度的座標值則複製水流 i 的座標值，即完成子水流 k 的位置向量指定。

產生子水流位置後，若所得的目標值較母水流為佳，則表示變動 I^D 內維度的座標值有較高的機率使目標值變好，因此增加 I^D 裡各維度的指標分量值指定成子水流 k 的指標分量，使下一代次 I^D 裡各維度索引有較大的機率持續變動。因此

$$\begin{aligned} Z_{ik} &= [z_{ik1} z_{ik2} \dots z_{ikq}] \\ z_{ikj} &\leftarrow \begin{cases} p_{ij}, & \text{if } j \notin I^D \\ p_{ij} + U(0,1), & \text{else if } f(U_{ik}) < f(X_i) \\ p_{ij} - U(0,1), & \text{otherwise} \end{cases}, \end{aligned} \quad (3-13)$$

式中 Z_{ik} 是子水流 k 的指標向量，添加量是介於 0 到 1 的隨機實數。如此下一代次決定維度順位序列 O 時， I^D 以外的維度索引有可能可排到較前的順位而進行變動。

SS 法的演算流程如下：

步驟 1：依指標向量 P_i 的分量由大到小排序得到一維度索引序列

$$O = \langle o_1 o_2 \dots o_q \rangle, p_{io_j} \geq p_{io_{j+1}}, \forall j = 1, 2, \dots, q-1 \text{。令 } k \leftarrow 1 \text{。}$$

步驟 2：指定水流 i 分出的第 k 股子水流的位置向量及指標向量。

步驟 2.1：以式(3-11)計算單向延伸長度 l 。再以式(3-12)定義變動範圍的維度索引集合 I^D 和位置座標集合 $D = \{d_1, d_2, \dots, d_{|D|}\}$ 。

步驟 2.2：令 $j \leftarrow 1$ 。

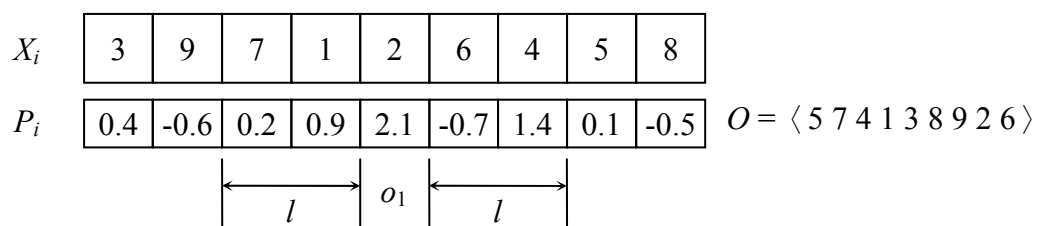
步驟 2.3：若 $j \in I^D$ 則由位置座標集合 D 中隨機選取一個元素 d_α ， $\alpha = \text{rand}(|D|)$ ，令 $u_{ikj} \leftarrow d_\alpha$ ，並更新 $D \leftarrow D - \{d_\alpha\}$ ；否則 $u_{ikj} \leftarrow x_{ij}$ 。

步驟 2.4：更新 $j \leftarrow j + 1$ 。若 $j > q$ 表示各位置分量皆指定完成，進到步驟 2.5，否則回到步驟 2.3 進行下一個座標值的設定。

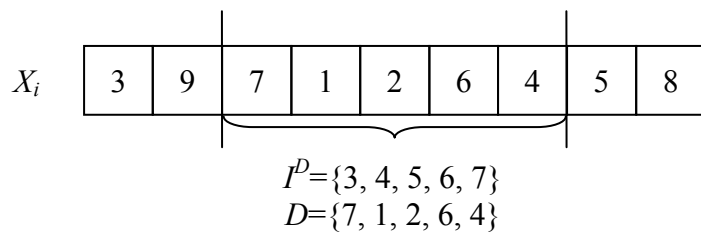
步驟 2.5：比較子水流 k 與母水流 i 位置向量的目標函數值，依式(3-13)設定子水流 k 的指標向量 Z_{ik} 。

步驟 3：更新 $k \leftarrow k + 1$ 。若 $k > n_i$ 則結束演算程序，否則回到步驟 2 進行下一股水流的位置設定。

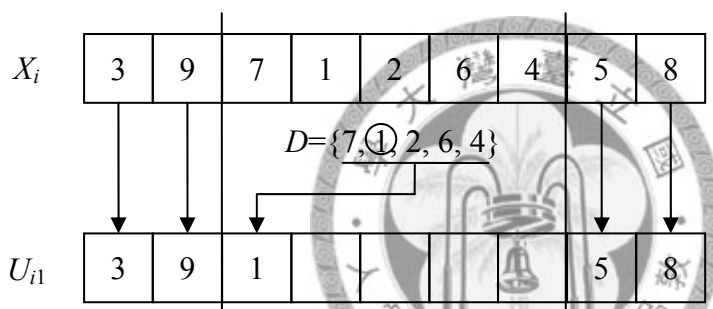
以圖 3-6 為例。首先依指標向量 P_i 的分量由大到小排列排出索引序列 $O = \langle 574138926 \rangle$ 。依式(3-11)設定單向變動範圍 $l = 2$ 。則子水流 1 以維度 $o_1 = 5$ 為中心，向左右延伸 2 個維度為變動範圍。因此維度索引集合 $I^D = \{3, 4, 5, 6, 7\}$ ，變動範圍內的水流 i 的座標值集合 $D = \{7, 1, 2, 6, 4\}$ 。接著依序指定子水流位置向量 U_{i1} 的各維度分量。因維度 1 和 2 都不屬於 I^D ，因此直接指定 $u_{i11} = x_{i1} = 3$ 和 $u_{i12} = x_{i2} = 9$ 。維度 $3 \in I^D$ ，由 $D = \{7, 1, 2, 6, 4\}$ 中任意選取到 $d_2 = 1$ 指定為 u_{i13} ，同時更新 $D = \{7, 2, 6, 4\}$ 。依此方法完成 U_{i1} 的各分量指定。位置向量指定完成後，複製母水流的指標向量 P_i 為子水流的指標向量 Z_{i1} 。本例假定子水流的目標函數直比母水流好，則在 Z_{i1} 中的維度 $\{3, 4, 5, 6, 7\}$ 分量分別增添一個介於 0 至 1 的隨機實數值，完成一股子水流的位置設定。



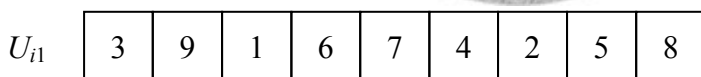
(a)



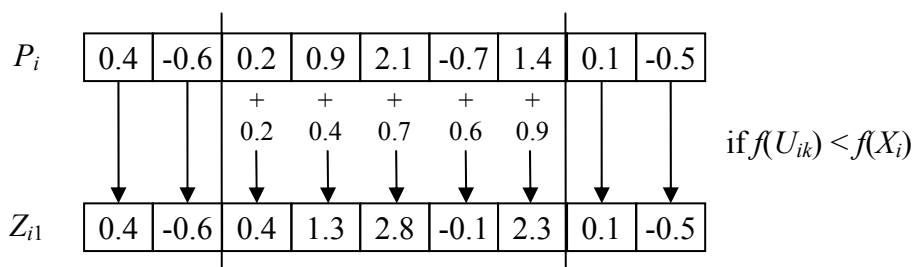
(b)



(c)



(d)



(e)

圖3-6. 水流 X_i 分出一股子水流 U_{i1}

c. 鏈結關係繼承法(link relationship inheritance method, LRI)

水流的位置向量代表一個排序解物件的排列序列。此序列由兩兩物件依序鏈結組成。本法考量當下水流 i 與母水流兩個位置向量內座標值鏈結的優劣關係，用以設定子水流的位置向量。圖 3-7 說明水流 i 及其母水流的兩座標值鏈結的優劣比較關係。將水流 i 的位置向量 X_i 內兩座標值鏈結視為一個集合 L ，母水流位置向量 \tilde{X}_i 內座標值鏈結視為另一個集合 \tilde{L} 。兩集合間的相對關係如圖 3-7(a)所示。當水流 i 的目標函數值較母水流好時，則表示 $L - \tilde{L}$ （圖 3-7(b)）內的鏈結關係有較大的可能使目標函數值變好，反之 $\tilde{L} - L$ 內的鏈結關係較佳。

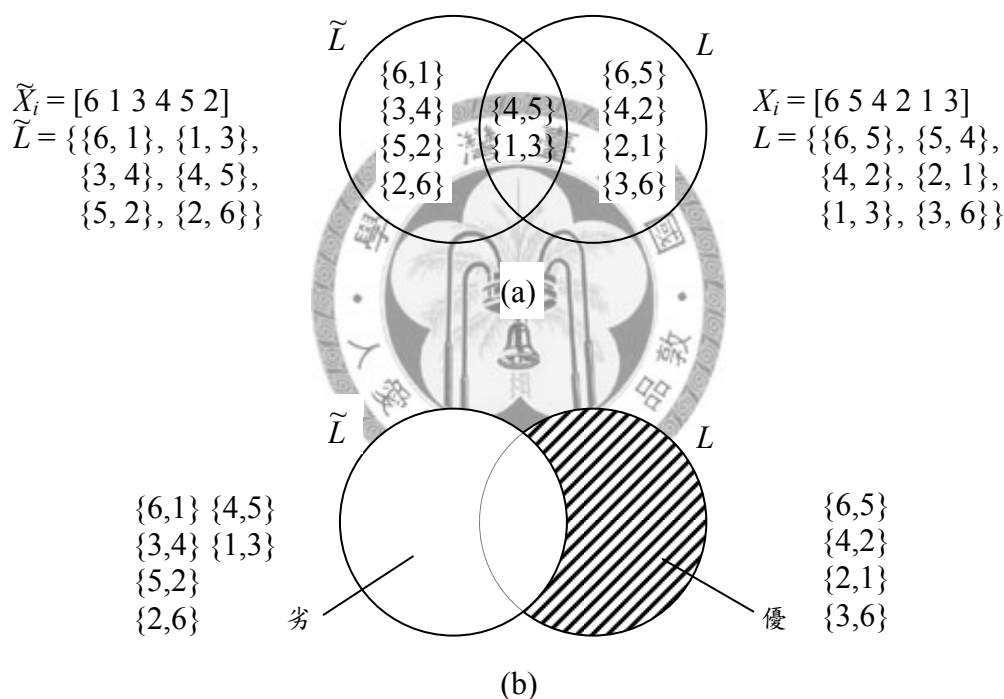


圖3-7. 水流 i 位置向量 X_i 與母水流位置向量 \tilde{X}_i 內物件鏈結關係分佈示意

本法的子水流位置向量座標值的設定是一建構式作法。由第一個座標值開始設定，過程會建立候選的座標值集合，由其中隨機選取。候選集合內的座標以繼承 $L - \tilde{L}$ 或 $\tilde{L} - L$ 中的優良鏈結關係為主要考量。若無優良鏈結關係時（ $L \subseteq \tilde{L}$ 或 $\tilde{L} \subseteq L$ ）則以驅避非優良的鏈結關係為考量。在序列的兩兩鏈結特性下，與前一個座標值鏈結的優良候選座標最多只有兩個：來自 X_i 或 \tilde{X}_i ；非優良座標最多也只有

