

第三章 研究方法

3.1 研究步驟

圖 3-1 為本研究之步驟，首先針對訂單分配問題進行定義，以釐清研究之範圍與限制，接著利用數學模型將問題模式化，以精確的表達訂單分配問題。由於此問題屬於 NP，因此本研究採用基因演算法求解此一數學模型，並且加強求解過程，以達更佳之結果，而參數設定部分將建立實驗設計，最後針對結果進行分析與比較。

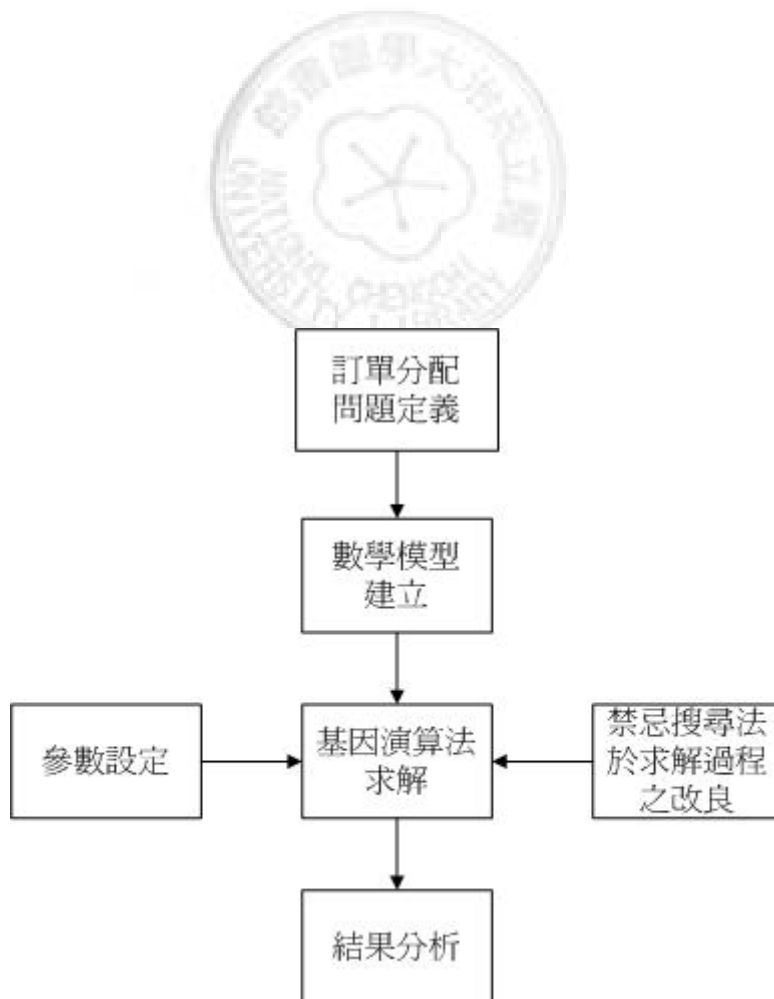


圖 3-1 研究步驟

3.2 訂單分配問題定義

多廠區訂單分配問題，圖 3-2 為其邏輯示意，企業由中央總公司接到訂單後，分配到世界各地之工廠，工廠生產完畢後再送到客戶手中，不過物流中心至客戶端的運送不在本研究中。

