

國立成功大學
機械工程學系
碩 士 論 文

以 β -Pareto預測法解決不確定因素下之多
目標最佳化

A Method to Predict β -Pareto Frontiers for Multiobjective
Optimization under Uncertainty

研 究 生：林東信
指 導 教 授：詹魁元博士

中 華 民 國 九 十 八 年 七 月

以 β -Pareto預測法解決不確定因素下之多目標最佳化

A Method to Predict β -Pareto Frontiers for Multiobjective
Optimization under Uncertainty

研 究 生：林東信

Student: Dong-Shin Lin

指 導 教 授：詹魁元博士

Advisor: Dr. Kuei-Yuan Chan

國立成功大學
機 械 工 程 學 系
碩 士 論 文

A Thesis

Submitted to Department of Mechanical Engineering
National Cheng Kung University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Dept. of Mech. Eng.

July 2009

Tainan, Taiwan

中華民國九十八年七月

以 β -Pareto預測法解決不確定因素下之多目標最佳化

學生：林東信

指導教授：詹魁元博士

國立成功大學機械工程學系

摘 要

多目標最佳化在許多工程領域已廣泛的被應用，最佳解所形成的Pareto集合可提供設計工程師或政策決定者了解各目標函數之間的妥協關係。近年來，可靠度最佳化利用隨機變數模擬不確定因素，並結合最佳化的概念，已成功的探討單目標設計問題面臨自然界不確定因素下所需因應的策略，然而對於Pareto集合在不確定因素下的表現卻尚無清楚的了解。

在此論文中，我們利用文獻上使用目標空間計算線性多目標問題之Pareto集合的技術，結合線性Pareto集合在考慮不確定因素下的平移特性，首先發展一套可有效的預測不同可靠度下Pareto集合(亦稱為 β -Pareto集合)的方法。此方法首先利用簡算法得到任一單目標的最佳解為非凌駕角解(non-dominated extreme point)，再利用拘束條件active的判定，形成一縮減成本係數矩陣(reduce cost coefficient matrix)，用該矩陣計算出此非凌駕角解之架構向量(frame)，然後便可用最佳化的計算依序得知完整的Pareto集合。在得到一組Pareto集合後，使用一階二次可靠度方法計算非凌駕角解在不確定因素下不同 β 量值的移動，直到active狀態改變，此改變可經由KKT必要條件的Lagrange Multiplier得知。

除了線性問題外，本論文並延伸討論非線性多目標問題的 β -Pareto集合。首先針對非線性問題的Pareto集合用區塊三明治夾擠法線性化，在針對線性化後之系統進行 β -Pareto集合的演算，並利用增加線性化區間的方法控制誤差。本論文提出之方法經由數學範例驗證可有效率的預測線性或非線性系統的 β -Pareto集合，使決策者能在使用最少資源的情況下得知各目標函數在不同可靠度要求下的妥協關係。

A Method to Predict β -Pareto Frontiers for Multiobjective Optimization under Uncertainty

Student: Dong-Shin Lin

Advisor: Dr. Kuei-Yuan Chan

Department of Mechanical Engineering
National Cheng Kung University

ABSTRACT

In this research we investigate design optimization problems under uncertainties with multiple objectives. The probabilistic formulation of constraints when uncertainties are considered in an optimization framework has received extensive studies in the literature. Most research use a single objective function to explore the optimum under uncertainty. In this study, we consider the same probabilistic constraint framework with multiple objective functions. The results of the multiobjective optimization under uncertainty is a set of Pareto frontiers at different reliability levels, β -Pareto. We investigate how Pareto frontiers move with different reliability values and then use a LP Pareto frontier generation method to predict β -Pareto frontiers without running multiple MOO problems. We also extend the solution method to nonlinear problems. From this work we demonstrate how the proposed method can be a useful tool in decision-making under uncertainty when quick estimates of outcomes of a design decision are needed in the early stage of product design process. Compared to existing methods the proposed work show significant efficiency for general MOO under uncertainty.

誌謝

這本論文能夠完成，首先謝謝指導教授，詹魁元老師。感謝能在忙碌的生活中不時給予我適當的建議與方向，讓我在遇到瓶頸時能夠用不同的角度來看待問題，研究才得已突破；更感謝老師能給我這個機會，來學習如何解決一個不熟析的問題，讓我在這個過程學習到應有的態度與處理技巧，還有增進表達能力；我想我學到的東西不只可以應用在研究上，也能幫助我在人生的旅途上航行順利。