兩個。因此候選座標集合以優良座標為優先考量，當無優良座標時，則以尚未排入的座標扣除非優良座標組成。因此假設建構過程擬設定水流 i 的子水流 k 位置向量 U_{ik} 的座標值 u_{ikj} 接續已設定的 $u_{ik(j-1)}$ ，若 X_i 較 \tilde{X}_i 佳，優良座標來自 X_i 中與 $u_{ik(j-1)}$ 相鄰者（左鄰或右接），否則來自 \tilde{X}_i 。且設

$$\text{indexOf}(A, \alpha) = k, \text{ if } a_k = \alpha \quad (3-14)$$

是座標向量 A 中座標值 α 的索引運算子。若

$$\begin{aligned} r &= \text{indexOf}(X_i, u_{ik(j-1)}) \\ s &= \text{indexOf}(\tilde{X}_i, u_{ik(j-1)}) , \end{aligned} \quad (3-15)$$

$$\begin{aligned} r' &= \begin{cases} q, & \text{if } r = 1 \\ r-1, & \text{otherwise} \end{cases} \\ r'' &= \begin{cases} 1, & \text{if } r = q \\ r+1, & \text{otherwise} \end{cases} \\ s' &= \begin{cases} q, & \text{if } s = 1 \\ s-1, & \text{otherwise} \end{cases} \\ s'' &= \begin{cases} 1, & \text{if } s = q \\ s+1, & \text{otherwise} \end{cases} , \end{aligned} \quad (3-16)$$

且求解目標是望小，而當下尚未排入的座標集合是 D ，則

$$H = \begin{cases} \{x_{ir'}, x_{ir''}\}, & \text{if } f(X_i) < f(\tilde{X}_i) \\ \{\tilde{x}_{is'}, \tilde{x}_{is''}\}, & \text{otherwise} \end{cases} , \quad (3-17)$$

$$W = \begin{cases} \{\tilde{x}_{is'}, \tilde{x}_{is''}\}, & \text{if } f(X_i) < f(\tilde{X}_i) \\ \{x_{ir'}, x_{ir''}\}, & \text{otherwise} \end{cases} . \quad (3-18)$$

H 是使目標函數值較佳的候選座標集合， W 是較差的候選座標集合。因此

$$C' = (H - W) \cap D , \quad (3-19)$$

是設定座標值 u_{ikj} 時可優先選擇的座標集合。因 C' 可能是空集合，座標值 u_{ikj} 的候選座標集合

$$C = \begin{cases} C', & \text{if } C' \neq \emptyset \\ D - W, & \text{else if } D - W \neq \emptyset \\ D, & \text{otherwise} \end{cases} \quad (3-20)$$

當 $C' \neq \emptyset$ 時，逕自挑選尚未排入的優良座標。 $C' = \emptyset$ 表示 $H = W$ 或 H 扣除 W 後剩下的座標值都已排入序列中，沒有優良座標可供選取。此時若 $D - W \neq \emptyset$ ，則改採當下尚未排入的座標集合 D 扣除非優良的座標 (W) 為候選座標集合。若 $D - W$ 為空集合，不得已則以未排入的座標 D 為候選座標集合 C 。

LRI 法決定一股子水流位置向量的演算流程如下：

步驟 1：建立待排入序列的座標候選集合 $D = \{1, 2, \dots, q\} = \{d_1, d_2, \dots, d_q\}$ 。令 $k \leftarrow 1$ ， $j \leftarrow 1$ 。

步驟 2：由座標候選集合 D 中隨機選取一個元素 d_α ， $\alpha \leftarrow \text{rand}(|D|)$ ，令 $u_{ikj} \leftarrow d_\alpha$ ，更新 $D \leftarrow D - \{d_\alpha\}$ 。

步驟 3：依式(3-17)至式(3-20)建立座標集合 H 、 W 、和 C 。 $j \leftarrow j + 1$ 。

步驟 4：假設 $C = \{c_1, c_2, \dots, c_{|C|}\}$ ，由候選位置座標集合 C 中隨機選取一個元素 c_α ， $\alpha \leftarrow \text{rand}(|C|)$ ，令 $u_{ikj} \leftarrow c_\alpha$ ，更新 $D \leftarrow D - \{c_\alpha\}$ 。

步驟 5：更新 $j \leftarrow j + 1$ ，若 $j > q$ ，結束演算程序，否則回到步驟 3 繼續下一個座標的設定。

以圖 3-8 為例。先建立待排入序列的座標候選集合 $D = \{1, 2, \dots, 9\}$ ，從中隨機選擇一個座標為序列起始點，此例為座標 4。更新 $D = \{1, 2, 3, 5, 6, 7, 8, 9\}$ 。假設本例中水流 i 的目標函數值 $f(X_i)$ 優於母水流，則依式(3-17)和(3-18)建立座標集合 $H = \{6, 5\}$ 、 $W = \{6, 2\}$ 。再依式(3-19)和式(3-20)，可得 $C' = \{5\}$ ， $C = \{5\}$ 。選取 C 中的座標值 5 接續座標 4 之後。更新 $D = \{1, 2, 3, 6, 7, 8, 9\}$ 。再以座標 5 為主依式(3-17)和(3-18)得 $H = \{4, 8\}$ 、 $W = \{3, 6\}$ 。再依式(3-19)和(3-20)，得 $C' = \{8\}$ ， $C = \{8\}$ 。設定座標值 8 接續座標 5 之後。更新 $D = \{1, 2, 3, 6, 7, 9\}$ 。依上述原則直到座標 7 排入序列中，此時

$D = \{1, 2, 6\}$ 。依式(3-18)和(3-19)建立座標集合 $H = \{9, 1\}$ 、 $W = \{1, 3\}$ 。再由式(3-19)得 $C' = \emptyset$ ，則由式(3-20)得 $C = \{2, 6\}$ 。則再由 C 中任選一個座標，本例選出座標 6，排入序列中，更新 $D = \{1, 2\}$ 。重覆利用鏈結關係找尋下一可排入的座標，直到座標候選集合 D 中的所有座標都排入序列，即完成子水流的位置向量 U_{il} 的指定。



\tilde{X}_i	5	6	4	2	9	8	1	7	3
X_i	3	9	7	1	2	6	4	5	8

$f(X_i) < f(\tilde{X}_i)$

(a)

U_{il}	4	5							
----------	---	---	--	--	--	--	--	--	--

$D=\{1, 2, 3, 5, 6, 7, 8, 9\}$
 $H=\{6, 5\}$
 $W=\{6, 2\}$
 $C'=(H-W) \cap D=\{5\}$
 $C=C'=\{5\}$

(b)

U_{il}	4	5	8						
----------	---	---	---	--	--	--	--	--	--

$D=\{1, 2, 3, 6, 7, 8, 9\}$
 $H=\{4, 8\}$
 $W=\{3, 6\}$
 $C'=(H-W) \cap D=\{8\}$
 $C=C'=\{8\}$

(c)

U_{il}	4	5	8	3	9	7	6		
----------	---	---	---	---	---	---	---	--	--

$D=\{1, 2, 6\}$
 $H=\{9, 1\}$
 $W=\{1, 3\}$
 $C'=\emptyset$
 $C=D-W=\{2, 6\}$

(d)

U_{il}	4	5	8	3	9	7	6	2	1
----------	---	---	---	---	---	---	---	---	---

(e)

圖3-8. LRI 法產生水流 i 的子水流位置向量 U_{il} 範例

總結而言：CPA 法仿照水流在連續空間流動的方式，以目前位置加上速度來設定子水流的位置。在物件排序問題的基本限制下，透過速度分量絕對值的大小來決定各維度設定座標值的優先順位。優先者先確定座標值，後續者座標值視計算結果設成尚未指派且最接近的座標值。SS 法利用一指標向量來決定各子水流分流時變動的一段子序列，藉由變動子序列內的座標來產生子水流的位置。LRI 法依據欲分流的水流和其母水流中目標函數值較好的一股水流的位置向量，繼承較好的座標鏈結關係並趨避較差者以設定子水流的位置。就設計概念而言，SS 法和 LRI 法有考量排序問題座標鏈結的特性。

質量設定

決定水流 i 分出的 n_i 個子水流位置後，若分出的子水流數大於一股，則依目標函數值之優劣分配各子水流的質量。目標函數值越好者，應得越大的質量。為因應不同求解目標的問題，分配質量時以目標函數值的改善量 Δf_k 來計算分配權重。以望小問題來說，改善量

$$\Delta f_k = f(X_i) - f(U_{ik}), k = 1, 2, \dots, n_i. \quad (3-21)$$

先將所有子水流的目標函數值改善量皆平移成正值，

$$\begin{aligned} \Delta f'_k &= \Delta f_k + \mu \\ \mu &= \max_{\forall k} \{\Delta f_k\} - 2 \min_{\forall k} \{\Delta f_k\}. \end{aligned} \quad (3-22)$$

式中 μ 是平移量。平移結果最小值為 $\max_{\forall k} \{\Delta f_k\} - \min_{\forall k} \{\Delta f_k\}$ ，最大值為最小值的兩倍，

都是正值。平移完成後即可依 $\Delta f'_k$ 分配質量。水流 k 分配的質量

$$w_{ik} = \frac{\Delta f'_k}{\sum_{s=1}^{n_i} \Delta f'_s} \cdot m_i, k = 1, 2, \dots, n_i. \quad (3-23)$$

這種分配法雖然較為繁複，但可適用於不同求解目標（望大、望小、或望目）的問題。重要的是以各子水流的目標函數值改善量決定質量分配的比例。

例如：若水流 i 的分支數是 3，分流前質量為 5，且分流後新水流代表的解 U_{i1} 、

U_{i2} 、 U_{i3} 目標函數的改善量分別為 5、-10、10。經過式(3-22)和式(3-23)平移後的目標函數值改善量分別為 35、20、40，則新水流的質量分別是 $w_{i1} = \left(\frac{35}{35+20+40} \right) \times 5$ 、 $w_{i2} = \left(\frac{20}{35+20+40} \right) \times 5$ 、 $w_{i3} = \left(\frac{40}{35+20+40} \right) \times 5$ 。