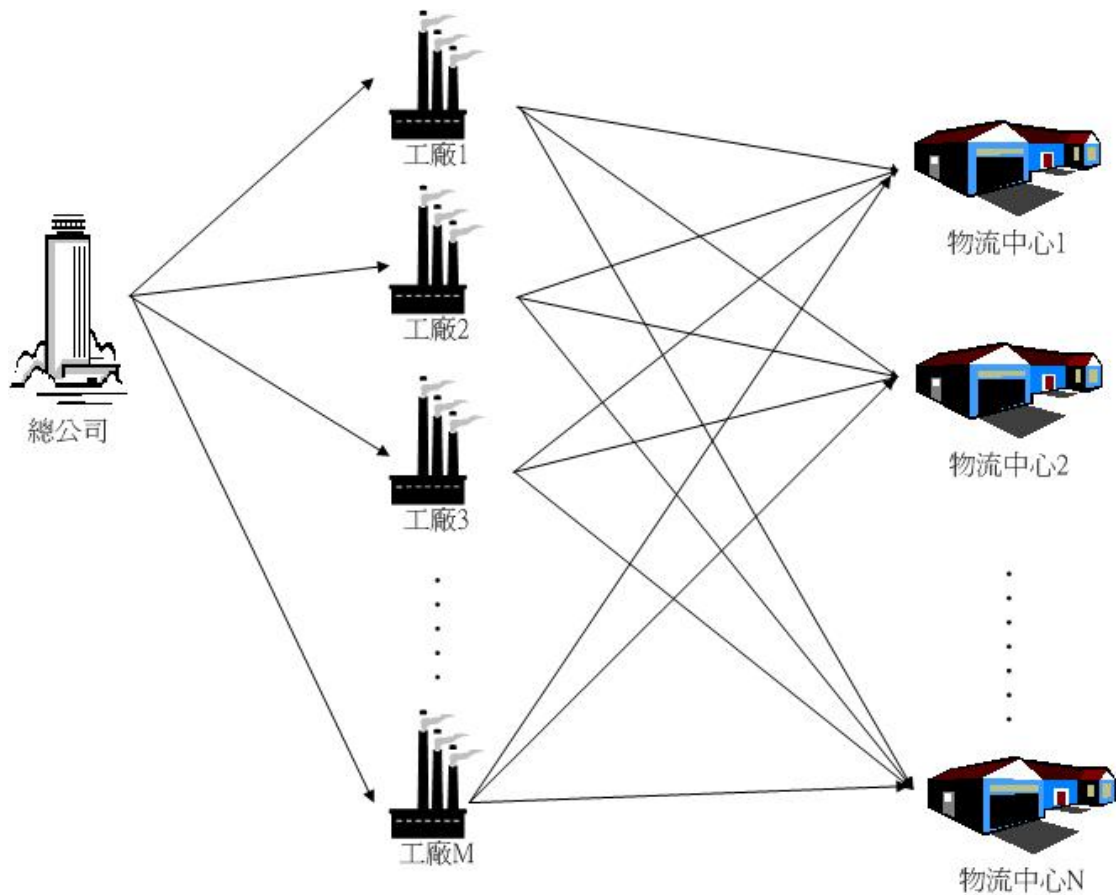


圖 3-2 問題架構邏輯圖

3.2.1 研究限制

本小節介紹本研究中之各項限制與假設：

- 採用非真實之數據，因為本研究針對求解效率與多目標規劃進行探討
- 不考慮訂單分派後，工廠內機台生產加工排程
- 訂單分配過程，不允許緊急插單

- 每張訂單包含該有的明確資訊（如：交期、訂購數量、產品種類等）
- 工廠之物料採購不在本研究中
- 生產方式非批量式（batch）生產
- 生產過程機器無故障情形發生
- 工廠生產之整備時間不在本研究中
- 層級分析法中的各個準則間權重設定上，無一致性的問題。

3.2.2 決策目標之選擇

Chan, F. et al[3]針對層級分析法結合基因演算法求解訂單分配問題中，多目標規劃之考量，提出下列三種目標：總成本、配送前置時間、產能利用率之均衡性。總成本最小，考量產品生產與運輸成本，此為製造業最基本之考量；配送前置時間，此段時間指的是從接到訂單至產品生產完成送至物流中心之時間，此時間的長短可表示企業服務水準，且避免訂單集中在部分工廠生產，而降低訂單完成時間；產能利用率之均衡性，我們以產能利用率之標準差來衡量，此項考量是避免訂單集中分配在部分工廠生產，而忽略了其他工廠之產能。