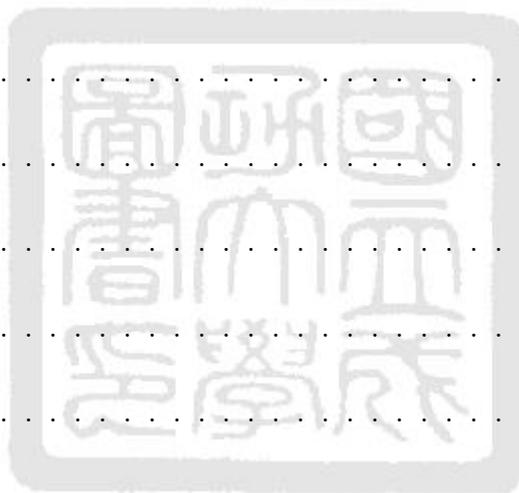
感謝我的口試委員，陳家豪老師以及劉宜欣老師的指教，讓這本論文可以更加完善；特別感謝劉老師先前的研究對本論文的貢獻，讓我猶如站在巨人的肩膀上，能看的更高更遠。

感謝系統最佳化實驗室的夥伴們，凱勛，彥智，子頡，昱達，琇雯，勝昌，季儒，佳豪與祐伸，還有已經畢業的智豪學長與岱景學長，能夠與大家一起煎熬，奮鬥是很開心的事情，很幸運能跟大家處在同一個實驗室；還有在成大認識的好友們，芳富，于庭，峻嘉，惠晴，芷誼，羽薰，冠辰，建宏，還有現在還在德國的世同，你們為我帶來一段歡樂的研究生生活，謝謝大家。

最後要感謝我的家人，給我不愁吃穿的環境，讓我能專心於研究，也謝謝佳旻不時的陪伴與鼓勵，有了你們的支持，我才能有今天，謝謝！

目 錄

書名頁	i
論文口試委員審定書	ii
中文摘要	iii
英文摘要	iv
誌謝	v
目錄	vi
表目錄	ix
圖目錄	x
符號說明	xii
第一章、背景與文獻探討	1
1.1 多目標最佳化	3
1.1.1 權重法	5
1.1.2 拘束法	6
1.1.3 混合式方法	7
1.2 線性多目標最佳化及Pareto集合計算方法	8
1.2.1 線性多目標最佳化	9
1.2.2 可達空間之邊界-架構(Frame)	11
1.2.3 在目標空間計算線性Pareto集合演算法	13
1.2.4 範例	15
1.3 不確定因素對多目標最佳化的影響	18



1.3.1	可靠度分析	20
1.3.2	可靠度最佳化設計	24
1.4	研究動機與研究方法	26
第二章	不確定因素下線性問題的Pareto集合平移	28
2.1	不確定因素對線性拘束條件的影響	28
2.2	拘束條件active情況在考慮不確定因素下的改變	30
2.3	非凌駕角解的移動	33
2.4	線性Pareto集合邊界的移動	34
2.5	改變active拘束條件的可靠度尋找	35
2.6	考慮不確定因素下之線性多目標最佳化的Pareto集合演算法	36
2.7	範例	39
第三章	不確定因素下非線性問題的Pareto集合的改變	41
3.1	Pareto集合的線性化	41
3.1.1	區塊三明治夾擠法	41
3.1.2	非線性拘束條件的線性化	43
3.2	不確定因素下之非線性多目標最佳化的Pareto集合演算法	45
3.3	範例	48
第四章	結果與展望	51
4.1	結論	51
4.2	研究貢獻	51
4.3	未來工作與展望	52
參考文獻		53

附錄一：專有名詞中英文對照表	56
自傳	57



表 目 錄

1.1 線性Pareto集合運算過程	17
2.1 非凌駕角解的相關資訊	40



圖 目 錄

1.1	雙目標函數最佳化之目標空間	3
1.2	拘束法計算雙目標最佳化之Pareto集合示意圖	6
1.3	\mathbf{R} 矩陣的行向量與錐體	11
1.4	線性多目標最佳化Pareto集合演算法	13
1.5	範例(1.24)的設計空間	15
1.6	範例(1.24)的目標空間	15
1.7	Pareto集合之Dauer's method與Constraint method比較	17
1.8	標準高斯分佈機率密度	19
1.9	文獻[31]中T型梁上翼與下翼的比較	25
1.10	文獻[31]中T型梁長寬比例的比較	25
1.11	文獻[32]中在不同公差配置下的成本與敏感度關係圖	26
2.1	與圖1.8相同的標準高斯分佈機率密度	29
2.2	範例(2.6)設計空間	30
2.3	範例(2.6)目標空間	30
2.4	文獻[34]中描述KKT條件的幾何意義	32
2.5	考慮不確定因素下之線性多目標最佳化Pareto集合演算法	36
2.6	考慮不確定因素下的線性Pareto集合組	40
3.1	區間三明治夾擠法示意圖	42
3.2	由上界函數 $u_i(f)$ 尋找下界函數 $l_i(f)$ 示意圖	43