3.5 匯流作業

當多股水流位置在鄰近範圍內須進行合併時，保留此範圍內目標函數值較佳的水流，將範圍內其他水流合併。鄰近範圍的界定是以水流位置的序列相似度而定。WFA4SP 設有合併相似度底限 β ，當時兩水流位置的序列相似度大於 β 視為鄰近。

TSP 的解是一個路徑迴圈，因此如 [12345] 和 [23451] 兩解都代表相同路徑。且本研究探討的是對稱型的 TSP，因此解的序列反轉仍是相同的解，即 [12345] 和 [54321] 兩解亦代表相同路徑。因此兩個解的城市繞行序列的相似度計算，須進行循環比對外，也須以其中之一的反轉序列進行循環比對。比對過所有的循環序列後，取座標相同數最高者除以問題維度 q 為兩解的位置向量相似度。

循環比對相似度計算法流程如下：

步驟 1：定義其中一條序列為基準序列 $E = [e_1 e_2 \dots e_q]$ ，另一條為 $S = [s_1 s_2 \dots s_q]$ 。

S 的循環路徑集合 $\hat{S} = \{\hat{S}_1, \hat{S}_2, \dots, \hat{S}_q\}$ ，其中 $\hat{S}_k = [\hat{s}_{k1} \hat{s}_{k2} \dots \hat{s}_{kq}]$ ，

$\hat{s}_{kj} = s_{\text{mod}(k+j,q)-1}$ 。另定義序列 S 的反轉序列 $R = [r_1 r_2 \dots r_q]$ ，其中 $r_j = s_{q-j+1}$ 。

R 的循環路徑集合 $\hat{R} = \{\hat{R}_1, \hat{R}_2, \dots, \hat{R}_q\}$ ，其中 $\hat{R}_k = [\hat{r}_{k1} \hat{r}_{k2} \dots \hat{r}_{kq}]$ ，

$\hat{r}_{kj} = r_{\text{mod}(k+j,q)-1}$ 。令序列相似度 $s \leftarrow 0$ ， $i \leftarrow 1$ 。

步驟 2：比對 S 和 R 兩循環路徑集合中第 i 條循環路徑和基準序列 E 的維度相同數。

步驟 2.1：令 $j \leftarrow 1$ ， $\rho \leftarrow 0$ ， $\sigma \leftarrow 0$ ， $s \leftarrow 0$ 。

步驟2.2：若 $\hat{s}_{ij} = e_j$ ，則 $\rho \leftarrow \rho + 1$ 。若 $\hat{r}_{ij} = e_j$ ，則 $\sigma \leftarrow \sigma + 1$ 。

步驟2.3：更新 $j \leftarrow j + 1$ 。若 $j > q$ 則進入，否則回到步驟 2.2。

步驟 3：更新 $s \leftarrow \max\{s, \rho, \sigma\}$ ， $i \leftarrow i + 1$ 。若 $i > q$ 則更新 $s \leftarrow \frac{s}{q}$ ，結束演算程序，

否則回到步驟 2。

兩位置向量 E 和 S 運算子 $\text{Similarity}(E, S)$ 演算流程如下：

Input:

$E = [e_1 e_2 \dots e_q]$: 比對序列 1

$S = [s_1 s_2 \dots s_q]$: 比對序列 2

Output:

s : 相似度

Procedure: Similarity

begin

$s \leftarrow 0$

for $i = 1$ to q do //計算 S 的循環路徑和 S 反轉後的循環路徑與 E 的相似度

$forwardCount \leftarrow 0$

$reverseCount \leftarrow 0$

for $j = 1$ to q do

if $e_j = s_{\text{mod}(i+j, q) - 1}$ then

$forwardCount \leftarrow forwardCount + 1$

end if

if $e_{n-1-j} = s_{\text{mod}(i+j, q) - 1}$ then

$reverseCount \leftarrow reverseCount + 1$

end if

end for

if $forwardCount > s$ then

$s \leftarrow forwardCount$

end if

if $reverseCount > s$ then




```

         $s \leftarrow reverseCount$ 
    end if
end for
return  $s / q$ 
end

```

匯流作業執行時由尚未和其他水流匯流的水流集合 G 中任選一股水流 k^* 作為基準序列，其位置向量 X_{k^*} 和 G 中水流位置相似度大於 β 的水流組成一匯流集合

$$G' = \{i \in G \text{ and } \text{Similarity}(X_{k^*}, X_i) > \beta\} \quad (3-24)$$

若 G' 中目標函數值最佳者為中心水流 i^* ，合併時更新質量為

$$M_{i^*} \leftarrow \sum_{i \in G'} M_i, \quad i^* = \arg \min_{i \in G'} \{f(X_i)\} \quad (3-25)$$

其餘水流質量併入中心水流後從水流集合 G 中移除。將中心水流 i^* 加入匯流結果保留的水流集合 Ω 內。將 G' 內的水流合併完成後，如 G 內仍有水流，再進行下一個匯流集合的組成和合併作業，直到 G 為空集合為止。最後的 Ω 集合即是匯流後保留的水流編號集合。

匯流的演算程序如下：

步驟 1：定義尚未和其他水流匯流的水流集合 $G = \{1, 2, \dots, N_t\}$ 為所有水流編號的集合，且令匯流後水流集合 $\Omega = \emptyset$ 。

步驟 2：從 G 任取一股水流 k^* 。

步驟 3：找出 G 中位置向量和 X_{k^*} 的相似度大於 β 的水流組成集合

$$G' = \{i \in G \text{ and } \text{Similarity}(X_{k^*}, X_i) > \beta\}。$$

步驟 4：將 G' 內的水流進行合併。

步驟 4.1：找出 G' 中目標函數值最好的水流 $i^* \leftarrow \arg \min_{i \in G'} \{f(X_i)\}$ 。

步驟4.2：以式(3-25)更新水流 i^* 質量。

步驟4.3：將 i^* 加入匯流後水流集合內， $\Omega \leftarrow \Omega \cup \{i^*\}$ ，並更新 $G \leftarrow G - G'$ 。

步驟4.4：如果 $G' = \emptyset$ 結束演算，否則回到步驟2。

3.6 蒸發作業

分流移步和匯流作業結束後執行蒸發作業。將匯流後的水流集合中每股水流皆蒸散部份水流質量到大氣中。水流 i 蒸發後的質量變化為

$$\begin{aligned} M_i &\leftarrow M_i - \Delta M_i \\ \Delta M_i &= \gamma M_i \end{aligned}, i \in \Omega, \quad (3-26)$$

式中 γ 是水氣蒸發率。同時更新大氣中的水氣含量

$$\sigma \leftarrow \sigma + \sum_{i=1}^{|\Omega|} \Delta M_i. \quad (3-27)$$

蒸發後若

$$M_i < \frac{M_0}{\eta}, \quad (3-28)$$

式中 η 是總水流數上限，判定水流 i 的質量不足以使水流繼續進行搜尋。將這些水流的質量蒸散於大氣中，更新水氣含量

$$\sigma \leftarrow \sigma + \sum_{i \in \Omega \text{ and } M_i < \frac{M_0}{\eta}} M_i, \quad (3-29)$$

然後將這些水流刪除以更新匯流後水流集合

$$\Omega \leftarrow \Omega - \left\{ i \mid i \in \Omega \text{ and } M_i < \frac{M_0}{\eta} \right\}. \quad (3-30)$$

蒸發作業後更新水流位置向量集合

$$\mathbf{X} \leftarrow \{X_i \mid i \in \Omega\}, \quad (3-31)$$

以及更新目前水流數

$$N_t \leftarrow |\Omega| \quad (3-32)$$

3.7 降水作業

降水作業的目的是將大氣中的水氣再次降於地面，開發新的搜尋區域以跳脫區域最佳解。降水的情形分為自然降水和強制降水兩種。若大氣中的水氣含量相對於原始水流質量的比例

$$\frac{\sigma}{M_0} > \rho \quad (3-33)$$

時水氣視為飽和，即執行自然降水作業。降水後的水流總數須在總水流數上限內，因此降水的水流數

$$K = \min\{N_t, \eta - N_t\} \quad (3-34)$$

降水以當下水流集合中隨機選取 K 股水流偏移位置產生，產生法後述。新產生的降水水流 i 質量

$$M'_i \leftarrow \frac{\sigma}{K}, i=1,2,\dots,K, \quad (3-35)$$

式中 σ 是目前的水氣質量。更新水流集合

$$\mathbf{X} \leftarrow \mathbf{X} \cup \{X'_i\}, i=1,2,\dots,K \quad (3-36)$$

式中 X'_i 是第 i 股降水的水流位置向量，由某地表水流位置偏移產生。

上述是自然降水作業，強制降水發生於特定情況。當所有水流都因式(3-28)質量小於最低值全數蒸發於大氣時進行強制降水，產生單一股水流。WFA4SP 以目前搜尋到的最佳解的位置向量進行位置偏移產生新水流位置向量。質量為初始質量，

$$M'_1 \leftarrow M_0 \quad (3-37)$$

位置向量

$$X'_1 \leftarrow X_{opt} \quad (3-38)$$

式中 X_{opt} 是目前最佳的水流位置向量。降水後得水流向量集合

$$\mathbf{X} \leftarrow \{X'_1\} \quad (3-39)$$

且

$$N_t = 1 \quad (3-40)$$

Yang 和 Wang (2007)的 WFA 降水的位置是參考現有水流的位位置向量任意偏移一定範圍設定。WFA4SP 求解物件排序問題時，序列上的各維度座標若任意變動易變成難以修補的不可行解。因此 WFA4SP 就原有水流 i 的位位置向量 X_i 選取一段序列變動產生合理偏移位置作為降水位置。選取序列時隨機設定任意一點為分段點 c ，隨機選取由 c 向左或向右的序列組成變動的維度集合

$$I^D = \begin{cases} \{1, 2, \dots, c\}, & \text{if } U(0,1) \leq 0.5 \\ \{c+1, c+2, \dots, q\}, & \text{otherwise} \end{cases}, \quad (3-41)$$

並設定相對的位位置向量座標集合

$$D = \{x_{ij} | j \in I^D\} = \{d_1, d_2, \dots, d_{|I^D|}\} \quad (3-42)$$

將 D 內的座標值隨機排入 I^D 內的各維度中即完成位位置向量偏移。

水流位位置偏移演算流程如下：

步驟 1：以式(3-41)和(3-42)定義集合 I^D 、 D 。

步驟 2：建構由水流 i 位位置偏移設定的降水水流 i 位位置向量 $X'_i = [x'_{i1} x'_{i2} \dots x'_{iq}]$ 。

步驟2.1：令 $k \leftarrow 1$ 。

步驟2.2：若 $k \notin I^D$ 則 $x'_{ij} \leftarrow x_{ij}$ 。若 $k \in I^D$ 則由位位置座標集合 D 中隨機選取一個