3.2.3 訂單分配之求解架構

本研究求解訂單分配問題，需考量三個目標：製造成本、配送前置時間、產能利用率之標準差，而這三個目標將利用層級分析法結合，並以基因演算法求解，演化過程中，當產生違反限制式的解時，會以禁忌搜尋法進行修補（原因將在 3.4.1 節中說明），最後產生接近最佳之訂單分配結果。整體架構我們可以參考圖 3-3：

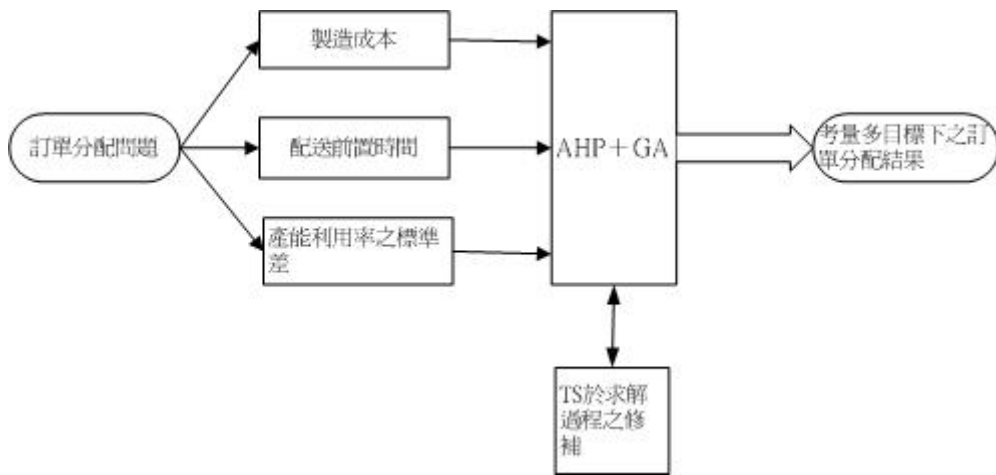


圖 3-3 訂單分配求解架構

3.3 研究模型

本節針對訂單分配問題模型進行說明，以逐步描繪研究之整體架構與概念：

3.3.1 模型限制

- 每張訂單可包含有多種種產品
- 每間工廠可製造多種產品
- 訂單可被拆解
- 產品之單位製造成本與生產數量無關
- 每間工廠之產能上限為一固定之常數

3.3.2 模型之輸出入

本小節針對訂單分配問題建立之模型

(一) 模型之輸入

表 3-1 模型輸入表

輸入項目	項目內容
訂單資訊	<ul style="list-style-type: none"> • 訂購數量 • 訂購之產品種類 • 訂單交期 • 運輸成本

	<ul style="list-style-type: none"> • 延遲成本
工廠資訊	<ul style="list-style-type: none"> • 各生產產品之成本 • 工廠產能上限
系統環境資訊	<ul style="list-style-type: none"> • 規劃期天數 • 工廠數量 • 各目標相互間之權重

(二) 模型之輸出

表 3-2 模型輸出表

輸出項目	項目內容
生產計畫	<ul style="list-style-type: none"> • 各工廠於規劃週期內各日所需之生產數量
目標資訊	<ul style="list-style-type: none"> • 考量權重下整體之適合度

3.3.3 模型建立

本小節首先介紹訂單分配問題之模型建立，接著介紹基因演算法結合層級分析法之架構，以求解問題模型，其中並說明禁忌搜尋法如何改善求解過程，這些都將在皆下來的小項中逐一介紹。

(一) 模型變數定義

表 3-3 模型變數定義表

變數	定義
i	訂單編號 i
k	工廠編號 k
t	目前日期 t
j	產品種類
d_i	訂單 i 之交期
n	工廠數量

p	規劃期天數
C _k	產能上限
q _i	訂單 i 之產品訂購量
c _i	訂單 i 之產品種類
X _{i,k,t}	訂單 i 於工廠 k 於日期 t 之生產量
U _{k,t}	工廠 k 在日期 t 之產能利用率
DL _i	訂單 i 之配送前置時間
UC _k	工廠 k 在規劃週期內之平均產能利用率
PCost _k	工廠 k 之單位生產成本
TCost _{k,i}	訂單 i 在工廠 k 所需之單位運輸成本
DCost _i	訂單 i 之懲罰成本 (單位：元/個)