3.3	下界函數 $l_i(f)$ 的適用範圍	44
3.4	非線性Pareto集合演算流程	45
3.5	非線性問題的設計空間	48
3.6	非線性問題的 β -Pareto集合	49
3.7	線性化不考慮可靠度的Pareto集合	50
3.8	對特徵點上active的拘束條件線性化示意圖	50



符 號 說 明

a_g	線性不等式拘束條件中，設計變數 x 的係數
a_g^B	沒有 active 的線性不等式拘束條件中，設計變數 x 的係數
a_g^D	active 的線性不等式拘束條件中，設計變數 x 的係數
a_h	線性等式拘束條件中，設計變數 x 的係數
b_g	線性不等式拘束條件中的常數項
b_h	線性等式拘束條件中的常數項
c^T	線性目標函數中，設計變數的係數
E^i	\mathcal{F} 中 x 與 \bar{x} 連線形成的邊界
\mathcal{F}	可行解空間
F	架構
$g(x)$	不等式拘束條件
$h(x)$	等式拘束條件
k	線性與非線性的目標函數個數
$l(f)$	逼近 Pareto 集合曲線函數的下界函數
m	線性與非線性的不等式拘束條件個數
n	設計變數 x 的維度
Pr	機率
P_f	破壞機率
$p(f)$	Pareto 集合曲線函數
R	簡算法中的縮減成本係數矩陣

Re	可靠度
\mathbb{R}	實數空間
r^i	R 矩陣中的行向量
s_g	差額變數
\mathcal{T}	可達空間
U	標準空間中的設計變數
$u(f)$	逼近Pareto集合曲線函數的上界函數
w	線性與非線性的等式拘束條件的個數
X	隨機設計變數
x^*	最佳解
x^p	效率解
\bar{x}	\mathbb{R}^n 中鄰近於 x 的任一角解
y^*	最佳解的目標函數值
y^p	非凌駕解
\bar{y}	\mathbb{R}^k 中鄰近於 y 的任一角解
β	可靠度指標
ϵ_j	拘束法中放進不等試拘束條件的目標函數之目標值
λ	KKT必要條件中的Lagrange multiplier
σ_X	隨機設計變數的標準差
θ_i	\mathcal{T} 中 y 與 \bar{y} 的距離
μ_X	隨機設計變數的平均值
$\Phi(x)$	在標準高斯分佈底下距離原點 x 的機率



第一章 背景與文獻探討

在工程上，解決一個問題的解答常常不是唯一，在這多組解答中，總有一個會是我們最想要的最佳解答。例如在機器的設計上，我們希望找出使機器性能最好的設計；在結構的設計上，我們會希望能找出能承受力量越大越好的設計。而最佳化設計演算法便是有系統的找尋最佳解的程序。

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{h}(\mathbf{x}) = \mathbf{0} \\ & \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \end{aligned} \tag{1.1}$$

方程式(1.1)是單目標最佳化的一般數學形式， \mathbf{x} 稱之為設計變數， \mathbf{x} 存在的空間稱為 n 維的設計空間： $\mathbf{x} \in \mathbb{R}^n$ ， \mathbf{g} 與 \mathbf{h} 為設計所考量的拘束條件，其中 $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ 稱為等式拘束條件， $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$ 稱為不等式拘束條件，這兩種拘束條件限制了設計變數 \mathbf{x} 在設計空間的活動範圍，只要符合這兩個拘束條件的 \mathbf{x} 都是決策者能接受的解答，這些解答形成的空間稱為可行解空間 \mathcal{F} ，其定義如方程式(1.2)。

$$\mathcal{F} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{h}(\mathbf{x}) = \mathbf{0}, \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\} \tag{1.2}$$

方程式(1.1)希望可以找出一個適當的設計變數讓某個我們希望的目標函數 $f(\mathbf{x})$ 最佳，而最佳可能是使 $f(\mathbf{x})$ 越大越好，或者是越小越好，在這裡我們為了統一問題的定義，便以越小越好當作最佳的判定，換句話說如果想要把越大越好的目標函數套用在此數學模式裡，只要將目標函數加個負號即可。假設 \mathbf{x}^* 是方程式(1.1)的最佳解，要找到 \mathbf{x}^* 則必須透過演算法來進行一連串的疊代過程來求解，已經有許多作者對單目標最佳化問題做相當程度的討論 [1]。

為了文章後面描述方便，這裡定義一個與最佳解 \mathbf{x}^* 有密切關係的觀念：在最佳化問題中，假如把某個拘束條件忽略不管，就會使最佳化的結果 \mathbf{x}^* 改變的話，就稱這個拘束條件active [2]，因此等式拘束條件一定active，但是除了不等式的拘束條件全部都是線性的情況之外，active的不等式拘束條件在 \mathbf{x}^* 的值卻不一定都等於零 [2]。