元素 d_α ， $\alpha = \text{rand}(|D|)$ ，令 $x'_{ij} \leftarrow d_\alpha$ ，並更新 $D \leftarrow D - \{d_\alpha\}$ 。

步驟2.3：更新 $k \leftarrow k+1$ 。若 $k > q$ 則結束演算程序，否則回到步驟 2.2。

以圖 3-9 為例，首先任意選取一個分段點 $c=3$ ，由分段點區分水流 i 的位置向量 X_i 為左右兩段序列。隨機任選其中一段。本例中選取右方的序列，以式(3-41)和(3-42)得 $I^D = \{4, 5, 6, 7, 8, 9\}$ 和 $D = \{7, 5, 2, 3, 1, 9\}$ 。依序指定偏移後的位置向量 X'_i 的各分量。當 $j \leq 3$ 時， $x'_{ij} = x_{ij}$ 。當 $j > 3$ 時，由 D 中任選一個座標排入序列中，完成偏移後的位置向量 X'_i 的座標設定。

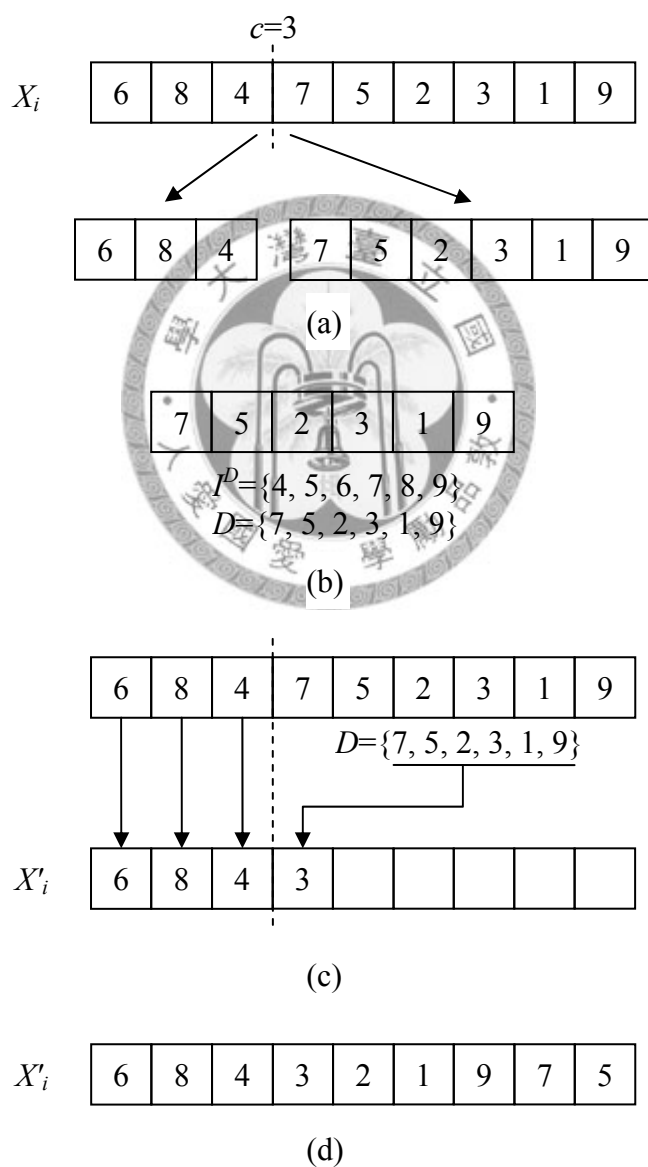


圖3-9. 降水水流 i 位置向量偏移演算範例

3.8 WFA4SP 整體演算程序

經初始設定後，WFA4SP 反覆進行分流移步、匯流、蒸發、和降水四項作業，每代次演化得更新目前搜尋到的最佳解

$$\begin{aligned} X_{opt} &\leftarrow X_{i^*} \\ i^* &\leftarrow \arg \min_{\forall i} \{f(X_i)\} \end{aligned} \quad (3-43)$$

停止條件可設定演化代次上限、目標函式呼叫次數上限、和求解時間上限，視需要選擇多項或其中一項。和停止條件相關的數學符號定義如下：

\bar{t} ：演化代次上限。

\bar{F} ：目標函式呼叫次數上限。

\bar{T} ：求解時間上限。

整個 WFA4SP 求解一般性物件排序優化問題的流程如下：

步驟 1：設定 WFA4SP 初始參數及停止條件。初始參數包括初始單股水流質量 M_0 、水流分流數上限 \bar{n} 、水流合併相似度 β 、每代次水流質量蒸發率 γ 、總水流數上限 η 、水氣飽和度 ρ 。視需要設定停止條件上限 \bar{t} 、 \bar{F} 、和 \bar{T} 。令目前演算代次 $t \leftarrow 0$ ，目前目標函式呼叫次數 $F \leftarrow 0$ ，求解時間 $T \leftarrow 0$ 。

步驟 2：以隨機方式設定初始解位置向量 X_1 和初始解的母水流位置向量 \tilde{X}_1 。

步驟 3：令代次起始時間 $T_{start} = T_{Now}$ 。

步驟 4：進行分流移步作業。

步驟 5：進行匯流作業。

步驟 6：進行蒸發作業。

步驟 7：進行降水作業。

步驟7.1：若水氣達飽和，即 $\frac{\sigma}{M_0} > \rho$ 則執行自然降水作業。

步驟7.2：若 $N_t = 0$ 則執行強制降水作業。

步驟8：以式(3-43)更新目前搜尋到的最佳解 X_{opt} 。更新目前目標函式呼叫次數 F 、更新目前演算代次 $t \leftarrow t+1$ 、和求解時間 $T \leftarrow T + T_{now} - T_{start}$ 。

步驟9：檢查停止條件。

步驟9.1：若啟用演化代次上限為停止條件且 $t \geq \bar{t}$ 則結束演算程序，否則進行步驟9.2 繼續查驗。

步驟9.2：若啟用目標函式呼叫次數上限為停止條件且 $F \geq \bar{F}$ 則結束演算程序，否則進行步驟9.3 繼續查驗。

步驟9.3：若啟用求解時間上限為停止條件且 $T \geq \bar{T}$ 則結束演算程序，否則返回步驟3 進行下一代次搜尋。

3.9 小結

WFA4SP 演算流程的分流移步、匯流、蒸發、降水四項演算作業的細節及功能為：

- (1) 分流移步：分出多個代理人向多方搜尋，
- (2) 匯流：縮減在同一區域搜尋的代理人數量，匯聚搜尋能量，
- (3) 蒸發：聚積水氣以供降水，
- (4) 降水：開發新搜尋區域。

本研究所開發出的 WFA4SP 以 WFA 為基，依照水流物理特性設定分流數。當水流的目標函數值改善率高時集中搜尋能量。此外，為符合物件排序優化問題的問題限制，分別提出 CPA、SS、和 LRI 三種分流法，設定分流移步後的子水流位置向量。位置向量決定後依目標函數值的改善量分配水流質量，可適用於不同求解目標的優化問題。水流的位置向量和分配的質量決定後進行匯流作業程序。匯

流時採用循環比對法來比較兩水流位置向量的相似度，將位置相似度高於門檻者合併。蒸發作業會蒸散部份水流質量到大氣中，同時可剔除質量太小無力進行搜尋的水流。降水作業時配合物件排序優化問題的問題限制，以原地面水流位置向量偏移法，產生新的可行解增加搜尋範圍。



第4章 仿水流優化演算法求解系統及範例驗證

本章展示本研究為求解物件排序優化問題開發的 WFA4SP 求解系統。為驗證所提演算法的效能，以 TSPLIB 中的標竿問題為測試範例，分別測試 WFA4SP 提出的三種分流移步法的求解結果。同時，也比較、分析不同規模問題的測試結果並與典型 GA 的求解結果比較。

4.1 仿水流離散優化演算法求解系統

本研究依據前一章提出的 WFA4SP 使用 Microsoft Visual Studio 2005 為程式開發工具，在 Microsoft .NET Framework 平台下，以 C# 程式語言開發一套 WFA4SP 的排序優化問題求解系統(Water Flow-like Algorithm Optimization System for Sequencing Problems，簡稱 WFAOS4SP)。

圖 4-1 是求解系統的操作介面，分為五大區塊。區塊 A 是 WFA4SP 的各種參數設定區域。須設定的參數包括初始單股水流質量、水流分流數上限、水流合併相似度、每代次水流質量蒸發率、總水流數上限、和水氣飽和度。區塊 B 供程式執行及停止條件設定。可設定執行代次上限、呼叫目標函式次數上限、和求解時間上限等三種停止條件。區塊 C 以圖形顯示目前求得之最佳物件排序解。若是 TSP 則顯示求得的最佳路徑。區塊 D 顯示求解過程中最佳目標函數值的變化折線圖，包括目標函數值隨求解時間變化折線圖、目標函數值對代次關係折線圖、目標函數值對目標函式呼叫次數折線圖、以及水流數隨代次變化折線圖，如圖 4-2 所示。區塊 E 展示搜尋過程的相關資訊，包括目前代次、呼叫目標式次數、目前水流數、平均水流數、降水次數、已知全域最佳解、目前搜尋到的最佳解及其求得的代次、和目前最佳解的路徑序列。