(二) 訂單分配模型模型建立

多廠區訂單分配模型如下：

製造成本=生產成本+運輸成本+整備成本+延遲成本

$$= \sum_i \sum_k \sum_t X_{i,k,t} \times (PCost_k + TCost_{k,w_i}) + \sum_i \sum_k \sum_t y_{j,k,t} \times SUC_{k,j} + \sum_i \sum_k \sum_{t>d_i} X_{i,k,t} \times [t - d_i] \times DCost_i$$

$y_{j,k,t}=1$ where $\exists i$ such that $c_i=j$ and $X_{i,k,t}$

=0 otherwise

(3-1)

$$\text{配送前置時間} = \sum_i DL_i, DL_i = \max(t) \text{ where } X_{i,t} \neq 0 \quad (3-2)$$

$$\text{產能利用率之標準差} = \sqrt{\frac{[UC_k - \text{avg}(UC_k)]^2}{n}} \quad (3-3)$$

$$U_{k,t} = \frac{\sum_i X_{i,k,t}}{C_k} \quad (3-4)$$

工廠 k 在日期 t 之產能利用率

$$UC_k = \frac{\sum_t U_{k,t}}{P} \quad (3-5)$$

工廠 k 在規劃週期內之平均產能利用率（每日）

$$avg.(UC_K) = \frac{\sum_k UC_k}{n} \quad (3-6)$$

各廠在規劃週期內之平均產能利用率

限制式：

$$\sum_i \sum_t X_{i,k,t} \leq C_k \quad (3-7)$$

$$\sum_k \sum_t X_{i,k,t} = q_i \quad (3-8)$$

$$X_{i,k,t} \geq 0 \quad \text{and} \quad X_{i,k,t} \in N, 0 \quad (3-9)$$

(三) 模型說明：

表 3-4 模型說明表

編號	說明
3-1	<p>此數學式列出生產之總成本，包含生產成本、運輸成本與懲罰成本。</p> <ul style="list-style-type: none"> • 生產成本的計算方式是以（訂單 i 於工廠 k 在日期 t 之生產量）×（工廠 k 之單位生產成本）。 • 運輸成本的計算方式是以（訂單 i 於工廠 k 在日期 t 之生產量）×（工廠 k 至訂單 i 欲送至之物流中心 w_i 之單位成本）。 • 懲罰成本之計算方式是以（訂單 i 於工廠 k 在日期 t 之生產量）×× 單位懲罰成本

3-2	此數學式計算所有訂單之配送前置時間總和，配送前置時間簡單的說就是接受訂單至生產完成送至物流中心之時間，因此每張訂單於工廠中「最後」生產之時間，即為所求。
3-3	工廠於規劃週期之平均產能利用率標準差計算，此式子僅為統計上標準差公式之套用。
3-4	工廠 k 在日期 t 之產能利用率之計算，是以此時間此工廠所有生產量除以該工廠產能上限計算。
3-5	工廠 k 在規劃週期內之平均產能利用率，是以此段期間內各日產能利用率之平均求得。
3-6	各工廠於規劃期間內「平均產能利用率」之平均數。
3-7	於工廠 k 之總生產量需小於該工廠之產能上限。
3-8	訂單 i 於各工廠之生產量總和需等於該訂單之訂購量。
3-9	各工廠之生產量需為 0 或正整數。

3.4 基因演算法於訂單分配問題之應用

本節針對求解模型之演算法架構進行說明，包括「基因演算法結合層級分析法之架構」，此為演算法之主體，以及「層級分析法於基因演算法中之實行方式」與「禁忌搜尋法於演化過程中之修正」，兩者是針對層級分析法與禁忌搜尋法在演算法中的應用作細部的說明。