如果方程式(1.1)中的目標函數 $f(\mathbf{x})$ 與拘束條件 $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ ， $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$ 皆為線性，則稱做線性單目標最佳化問題，又稱為線性規畫(Linear Programming, LP)，可以將方程式(1.1)改寫

成方程式(1.3)的形式。

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}_i^T \mathbf{x} & (1.3) \\ \text{s.t.} \quad & \mathbf{a}_h^T \mathbf{x} - \mathbf{b}_h = 0 \\ & \mathbf{a}_g^T \mathbf{x} - \mathbf{b}_g \leq 0 \end{aligned}$$

方程式(1.3)是以矩陣的方式描述線性單目標最佳化問題，其中 \mathbf{c}_i^T 是一個 $(n \times 1)$ 的矩陣，而這個問題的設計變數 \mathbf{x} 被限制在 w 條線性的等式拘束條件與 m 條線性的不等式拘束條件： \mathbf{a}_h^T 為一個 $(w \times n)$ 的矩陣， \mathbf{a}_g^T 為一個 $(m \times n)$ 的矩陣， $\mathbf{b}_h \in \mathbb{R}^w$ ， $\mathbf{b}_g \in \mathbb{R}^m$ 。

在這個線性的單目標最佳化問題上，因為相較於非線性的問題形式簡單，有不少文章與書都在討論解決方程式(1.3)的演算法 [3,4]，其中簡算法(Simplex Method) [4]是解決線性單目標最佳化問題最具代表性的演算法；因為線性的可行解空間 \mathcal{F} 為多面體，因此若定義角解(Extreme Point)為方程式(1.3)中 \mathcal{F} 上的角點(Vertex)，則其演算法的概念便是從可行解空間裡的一個角解移動到鄰近的角解，直到找到最佳解為止。

在文獻 [4]定義角點就是不能用可行解空間 \mathcal{F} 中的點以非零的線性組合而成的點，即為 \mathcal{F} 這多面體的尖端；以三維的設計空間為例，一般情況是三個形成 \mathcal{F} 邊界的平面相交在一點，這點就叫角點。

1.1 多目標最佳化

在單目標最佳化的問題裡，我們雖然可以找到使目標最好的解，但在這些我們想要改善的主要目標之外，往往需要考慮另外一個或是多個目標函數，例如在機器的設計上，除了使性能最佳之外，成本往往也是我們想要越少越好的目標，因此我們需要對多個目標的最佳化問題做討論。

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{f} = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})\} \\ \text{s.t.} \quad & \mathbf{h}(\mathbf{x}) = 0, \quad \mathbf{g}(\mathbf{x}) \leq 0 \end{aligned} \quad (1.4)$$

方程式(1.4)是一般多目標最佳化的數學形式，與方程式(1.1)最大的不同點在於方程式(1.4)同時考慮了多個目標函數 \mathbf{f} ，而此多目標函數存在的空間稱為目標空間， $\mathbf{f} \in \mathbb{R}^k$ ；因為同一個設計變數 \mathbf{x} 會有多個目標函數值，所以我們難以直接判定此設計變數是否比其他設計變數好。

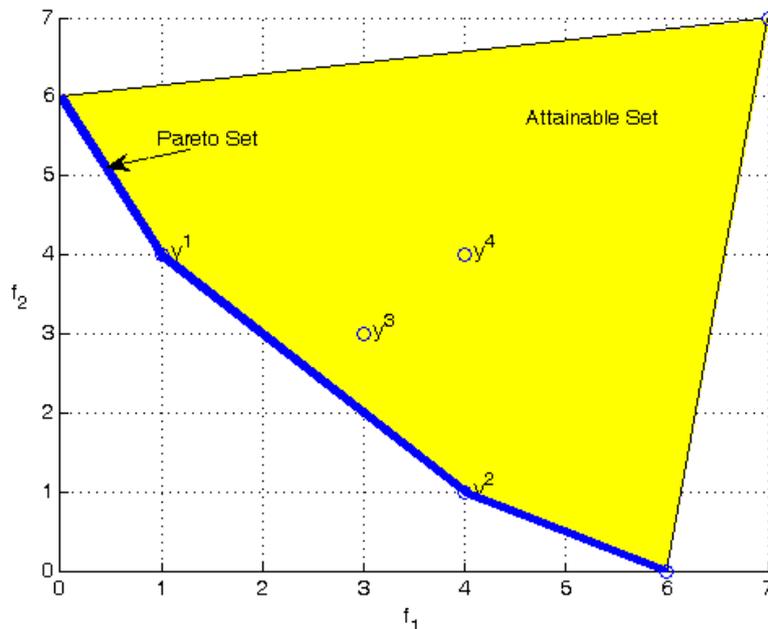


圖 1.1: 雙目標函數最佳化之目標空間

圖1.1為一個雙目標最佳化問題在目標空間的示意圖，圖中的區塊表示可行解空間 \mathcal{F} 能夠在目標空間中達到的目標函數值，稱之為可達空間(Attainable Set)， \mathcal{T} ；我們可說 $y^3 = (3, 3)$ 比 $y^4 = (4, 4)$ 來的小，因此 y^4 被 y^3 凌駕(Dominated)。