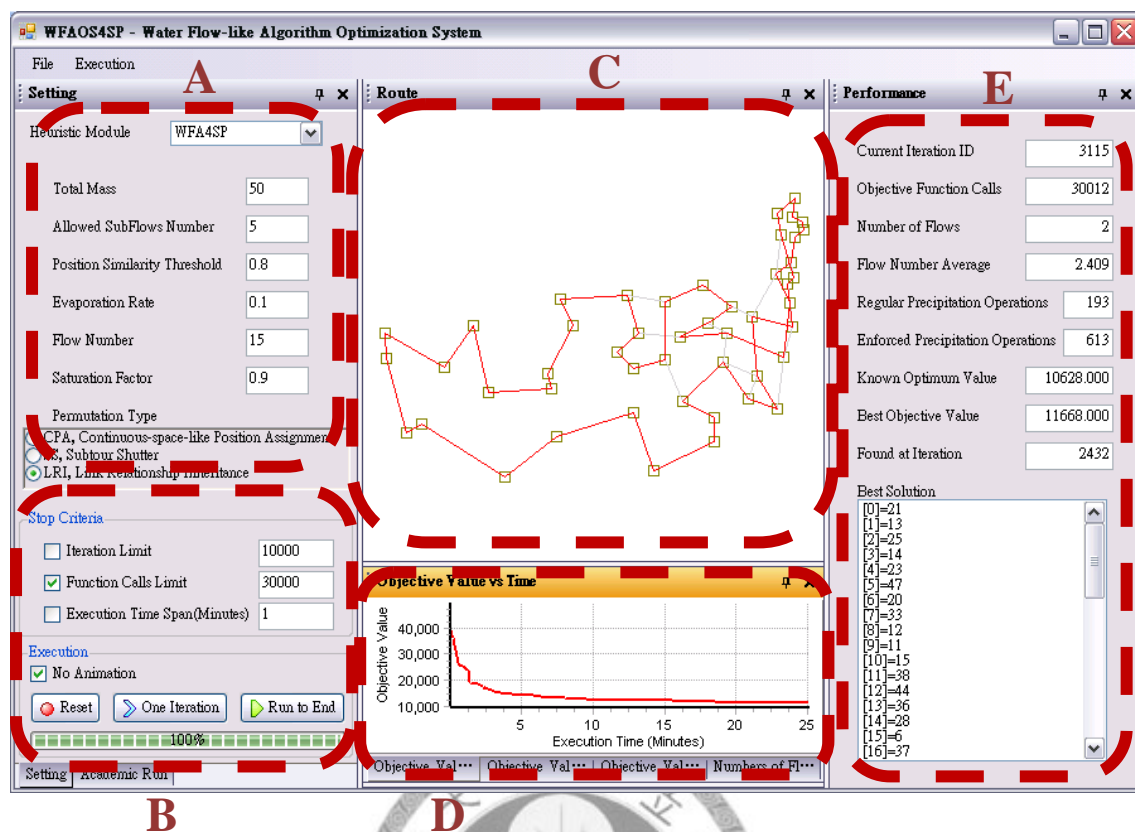


圖4-1. WFA4SPDOS 介面

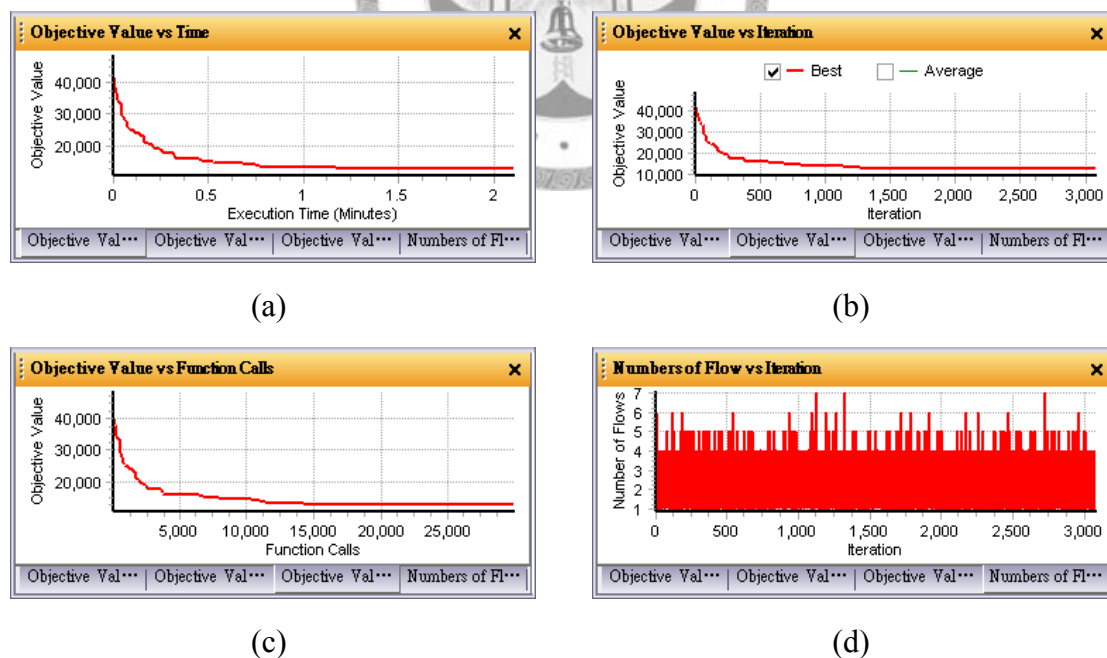


圖4-2. WFA4SPDOS 求解過程數值變化折線圖

區塊 A 和 B 有另一頁籤可設定批次求解，如圖 4-3 所示。批次求解功能可依選定的範例及設定的求解模式自動進行多次求解，並將所使用的參數設定、每

次求解結果的數據資料、數據變化折線圖、求解結果的路徑圖、以及批次求解統計結果等數據及圖形資料直接記錄在設定的 Microsoft Excel 檔案中。可選擇是否要輸出求解過程產生的折線圖及各演算代次的目標函式求解次數和所耗時間。供進一步分析和報表輸出使用。

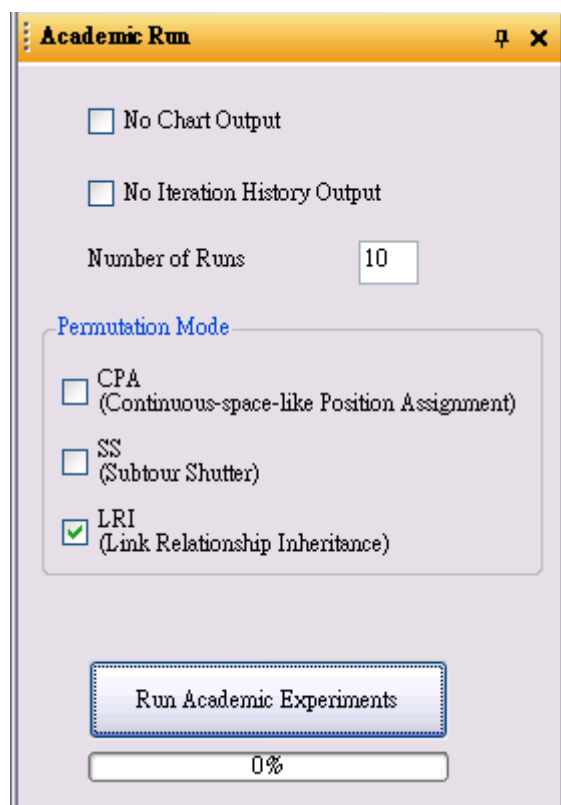


圖4-3. 批次求解介面

4.2 系統驗證分析

為驗證本研究提出之 WFA4SP 演算法，本節以 WFAOS4SP 求解 TSPLIB 內的標竿問題。測試平台使用的個人電腦配備 Pentium 4 CPU 具有 3.6GHz 運算能力及 1.5GB RAM 記憶容量。每個範例皆求解 10 次，並以平均值展示數據求解結果。

4.2.1 WFA4SP 三種分流移步法比較

本測試以提出的三種 WFA4SP 分流移步法測試 att48、berlin52、和 kroA100 三個標竿問題，比較求解結果並分析演算方法的優劣。WFA4SP 演算參數為初始質量 50、每股水流最大分支數 5、匯流相似度 0.8、蒸發率 0.1、總水流數上限

15、降水飽和度 0.9。

為比較不同目標函數值刻劃的問題，本研究使用求得的最佳路徑長度與已知最佳值的誤差率(error percentage)作為比較結果優劣的基準。誤差率計算方式為

$$\varepsilon = \frac{\text{the average shortest length} - \text{the know shortest length}}{\text{the know shortest length}} \times 100(\%) \quad (4-1)$$

表 4-1 展示三種分流模式的求解結果。表中列出各範例的名稱和求解結果，包括求得的最佳路徑長度、十次求解結果的標準差、求得的最佳路徑長度與已知最佳值的誤差率、求得最佳解的目標函式呼叫次數、總目標函式呼叫次數、和求解時間。圖 4-4 是三種分流模式求解結果與全域最佳解的誤差值比較。圖 4-5 展示三種分流模式搜尋時的收斂趨勢。

表4-1. WFA4SP 三種分流模式在相同目標函式呼叫次數下的求解結果

Benchmark (shortest route length)	Methods	Avg. of the best route length	Route length STDEV	Error percentage (%)	Avg. of the number of function calls to find the best route	Avg. of the number of function calls	Computation Time (m:s)
att48 (10628)	CPA	33321.100	820.072	213.522	265385	500008	04:24.1
	SS	23601.800	1339.111	122.072	347767	500007	03:05.2
	LRI	10839.200	102.784	1.987	328514	500006	02:41.4
berlin52 (7544.366)	CPA	19010.895	500.871	151.988	310712	600004	06:17.5
	SS	16896.454	1238.973	123.961	369016	600008	04:26.0
	LRI	7852.725	211.516	4.087	351497	600007	03:45.9
kroA100 (21285.44)	CPA	131242.050	1665.980	516.581	414054	1000003	46:55.1
	SS	105502.627	4120.394	395.656	783394	1000009	38:44.8
	LRI	22681.672	560.204	6.560	820012	1000005	24:05.3