3.4.1 基因演算法結合層級分析法之架構

在第二章文獻探討中，已經對基因演算法的架構說明過，在此我們將藉由 Chan, F. et al 提出之基因演算法結合層級分析法架構求解訂單分配問題，其中並加入禁忌搜尋法之修正，以期在速度與品質上求得更佳的分配結果。整體架構圖

可參考圖 3-xxx，在此我們將逐項說明：

(一) 染色體編碼與參數設定：

在林慈傑(2002)的研究中，提出「廠段」的概念，將訂單分配結果陣列，由三維轉化為二維，本研究引用此概念進行矩陣型基因編碼，將工廠編號與時間合併，再依序重新統一編號：

訂單分配至各工廠生產量之矩陣：

	工廠 1	工廠 2
訂單 1		
訂單 2		

訂單於各日期生產量之矩陣：

	日期 1	日期 2	日期 3
訂單 1			
訂單 2			

合併上列兩者後之訂單與廠段矩陣：

	工廠 1			工廠 2		
	日期 1	日期 2	日期 3	日期 1	日期 2	日期 3
	廠段 1	廠段 2	廠段 3	廠段 4	廠段 5	廠段 6
訂單 1						
訂單 2						

利用上列矩陣，進行矩陣型染色體編碼。

(二) 產生起始解

本研究利用亂數產生一組起始解，除了效率高外，還可讓整個解空間都有均等的機會，若產生之起始解為不可行解，則再以禁忌搜尋法修正。

(三) 啟動禁忌搜尋法找尋鄰近可行解

在演化的過程中，若產生之起始解、交配或突變後之染色體為不可行解時，則啟動禁忌搜尋法尋找鄰近可行解(feasible solution)來替代不可行解(infeasible solution)，

(四) 利用層級分析法計算適合度

本研究根據 Chan 所提之架構進行修改與延伸，整體架構可見圖 3-4。在

基因演算法產生的群體中，每個染色體都會先計算其三個目標式之數值，每個染色體可代表一個可行解，將其套入模型中的三個目標式中，可得到其值，也就是該組可行解的總成本、配送前置時間與平均產能利用率之標準差。接著利用層級分析法，共分下列幾個步驟：

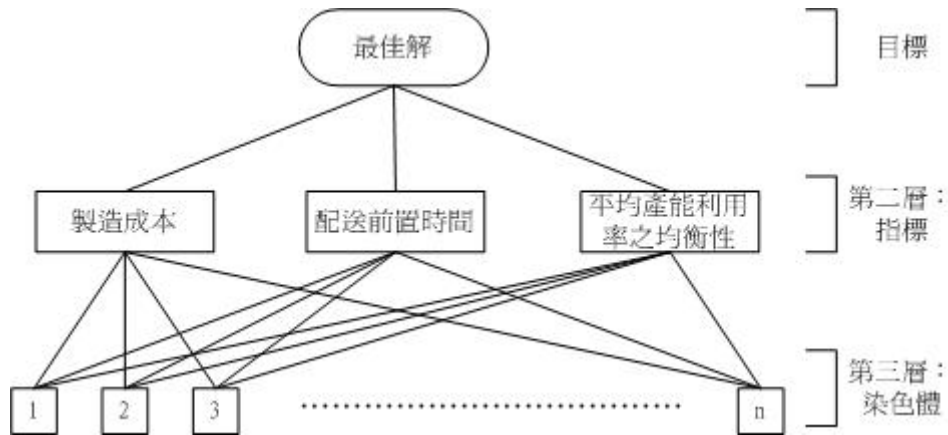


圖 3-4 層級分析法於基因演算法中應用示意圖

步驟 1: 計算相對目標值

例如：染色體*i*之相對總成本 =
$$\frac{\text{染色體}i\text{之總成本}}{\text{該群體所有染色體中最高總成本}}$$

$$\text{染色體}i\text{之相對配送前置時間} = \frac{\text{染色體}i\text{之配送前置時間}}{\text{該群體所有染色體中最高配送前置時間}}$$

$$\text{染色體}i\text{之相對平均產能利用率標準差} = \frac{\text{染色體}i\text{之平均產能利用率標準差}}{\text{該群體所有染色體中最高平均產能利用率標準差}}$$