接著我們觀察到可達空間中的點不存在任何點可凌駕 $y^1 = (1, 4)$ 與 $y^2 = (4, 1)$ ，我們稱這兩個點為非凌駕解(Nondominated)，我們對非凌駕解 y^p 有個一般性的定義 [5]如下： \mathcal{F} 為可行解空間，其中 $\mathbf{x}^p \in \mathcal{F}$ ，若 \mathbf{x}^p 為效率解，而 $y^p = (f(\mathbf{x}^p))$ 為非凌駕解，則必不存在 $\mathbf{x} \in \mathcal{F}$ ，使得 $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^p)$ ，其中 $i = 1, \dots, k$ ，且至少一個 j 使得， $f_j(\mathbf{x}) < f_j(\mathbf{x}^p)$ ，其中 $i \neq j$ 。

上一段對非凌駕解的定義說明了在設計空間中，找不到其他設計變數 \mathbf{x} 的每一個目標函數值都比非凌駕解 \mathbf{x}^p 的每一個目標函數值好，但是卻一定可以找到某個設計變數 \mathbf{x} 的某個目標函數值優於非凌駕解的同一個函數值，例如圖1.1我們沒辦法直接的比較 y^1 與 y^2 兩者哪個比較小，像這種想要改善其中一個目標函數值卻會讓另一個或是多個目標函數值變差的情形，稱之為權衡得失(Trade-off)。在無法直接判斷哪個解答最好的情形下，最適當的方法是將所有的非凌駕解全列出來，再讓決策者依照特殊考量來做決定。如圖1.1較為粗的線段上便是此圖所有的非凌駕解，所有非凌駕解的集合就稱之Pareto集合，是義大利經濟學家Pareto在二十世紀初提出的概念 [6]。

多目標最佳化的計算結果，可能是分析者(電腦)演算之後直接提供給決策者一個解，或者是產生出一個Pareto集合之後再讓決策者依照自己的偏好選出一個解。決策者還是希望可以得到一個解就好，而不是一個集合，前人對多目標最佳化的討論提供了許多求解的方法 [5]，這些求解過程依照決策者參與的程度不同而可以分成四類：

- **完全不考慮決策者偏好的方法**

在決策者對目標函數的了解不明確，沒有辦法偏好於任何一個目標函數時，此類的選擇是很適合用來求解的方法。此類的方法把每一個目標函數皆視為相同重要，最具代表性的方法為妥協法(Compromise programming) [7]，此方法把每一個目標函數與最佳參考點(通常為原點)的距離加總，加總距離越短的設計越好，轉換成一個單目標問題求解。

- **先產生Pareto集合的方法**

此類的方法是決策者希望分析者把Pareto集合先產生出來，再讓決策者觀察Pareto集合權衡得失的情形與決策者依自己的經驗與偏好來做決定，此類方法具代表性的是權重法(Weighting Method)，拘束法(Constraint Method)，與混合式方法(Hybrid Method)，將在下一段詳細介紹。

- **考慮決策者偏好的方法**

假如決策者對於目標函數的偏好已經十分清楚，此類方法可以讓決策者在分析者開始

求解過程之前就把偏好的部分考慮進去，具代表性的方法是評估函數法(Value Function Method)與目的規劃(Goal Programming)：評估函數法是決策者有一個評估函數(Utility function)可以將不同的函數的偏好量化，並加總，數學形式 $U: \mathbb{R}^k \rightarrow \mathbb{R}$ ，而求解過程便是讓此評估函數值越大越好；而目標規畫則是依序的考慮目標函數，決策者希望先達到某個目標函數的最佳解之後再一一考慮其他的目標函數。

● 與決策者互動的方法

此類方法是四類方法中發展最蓬勃的一類，此類方法在分析者求解的每一次疊代過程，都會提供決策者一些資訊讓決策者參考，並詢問決策者一些偏好上的問題好進行下一個疊代直到找出決策者滿意的解。這類方法可以參考 [5]。

決策者對目標函數的偏好取決於決策者在其領域的經驗與知識，在還沒有足夠的能力決定每個目標函數的偏好時，最好的方法還是產生出Pareto集合來讓決策者做決定，因此本論文將以計算出Pareto集合為目標，下面對產生Pareto集合的三個代表性的方法做介紹：