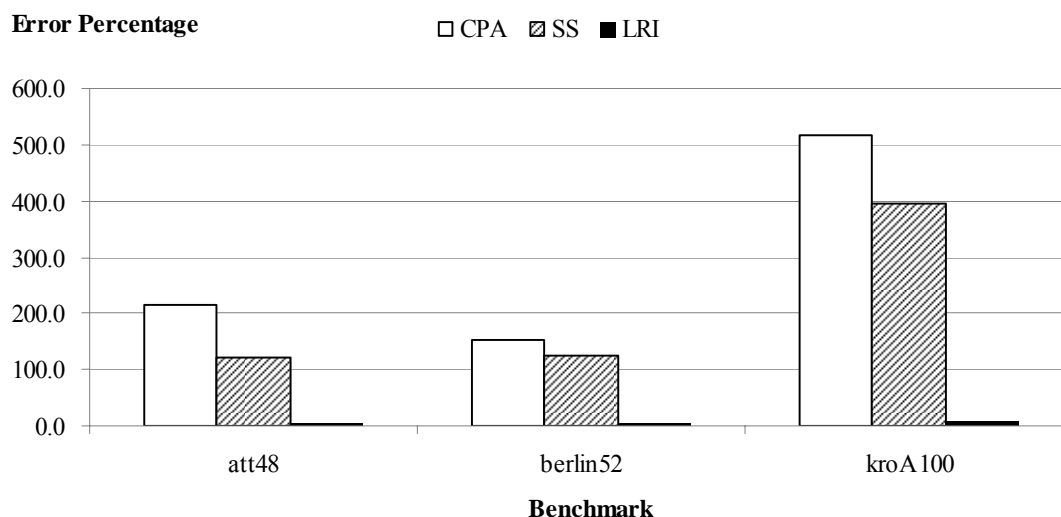


圖4-4. WFA4SP 求解 att48 求解結果比較

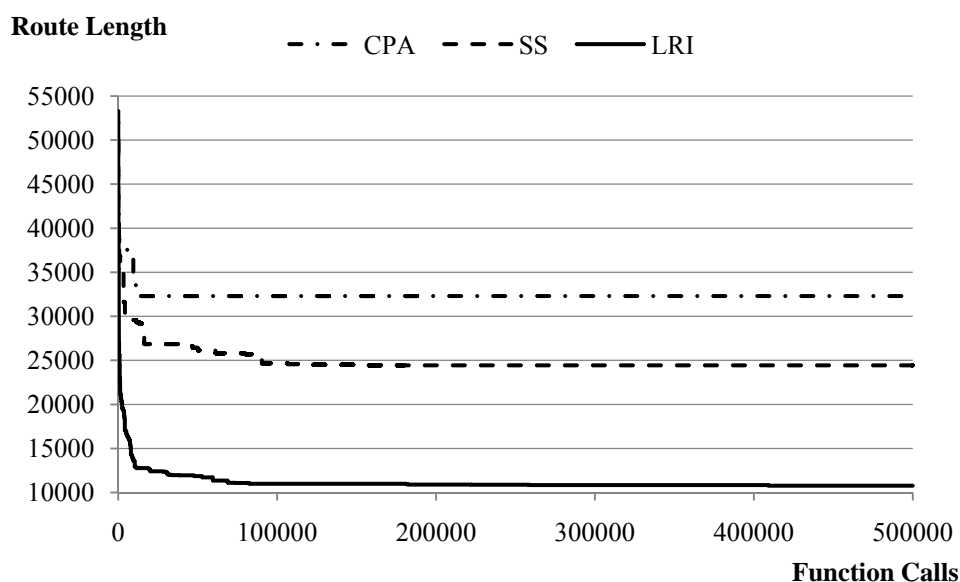


圖4-5. WFA4SP 三種分流模式求解 att48 收斂狀況

由圖 4-4 可看出 LRI 法明顯優於 CPA 和 SS 法。這是由於 LRI 法在分流時會依照物件排序問題的特性，保留演化後較好的鏈結關係，使搜尋成果較好。從圖 4-5 來看，CPA 法收斂到一定程度即無法找到更好的解，顯示搜尋機制搜尋出的解變動過大，而無法往最佳解方向演化。CPA 法雖然較符合移步的特性，但因 TSP 解空間變動率極大且不可預測，因此無法預期所產生出的解會偏向較好的方向。SS 法依照物件排序問題的特性保留一段序列固定，但變動的子序列是屬於不確定性變動，因此僅能提高新產生的解品質會變好的機率，而不能保證調動之後的改變趨向。相較之下 LRI 法能持續往更好的解演化搜尋。

4.2.2 WFA4SP 演算成效分析

經分流方法比較後，本研究以 WFA4SP 選用 LRI 分流移步法與典型 GA 進行求解比較。WFA4SP 演算參數為初始質量 50、每股水流最大分支數 5、匯流相似度 0.8、蒸發率 0.1、總水流數上限 15、降水飽和度 0.9。參數設定經初步測試此種組合較佳，總水流數上限不宜過高，以免匯流作業耗費過多時間。GA 的交配法採用 Greedy Subtour Crossover (GSX)及 Greedy Crossover (GX)兩種針對物件排序優化問題特別設計的演化模式。使用的遺傳演算參數為機體數 50、交配率 0.9，及突變率 0.05。求解時以 WFA4SP 及 GA 演化過程呼叫目標函式次數達設定的上限為停止條件。

其中 GX 法屬啟發式交配法，在進行交配時會參照 TSP 的距離成本矩陣，使交配結果能確實收到改善成效，而 GSX 法則無。一般而言，求解 TSP 時，有參照距離成本矩陣的演算法收斂速度較快。因演化過程中會確實往較小的距離成本搜尋，而使收斂效果較好。

表 4-2 展示 WFA4SP 及 GA 在相同目標函式呼叫次數上限下的求解結果。圖 4-6 是使用 WFA4SP 的 LRI 分流移步法，和 GA 的 GSX 法和 GX 法求解結果比較。圖 4-7 是三者求解 att48 的目標函數收斂狀況的比較。

表4-2. WFA4SP 及 GA 在相同目標函式呼叫次數下的求解結果

Benchmark (shortest route length)	Methods	Avg. of the best route length	Route length STDEV	Error percentage (%)	Avg. of the number of function calls to find the best route	Avg. of the number of function calls	Computation Time (m:s)
oliver30 (424.869)	GA/GSX	451.249	23.994	6.209	228860	300000	00:07.7
	GA/GX	433.730	4.043	2.086	142300	300000	00:11.5
	WFA/LRI	424.869	0.000	0.000	91016	300007	00:41.3
att48 (10628)	GA/GSX	13066.600	2288.697	22.945	424335	500000	00:17.5
	GA/GX	11141.000	150.974	4.827	263000	500000	00:32.9
	WFA/LRI	10801.700	108.175	1.634	404843	500006	02:39.3
eil51 (429.983)	GA/GSX	467.750	12.800	8.783	474170	500000	00:18.8
	GA/GX	441.974	4.178	2.789	300995	500000	00:35.5
	WFA/LRI	437.514	6.216	1.751	319586	500006	03:03.8
berlin52 (7544.366)	GA/GSX	9630.601	1214.505	27.653	460065	600000	00:22.2
	GA/GX	7822.641	179.959	3.689	348355	600000	00:44.1
	WFA/LRI	7939.425	325.311	5.236	354041	600008	03:38.4
rat99 (1211)	GA/GSX	1447.342	45.163	19.516	965760	1000000	01:16.2
	GA/GX	1345.736	38.216	11.126	778625	1000000	03:15.4
	WFA/LRI	1315.104	31.138	8.597	931878	1000007	19:48.6
kroA100 (21285.44)	GA/GSX	26060.961	2284.442	22.436	903285	1000000	01:17.3
	GA/GX	23801.746	604.894	11.822	679060	1000000	03:25.4
	WFA/LRI	22895.703	374.968	7.565	866265	1000007	20:14.7
a280 (2586.77)	GA/GSX	3787.505	127.577	46.418	2976805	3000000	19:05.5
	GA/GX	3850.712	150.151	48.862	2498335	3000000	1:15:05.6
	WFA/LRI	3624.971	126.290	40.135	2983878	3000005	7:10:06.3

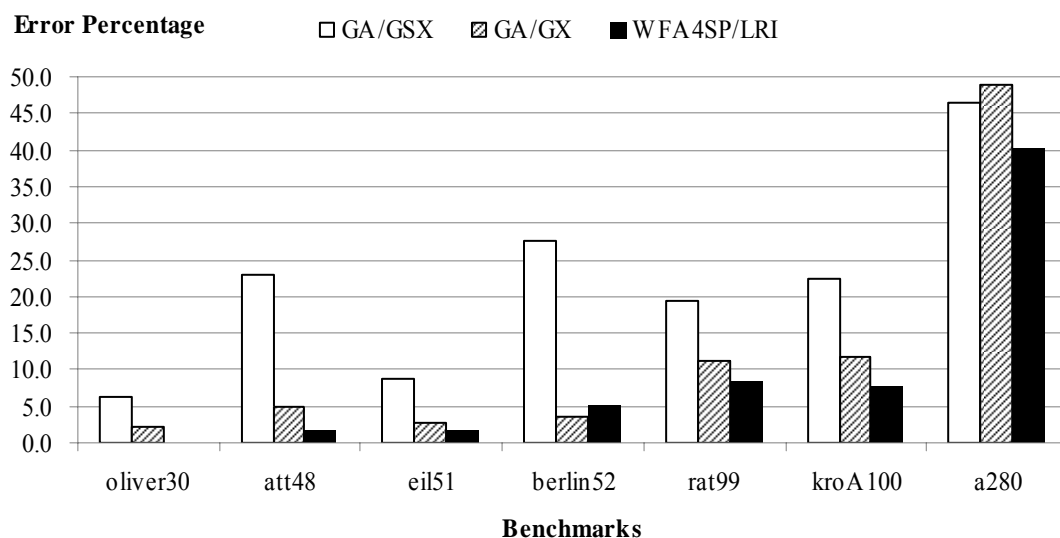


圖4-6. WFA4SP 和 GA 在相同目標函式呼叫次數下的求解結果比較

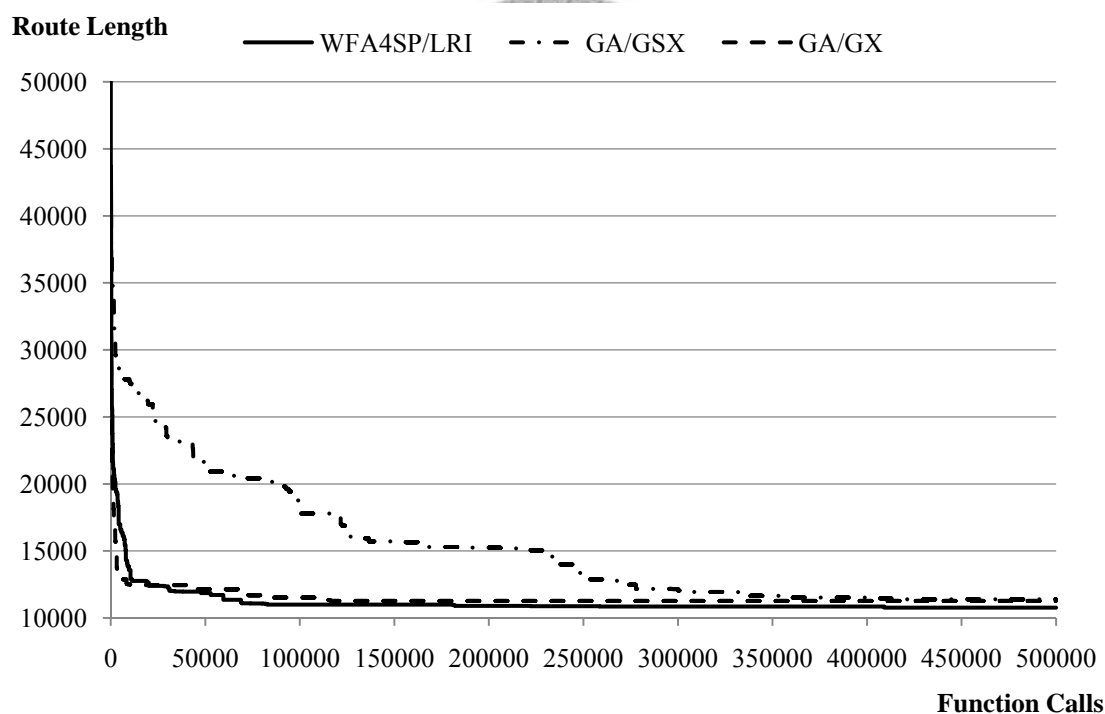


圖4-7. WFA4SP 和 GA 求解 att48 的收斂狀況

由表 4-2 及圖 4-6 可知，在相同目標函式呼叫次數下，WFA4SP 明顯優於 GSX 法。而和 GX 法比較，除 berlin52 外都是 WFA4SP 勝出。但 WFA4SP 的演算步驟較 GA 繁複，求解所需時間較 GSX 法和 GX 法為長。雖然求解所花費的時間與程式實作的效能優劣有很大關連，且在迭代演化的啟發式演算法尤其顯著，為求比較的公允性，再使用相同求解時間為停止條件比較三者的求解效能。結果