步驟 2: 計算染色體間相對權重

例如：染色體 A 與染色體 B，以總成本為例，則

$$\text{染色體 A 之權重 (相對於 B)} = (R_b - R_a) + 1$$

$$\text{染色體 A 之權重 (相對於 B 且 B 之值大於 A 時)} = 1 / [(R_b - R_a) * 10 + 1]$$

說明：常數 10 的作用是為了使相減後的值大於 1

常數 1 則是使 A 和 B 相等時，權重為 1；且使權重必大於 1

步驟 3: 計算指標間之權重

例如： 建立指標之成偶比對矩陣

A,B,C : 各個指標

$$\begin{array}{c} A \quad B \quad C \\ A \left[\begin{array}{ccc} 1 & 2 & 3 \\ B \left[\begin{array}{ccc} 1/2 & 1 & 5 \\ C \left[\begin{array}{ccc} 1/3 & 1/5 & 1 \end{array} \right] \end{array} \right. \end{array} \right]$$

接著利用特徵值(eigenvalue)計算，得知各指標間之比重

步驟 4: 計算總適合度並挑選最佳解

利用染色體之權重與各目標權重做運算後加總，最高適合度者為最佳解。式子如下：

$$\sum_j \sum_i C_{i,j} \times O_j = \text{染色體 } i \text{ 之總適合度}$$

$C_{i,j}$: 染色體 i 在指標 j 下的適合度

O_j : 指標 j 之權重

(五) 染色體複製

多數基因演算法的研究中，染色體皆依照適合度之大小來決定是否存活至下一代，本研究也同樣採用輪盤法 (Roulette Wheel Selection) 篩選染色體，適合度高的染色體有較高的機率被選中保留至下一代，沒被選中的則被淘汰，符合基因演算法的演化策略。

$$r_i = \frac{f_i}{\sum_i f_i}$$

n : 群體數目

r_i : 染色體 i 被選中之機率

f_i : 染色體 i 之適合度(fitness)