1.1.1 權重法

權重法是由Gass和Saaty在1955年 [8]與Zadeh在1963年提出 [9]，其主要的特色在於將個別的目标函數，經過權重的分配，組合成單目標函數，進而得到一個單目標最佳解，再透過改變權重比例分配的方式，得到另外一組單目標最佳解；不斷有系統地嘗試各種不同的權重比，即可得到一組由多個不同的單目標最佳解所組成的效率解集合。主要的特色就是將多目標最佳化的數學模型轉換成爲單目標最佳化數學模型如1.5：

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_{i=1}^k \omega_i f_i(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{h}(\mathbf{x}) = 0, \quad \mathbf{g}(\mathbf{x}) \leq 0 \\ \text{where} \quad & \omega_i \in [0, 1]; \sum_{i=1}^k \omega_i = 1 \end{aligned} \tag{1.5}$$

其中 ω_i 爲權重比例，其總合爲1，經由上述的轉換，即可將一個原本爲多目標最佳化問題，成功地轉換成爲單目標最佳化問題。但亦由於目標函數爲的線性組合的關係，當設計空間爲凹集合空間時，則權重法將無法找到凹口部份的效率解，因此權重法在使用上將有其限制 [10]。

1.1.2 拘束法

由Marglin在1967年所提出 [11]，其主要將多目標函數中的某些目標函數，轉換成限制條件來進行最佳化，進而將原本的多目標最佳化問題，成功的轉換成單目標最佳化問題，如方程式1.6所示。

$$\begin{aligned} \min_{\mathbf{x}} \quad & f_l(\mathbf{x}) & (1.6) \\ \text{s.t.} \quad & \mathbf{h}(\mathbf{x}) = 0, \quad \mathbf{g}(\mathbf{x}) \leq 0 \\ & f_j(\mathbf{x}) \leq \varepsilon_j, \quad \forall j = 1, \dots, k, \quad j \neq l \end{aligned}$$

我們可以從(1.6)發現，原本的多目標函數，藉由將其它的目標函數轉變成為拘束條件後，已經成功轉換為單目標最佳化問題。圖1.2是以拘束法計算雙目標最佳化的Pareto集合示意圖，演算步驟如下：

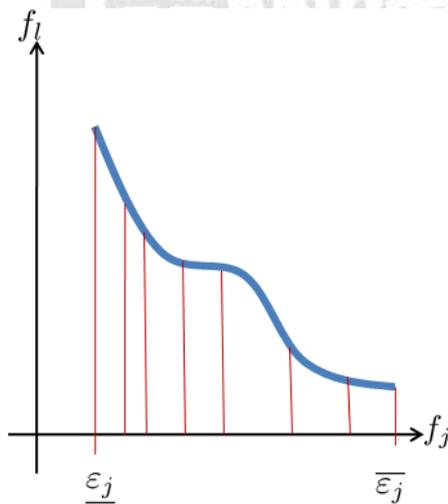


圖 1.2: 拘束法計算雙目標最佳化之Pareto集合示意圖

- 步驟一計算第 j 個目標函數的單目標最佳值做為方程式(1.6)中 ε_j 的下界 $\underline{\varepsilon}_j$ 。

$$\begin{aligned} \min_{\mathbf{x}} \quad & f_j(\mathbf{x}) & (1.7) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{x}) \leq 0 \end{aligned}$$

$$\underline{\varepsilon}_j = f_j(\mathbf{x}^*), \quad \text{where } \mathbf{x}^* = \operatorname{argmin}\{f_j(\mathbf{x}), \quad \text{s.t. } \mathbf{g}(\mathbf{x}) \leq 0\}$$

- **步驟二** 計算另一個目標函數的最佳解 $f_l(\mathbf{x}_l^*)$ 後，將此最佳解代入目標函數 $f_j(\mathbf{x}_l^*)$ 做為 ε_j 的上界 $\bar{\varepsilon}_j$ 後均勻的在 $[\varepsilon_j, \bar{\varepsilon}_j]$ 取點後計算相對應方程式(1.8)的最佳結果 $f_l(\mathbf{x}^*)$ ，Pareto集合便是包含所有 $(\varepsilon_j, f_j(\mathbf{x}_l^*))^T$ 的集合。

$$\bar{\varepsilon}_j = f_j(\mathbf{x}_l^*)$$

$$\begin{aligned} \min_{\mathbf{x}} \quad & f_l(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{h}(\mathbf{x}) = 0, \quad \mathbf{g}(\mathbf{x}) \leq 0 \\ & f_j(\mathbf{x}) \leq \varepsilon_j, \quad \forall j = 1, 2, \quad j \neq l \end{aligned} \tag{1.8}$$