如表 4-3 和圖 4-8 所示。圖 4-9 以 att48 為例，展示 WFA4SP 和 GA 求解的收斂狀況。

表4-3. WFA4SP 及 GA 在相同求解時間下的求解結果

Benchmark (shortest route length)	Methods	Avg. of the best route length	Route length STDEV	Error percentage (%)	Avg. of the number of function calls to find the best route	Avg. of the number of function calls	Computation Time (m:s)
oliver30 (424.869)	GA/GSX	429.924	5.086	1.190	1161755	3953895	01:00.0
	GA/GX	428.150	2.506	0.772	1127965	2126850	01:00.0
	WFA/LRI	426.650	5.630	0.419	66444	435100	01:00.0
att48 (10628)	GA/GSX	11113.900	122.693	4.572	5423105	9496345	04:00.0
	GA/GX	10747.100	78.818	1.121	2244505	4285885	04:00.0
	WFA/LRI	10813.400	86.879	1.744	468073	752920	04:00.0
eil51 (429.983)	GA/GSX	450.734	11.340	4.826	10202290	13419705	06:00.0
	GA/GX	431.622	1.800	0.381	3056195	5589720	06:00.0
	WFA/LRI	440.482	4.455	2.442	631768	999931	06:00.0
berlin52 (7544.366)	GA/GSX	8129.031	184.773	7.750	4248645	13170265	06:00.0
	GA/GX	7546.739	2.503	0.031	3227610	5463955	06:00.0
	WFA/LRI	7797.022	239.221	3.349	391575	976430	06:00.0
rat99 (1211)	GA/GSX	1349.895	27.182	11.469	6565885	17646335	20:00.0
	GA/GX	1278.993	21.463	5.615	4159720	6374400	20:00.0
	WFA/LRI	1326.858	31.463	9.567	913920	1004379	20:00.0
kroA100 (21285.44)	GA/GSX	22998.512	788.200	8.048	13020330	26321845	30:00.0
	GA/GX	22719.948	494.920	6.739	2645505	4066830	30:00.0
	WFA/LRI	22260.656	444.075	4.582	1225228	1472473	30:00.0
a280 (2586.77)	GA/GSX	2943.696	52.496	13.798	46001455	64906395	07:00:00.0
	GA/GX	3428.205	100.337	32.528	9285615	20157835	07:00:00.0
	WFA/LRI	3570.921	122.952	38.046	2891865	2930572	07:00:00.0

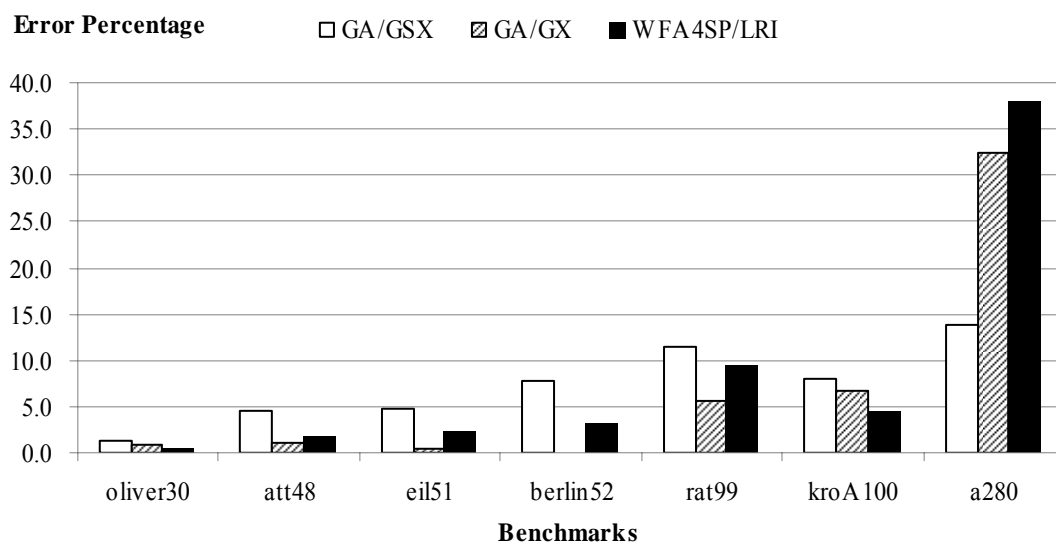


圖4-8. WFA4SP 及 GA 在相同求解時間下的求解結果

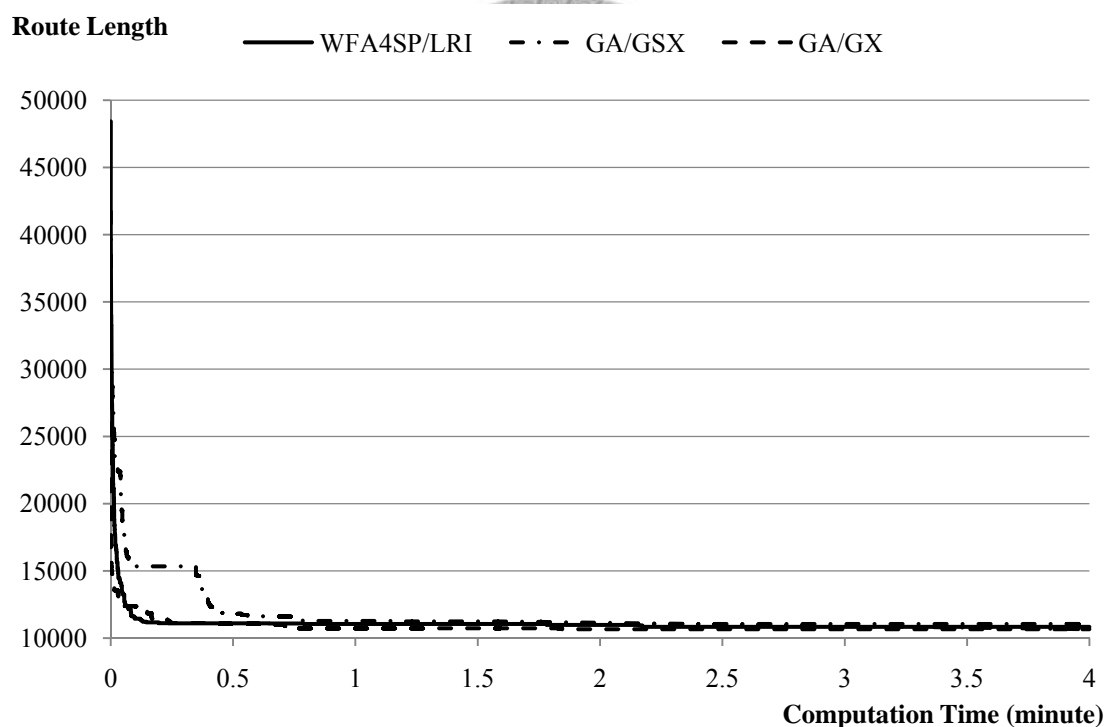


圖4-9. WFA4SP 和 GA 求解 att48 以求解時間為基的收斂狀況

由表 4-3 和圖 4-8 來看，雖然 WFA4SP 在大型問題上的成效不如 GA，但所使用的目標函數求解次數較 GA 少，因此找到較佳解的機率較小。為探究 WFA4SP 耗時的原因，分別計算各作業流程所耗的時間，列於表 4-4。

表4-4. WFA4SP 各作業流程所耗時間

	Computation Time (m:s)					Percentage (%)			
	Total	Splitting	Merge	Evaporation	Precipitation	Splitting	Merge	Evaporation	Precipitation
oliver30	00:41.3	00:01.9	00:39.2	00:00.0	00:00.1	4.57%	94.95%	0.10%	0.17%
att48	02:39.3	00:05.3	02:33.7	00:00.1	00:00.1	3.33%	96.47%	0.04%	0.07%
eil51	03:03.8	00:05.8	02:57.6	00:00.1	00:00.1	3.18%	96.64%	0.03%	0.08%
berlin52	03:38.4	00:06.9	03:31.2	00:00.1	00:00.2	3.14%	96.69%	0.03%	0.07%
rat99	19:48.6	00:22.7	19:25.1	00:00.1	00:00.3	1.91%	98.02%	0.01%	0.03%
kroA100	20:14.7	00:23.2	19:50.8	00:00.1	00:00.3	1.91%	98.03%	0.01%	0.03%
a280	7:10:06.3	03:26.5	7:05:48.2	00:00.2	0:00:00.4	0.80%	98.94%	0.00%	0.00%

由表 4-4 可看出，WFA4SP 的求解時間大部份都耗在匯流作業上。再深究其因，推測是匯流時計算相似度會使求解時間增長。統計計算相似度所耗的時間如表 4-5。

表4-5. 匯流作業耗時分析

	Merge	Similarity Computation	
	Computation Time	Time	Percentage
oliver30	00:39.2	00:38.7	98.56%
att48	02:33.7	02:32.7	99.34%
eil51	02:57.6	02:56.5	99.36%
berlin52	03:31.2	03:29.9	99.36%
rat99	19:25.1	19:21.8	99.72%
kroA100	19:50.8	19:47.7	99.74%
a280	7:05:48.2	7:05:04.8	99.83%