(六) 染色體的交配

在研究的重點非在於探討染色體的交配運算子，因此我們沿用第二章中介紹的矩陣型交配運算子，來搭配矩陣型染色體編碼，而交配率則設定在0.5~1之間。

(七) 染色體的突變

染色體的突變採用矩陣型突變運算子，而突變率則照一般設定為0.07。

(八) 停止條件

由於訂單分配問題相當複雜，無法判斷其所需要的時間約略為何，但也不可能無止盡的進行運算，因次我們將以子代產生的次數為停止條件，使其在某一子代停止。

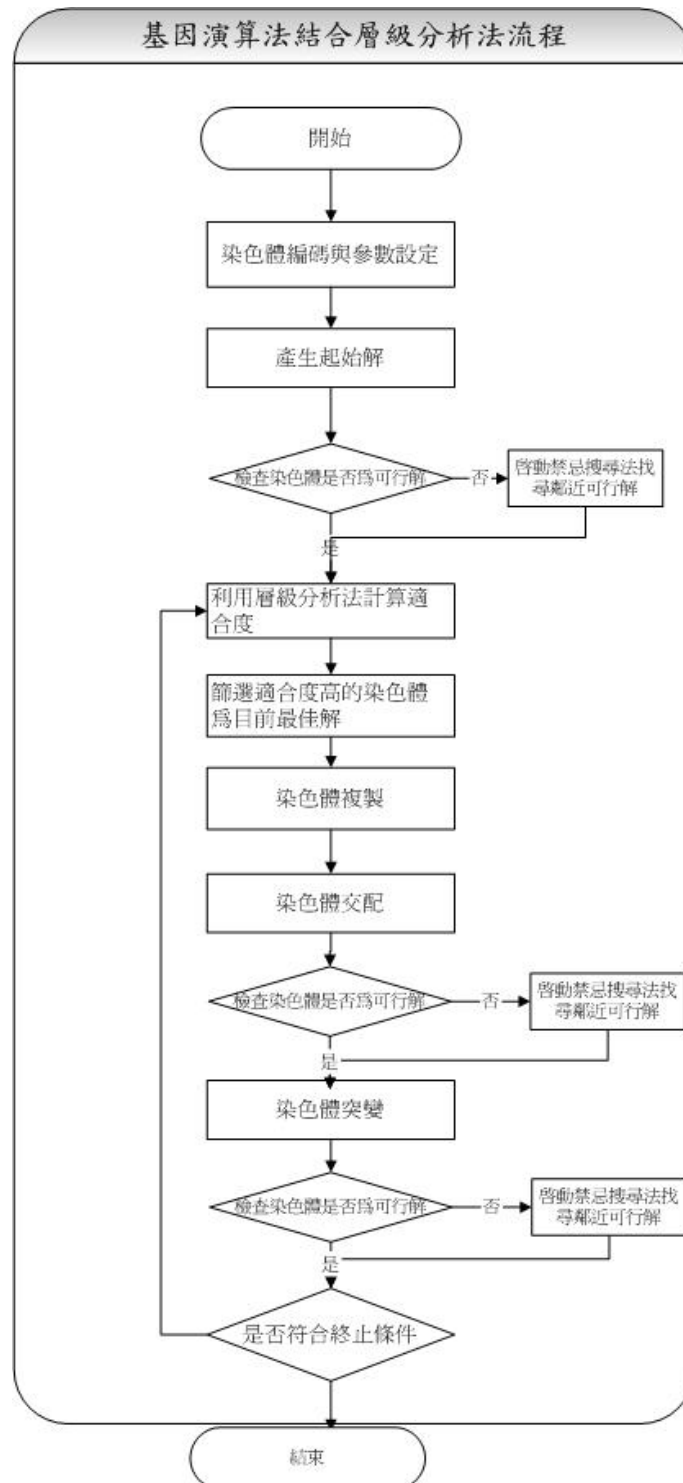


圖 3-5 基因演算法結合層級分析法流程圖

3.4.2 禁忌搜尋法於演化過程中之修正

禁忌搜尋法的主要架構已於第二章有了充分介紹，本小節將探討如何利用它

來尋可一鄰近可行解。禁忌搜尋法架構包括起始解、移步、禁忌名單與停止條件，此外的長期、短期記憶和破禁原則，由於此處之應用所期望的是速度與可行解，並非解的品質，所以這幾項不列入架構中。

(一) 起始解

由於本研究是利用禁忌搜尋法來尋找一可行解，進而去替代原先之不可行解，且找到之可行解可能是在原先不可行解鄰近之最佳解，以起改善求解速度與品質，因此我們的起始解採用由基因演算法輸入之不可行解染色體。

(二) 移步

Glover, F. 對於移步提出了幾種方式，交換(Swap)、插入(Insert)、加入/消去(Add/drop)和增加/減少(Increase/decrease)等等方式，本研究採用林師檀(民91)所提出之根據增加/減少之概念所設計之移步方式，其作法如下：

$$X1' = X1 \pm M$$

$$X2' = X2 \pm M$$

X1 : 原先解中之某一變數

X1' : X1 之鄰近解中的變數

X2 : 原先解中之某一變數

X2' : X2 之鄰近解中的變數

M : 移步值

利用對解中之變數的增加或減少，來產生新的變數，以求得新的鄰近解。

(三) 停止條件

由於使用禁忌搜尋法之目的是尋找一可行解，因此我們將停止條件設為當鄰近解中出現一可行解時，即停止。

3.5 結果分析設計

本研究最後將針對(1)基因演算法結合禁忌搜尋法與基因演算法(林慈傑)

兩者之結果比較 (2) 不同參數設定下基因演算法結合禁忌搜尋法之結果比較，由(1)可確定是否採用基因演算法與禁忌搜尋法結合，對於求解過程有所幫助，而(2)則可以瞭解這個方法是否可以利用參數設定更佳提升結果的品質。