1.1.3 混合式方法

此方法可以視為權重法與拘束法的一般形式，由Corley在1980年 [12]與Wendell和Lee在1977年 [13]提出，其數學形式如方程式(1.9)

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_{i=1}^k \omega_i f_i(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{h}(\mathbf{x}) = 0, \quad \mathbf{g}(\mathbf{x}) \leq 0 \\ & f_j(\mathbf{x}) \leq \varepsilon_j, \quad \forall j = 1, \dots, k \end{aligned} \tag{1.9}$$

where $\omega_i > 0 \quad \forall i = 1, 2, \dots, k.$

而從計算方程式(1.9)的觀點來看，混合式方法與拘束法比較相近(逐漸增加 ε_j 來計算方程式(1.9))。

這三種方法中，因為權重法中的權重較難系統的改變而得到Pareto集合，相較之下權重法撰寫簡單而且在描述Pareto集合的分佈較為平均 [14]，因此本論文將採用拘束法來做為檢驗工具。

1.2 線性多目標最佳化及Pareto集合計算方法

線性多目標最佳化的數學形式如方程式(1.10)

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{a}_g \mathbf{x} - \mathbf{b}_g \leq 0 \end{aligned} \quad (1.10)$$

在方程式(1.10)中， \mathbf{x} 所形成的是設計空間： $\mathbf{x} \in \mathbb{R}^n$ ， \mathbf{a}_g 為一個 $m \times n$ 的矩陣， \mathbf{b}_g 是一個 $m \times 1$ 的矩陣，說明此問題有 m 條不等式拘束條件；矩陣 \mathbf{c} 為一個 $k \times n$ 的矩陣，因此 $\mathbf{c}^T \mathbf{x} \in \mathbb{R}^k$ 說明這是個 k 維度的目標空間。

因為數學上一條方程式可以決定一個變數的值，等式拘束條件理論上可以用帶入函數的方式來減少設計變數的維度，但是實際處理上大部分的等式拘束條件很難以此方法消去，本論文為了簡化問題與專注在Pareto集合的計算方法，因此將等式拘束條件排除在本論文的討論範圍外。

線性多目標最佳化的非凌駕解定義與先前定義相同，可用矩陣形式書寫為：假如有一個線性多目標最佳化問題如方程式(1.10)的效率解 $\mathbf{x}^p \in \mathcal{F}$ ，則在 \mathcal{F} 中找不到其他的 \mathbf{x} 使得 $\mathbf{c}^T \mathbf{x} \leq \mathbf{c}^T \mathbf{x}^p$ ，則稱 $\mathbf{c}^T \mathbf{x}^p = \mathbf{y}^p$ 為方程式(1.10)的非凌駕解。

在此章前部分討論計算Pareto集合的方法不僅可以計算非線性的多目標最佳化問題，也可以處理線性的多目標最佳化問題，但是無論是拘束法還是權重法，都必須要計算一定數量的單目標最佳化問題之後，才能產生與單目標最佳化問題相同數量的非凌駕解，也就是說如果想要更精確的了解Pareto集合權衡得失的情形，就必須計算更多數量的單目標最佳化問題。

線性多目標最佳化就因為可行解空間 \mathcal{F} 是個凸集合的多面體，因此線性多目標最佳化的可達空間 \mathcal{T} 也是個凸集合的多面體 [15]，而且線性多目標最佳化可達空間 \mathcal{T} 上的角點也會剛好對應到 \mathcal{F} 上的角點，如果定義在 \mathcal{T} 中非凌駕解的角點為非凌駕角解，而其對應在 \mathcal{F} 的效率解也會在角點上，稱為效率角解，因此只要分析 \mathcal{F} 上的所有角點，做其是否為非凌駕解的檢查即可判斷出所有的效率角解，Pareto集合即可依這些非凌駕角解建構出來。

過去已有文獻透過分析在設計空間中所有的角點或是邊界來計算出整個Pareto集合 [16,17]，但是在處理實際問題上經常是設計變數遠多於目標函數的個數，拘束條件的個數也是一個很大的數目，這時去分析計算設計空間中所有的角點時會使得計算量急遽

增加，Dauer與Krueger在 [18]中討論一個水資源分配的問題，其中有52個設計變數，30個拘束條件，3個主要的目標函數，這個問題在設計空間的角解可能超過1000個，但目標空間的非凌駕解卻只有21個 [19]；Dauer與Liu因此提出在目標空間處理線性多目標最佳化問題的方法 [19]，可以較有效率的計算Pareto集合，因此在這篇論文中將使用在目標空間計算Pareto集合的方法，方法介紹如下。