由表 4-5 可看出，匯流作業流程中計算兩序列的相似度會佔用大部份的時間。因計算相似度需分別計算正向循環序列和反轉的循環數列，因此時間複雜度會隨維度呈二次成長，亦即複雜度為 $O(q^2)$ 。所以在求解大維度問題時，WFA4SP 的求解時間會大幅增加，使得在同樣求解時間內呼叫目標函數求解次數較少，導致嘗試的解搜尋個數太少，因此較不易求得較佳解。

由於計算序列相似度的目的是區別兩解序列是否需合併，因此只需判斷兩序列的相似度是否大於設定的合併相似度底限 β 。換言之，若兩序列的不相似度大於 $1-\beta$ 則不需合併。所以兩序列 E 和 S 進行循環比對時，需比對 S 和 S 的反轉序

列 R 個別的循環路徑集合 $\hat{\mathbf{S}}$ 和 $\hat{\mathbf{R}}$ ，而其中 \hat{S}_k 或 \hat{R}_k 與 E 比對時，只要不相同的座標數大於 $q(1-\beta)$ 即可判定兩序列不相似，則不需繼續比對此序列，再接著比對下一條循環序列 \hat{S}_{k+1} 或反轉序列 \hat{R}_{k+1} 。此外，在 $\hat{\mathbf{S}}$ 和 $\hat{\mathbf{R}}$ 中找到和 E 的相似度大於 β 的序列時，即可判定兩序列需合併，而不需再繼續比對其他序列。依此方法判別兩序列是否鄰近，可大幅減少匯流作業所需時間。**錯誤！找不到參照來源。**展示以改進後的相似度計算法和原本的求解結果比較。表 4-7 列出改進後的相似度計算法相較原本計算方式所節省的時間。

表4-6. 改進後的相似度計算法和原計算法的求解結果

Benchmark (shortest route length)	Merge Methods	Avg. of the best route length	Route length STDEV	Error percentage (%)	Avg. of the number of function calls to find the best route	Avg. of the number of function calls	Computation Time (m:s)
oliver30 (424.869)	Original	424.869	0.000	0.000	91016	300007	00:41.3
	Improved	424.869	0.000	0.000	99235	300004	00:19.7
att48 (10628)	Original	10801.700	108.175	1.634	404843	500006	02:39.3
	Improved	10889.500	145.276	2.460	298741	500004	01:10.8
eil51 (429.983)	Original	437.514	6.216	1.751	319586	500006	03:03.8
	Improved	443.929	5.251	3.243	343709	500007	01:19.3
berlin52 (7544.366)	Original	7939.425	325.311	5.236	354041	600008	03:38.4
	Improved	7925.156	174.369	5.047	303501	600008	01:37.7
rat99 (1211)	Original	1315.104	31.138	8.597	931878	1000007	19:48.6
	Improved	1328.678	27.638	9.717	873797	1000007	07:53.8
kroA100 (21285.44)	Original	22895.703	374.968	7.565	866265	1000007	20:14.7
	Improved	22566.228	469.902	6.017	909217	1000005	08:01.2
a280 (2586.77)	Original	3624.971	126.290	40.135	2983878	3000005	7:10:06.3
	Improved	3397.736	153.247	31.351	2975954	3000009	2:36:23.2

表4-7. 改進後的相似比對法和原計算法所耗時間比較

Benchmark	Similarity Computation		Improvement Percentage (%) of Improved over Original
	Original	Improved	
oliver30	00:39.2	00:16.4	58.304%
att48	02:33.7	01:00.0	60.983%
eil51	02:57.6	01:07.7	61.876%
berlin52	03:31.2	01:23.7	60.363%
rat99	19:25.1	07:07.6	63.297%
kroA100	19:50.8	07:13.4	63.607%
a280	7:05:48.2	2:26:36.8	65.568%

由表 4-6 和表 4-7 可看出，改進後的相似度計算法不影響原本的求解品質，並且可有降低求解時間。但求解時間仍較 GA 為長。

4.3 小結

本章展示本研究所開發的 WFA4SP 排序優化問題求解系統，並以此系統測試 TSPLIB 的標竿問題，驗證本研究所提出的三種分流移步模式。分析結果顯示，LRI 模式求解 TSP 的效能最好。再以 LRI 模式，與 GA 以 GSX 法和 GX 法兩種交配法共同測試七個標竿問題，以目標函數呼叫次數與共同求解時間為兩種停止條件分別求解。在共同目標式呼叫次數下，WFA4SP 的 LRI 模式得到的結果皆較 GA 的兩種交配法優秀。但因 WFA4SP 的搜尋方法較為繁複，因此在共同求解時間下演化的次數較少。原因在於 WFA4SP 的匯流作業流程中，須反覆比對序列的相似度，因此所耗時間會隨問題大小以平方增加。經改進後，求解時間可大幅減少。

第5章 結論與未來研究建議

仿水流優化演算法是基於仿照水流形式，動態調整代理人概念而設計的啟發式演算法。本研究承襲仿水流演算法的架構，研擬求解一般性物件排序優化問題的水流分流法。同時開發一套求解系統驗證所提出的分流模式，並與遺傳演算法比較求解成效。本章提出結論和未來研究建議。

5.1 結論

WFA 是近年新創的優化演算法，以水流的自然特性為設計概念，試圖尋找水流空間中的最低點位置以及問題解空間中的最佳解。本研究沿襲 WFA 的求解架構，建構符合物件排序優化問題特性的演算模式，提出求解排序優化問題的仿水流優化演算法，WFA4SP。同時並開發 WFA4SPDOS 的仿水流演算系統，驗證所提方法。本研究使用國際 TSPLIB 內的標竿問題為測試對象，並與迄今較佳的遺傳演算法求解結果比較分析，並進行效能評估，結果有：

1. 為符合水流物理特性，本研究改進原有的 WFA 分流數設定，當水流的目標函數值改善率越大時水流越集中於少數幾股，改善率小時會分散出較多水流進行搜尋。如此可使搜尋到的解改善率大時，能集中往較可能尋到最佳解的區域搜尋。而解的改善率小時，則可多方搜尋鄰近區域以覓得鄰近較佳解，找出較好的改善移步再前進。
2. 本研究提出三種分流移步法供 WFA4SP 求解物件排序優化問題，分別為 CPA、SS、和 LRI 法。驗證結果顯示，以針對 TSP 問題特性規劃的 LRI 分流移步法效果最佳。CPA 法雖較符合水流移動特性，但在離散空間求解時解的品質變動過大，無法找到適當的移步進行搜尋，導致求解成效不佳。SS 法保留解序列中部份序列的方式進行演化，但解的變動略為保守而無法有效率地收斂。LRI 法因繼承水流或其母水流中較好的座標值鏈結關係，所以較易演化出較好的解。
3. WFA4SP 以 LRI 法與 GA 的 GSX 法和 GX 法兩種交配法作驗證比較。在相同目標函式呼叫次數下，WFA4SP 的成效比 GA 的兩種交配法為佳。這是由

於 LRI 法在演化機制上會保留較好的鏈結關係，使得搜尋結果較好。但因 WFA4SP 的演算程序遠較典型 GA 繁複，當問題規模大時求解時間會大幅增加。因此在相同求解時間下，當問題規模較小時，LRI 法的成效與 GA 的 GX 法並不遜色；當問題規模大時，演化次數及解搜尋的個數遠較 GA 少，因此求解成效較差。

4. WFA4SP 以 WFA 為基礎架構，提出符合排序問題限制的演算作業流程來求解一般性物件排序優化問題。經實驗驗證可改善求解 TSP 的誤差率，顯示 WFA 可用於求解一般性排序優化問題。

5.2 未來研究建議

WFA 自 2006 年問世迄今，已順利提出物件分群優化問題及物件排序優化問題的求解模式且也驗證其效能及可行性，但仍有後續研究改善的可能。本研究提出幾點建議，供後續研究參考：

1. 本研究提出三種分流移步法，相較於原有的 WFA 已不再使用禁制清單的型式來避免循環搜尋。但不保證演化時就不會重覆搜尋到相同的解。未來可研擬完整的回流防制方法，掌握水流的搜尋方向，以開拓出具全域最佳解潛能的搜尋區域。
2. WFA4SP 提出的 LRI 分流移步法雖然可有效搜尋求解，但 WFA4SP 整體的匯流作業計算兩序列的相似度耗時太大，雖然經過改良，但求解時間仍較 GA 長。未來可嘗試以 LRI 法的概念為基礎，設計較為精簡的相似度比對演算步驟，則更能有效求解。
3. WFA4SP 係針對物件排序優化問題的特性而設計的演算機制。未來可開發其他特定優化問題的求解模式，如 JSP，以拓展 WFA4SP 的應用。
4. 許多求解 TSP 的文獻皆使用區域搜尋來輔助尋找最佳解。可嘗試在 WFA4SP 中加入區域搜尋機制，更能有效搜尋到較佳解。

參考文獻

- Birbil, S. I. and S.-C. Fang (2003). "An electromagnetism-like mechanism for global optimization." Journal of Global Optimization **25**(3): 263-282.
- Dorigo, M., V. Maniezzo, et al. (1996). "Ant system: optimization by a colony of cooperating agents." Systems, Man, and Cybernetics, Part B, IEEE Transactions on **26**(1): 29-41.
- Flood, M. M. (1956). "The traveling salesman problem." Operations research **4**(1): 61.
- Glover, F. and F. Laguna (1997). Tabu Search, Kluwer Academic Publishers.
- Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Longman Publishing Co., Inc.
- Holland, J. H. (1992). Adaptation in natural and artificial systems, MIT Press.
- Karp, R. M. (1972). "Reducibility among Combinatorial Problems." Complexity of Computer Computations (R. E. Miller and J. W. Thatcher, eds.): 85-103.
- Kennedy, J. and R. Eberhart (1995). Particle swarm optimization. Neural Networks, 1995. Proceedings., IEEE International Conference on.
- Reinelt, G. (1991). "TSPLIB. A traveling salesman problem library." ORSA Journal on Computing **3**(4): 376-384.
- Sengoku, H. and I. Yoshihara (1998). A Fast TSP Solver Using GA on JAVA. Proc. 3rd int. Symp. Artif. Life and Robot.
- Yang, F.-C. and Y.-P. Wang (2007). "Water Flow-Like Algorithm for Object Grouping Problems." Journal of the Chinese Institute of Industrial Engineers **24**(6): 475-488.

Yang, R. (1997). Solving large travelling salesman problems with small populations.
Genetic Algorithms in Engineering Systems: Innovations and Applications, 1997.
GALESIA 97. Second International Conference On (Conf. Publ. No. 446).

王元鵬 (2006). 仿水流離散優化演算法. 工業工程學研究所, 國立臺灣大學. 碩士論文.