1.2.1 線性多目標最佳化

Dauer與Liu在 [19]提出在目標空間計算Pareto集合的方法，是以單目標最佳化演算法中的簡算法當作基礎，發展出一套類簡算法(Simplex-like method)。首先把方程式(1.10)中的不等式拘束條件如同簡算法一般的加上差額變數(Slack variable)使不等式拘束條件轉換成等式拘束條件，如方程式(1.11)：

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{s}_g} \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{a}_g^T \mathbf{x} - \mathbf{b}_g + \mathbf{s}_g = 0 \\ & \mathbf{s}_g \geq 0, \quad \mathbf{s}_g \in \mathbb{R}^m \end{aligned} \quad (1.11)$$

方程式(1.11)中 \mathbf{x} 是 n 維的設計變數，而 \mathbf{s}_g 是個 m 維的差額變數，為了計算方便，先將這兩種變數視為相同性質處理，定義一個 $(n+m)$ 維的新變數 \mathbf{z} 使得 $\mathbf{z}^T = (\mathbf{x}^T, \mathbf{s}_g^T)$ ，並改寫方程式(1.11)成方程式(1.12)：

$$\begin{aligned} \min_{\mathbf{z}} \quad & \mathbf{c}_z^T \mathbf{z} \\ \text{s.t.} \quad & \mathbf{a}_z^T \mathbf{z} = \mathbf{b}_z \\ & \mathbf{z} \in \mathbb{R}^{(m+n)} \end{aligned} \quad (1.12)$$

接下來可定義出矩陣形式(1.13)：

$$\begin{bmatrix} \mathbf{a}_z^T & \mathbf{b}_z \\ \mathbf{c}_z^T & 0 \end{bmatrix} \quad (1.13)$$

因為方程式(1.10)的效率角解發生在可行解空間 $\mathcal{F} \in \mathbb{R}^n$ 的角點上，而在沒有退化(degeneracy)的情況下，通常 \mathcal{F} 的角點是 n 條active拘束條件的交點，也就是這 n 條active拘束條件在方程式(1.12)的差額變數 \mathbf{s}_g 值等於零(退化是指多於 n 個 \mathbf{s}_g 值為零)，此時可以根據這些 \mathbf{s}_g 相對應的行向量把矩陣 \mathbf{a}_z^T 分解如方程式(1.14)：

$$\begin{bmatrix} \mathbf{a}_z^{TB} & \mathbf{a}_z^{TD} & \mathbf{b}_z \\ \mathbf{c}_z^{TB} & \mathbf{c}_z^{TD} & 0 \end{bmatrix} \quad (1.14)$$

如果令 \mathbf{s}_g 中值等於零的差額變數為 \mathbf{s}_g^0 ，則方程式(1.14)中 $(m \times n)$ 的矩陣 \mathbf{a}_{zT}^D 是把 \mathbf{a}_{zT} 中會與 \mathbf{s}_g^0 內積的行向量放在一起，剩餘的行向量組成 $(m \times m)$ 的矩陣 \mathbf{a}_{zT}^B ；而矩陣 \mathbf{c}_{zT} 也依此規則分類成 $(k \times m)$ 的矩陣 \mathbf{c}_{zT}^B 與 $(k \times n)$ 的矩陣 \mathbf{c}_{zT}^D 。

接下來把方程式(1.14)改寫成典型的簡算法表格如方程式(1.15)，下面爲了表達方便，將以 \mathbf{a}_{zT} 代替 \mathbf{a}_z^T ， \mathbf{c}_{zT} 代替 \mathbf{c}_z^T

$$\mathbf{t}(\mathbf{z}) = \begin{bmatrix} I & \mathbf{a}_{zT}^{B^{-1}} \mathbf{a}_{zT}^D & \mathbf{a}_{zT}^{B^{-1}} \mathbf{b}_z \\ 0 & \mathbf{R} & \mathbf{y} \end{bmatrix} \quad (1.15)$$

方程式(1.15)中 $\mathbf{z} = (\mathbf{a}_{zT}^{B^{-1}} \cdots \mathbf{b}_z, 0)^T$ ， $\mathbf{y} = -\mathbf{c}_{zT} \cdots \mathbf{z} \in \mathbb{R}^k$ ，而且 $k \times (n - m)$ 的矩陣 \mathbf{R} 稱爲每個目標函數的縮減成本係數(Reduced Cost coefficients)，如方程式(1.16)

$$\mathbf{R} = \mathbf{c}_{zT}^D - \mathbf{c}_{zT}^B \cdot \mathbf{a}_{zT}^{B^{-1}} \cdot \mathbf{a}_{zT}^D \quad (1.16)$$

矩陣 $\mathbf{R} = \{\mathbf{r}^1, \mathbf{r}^2, \dots, \mathbf{r}^{n-m}\}$ ，其意義將章節1.2.2有更詳細的描述。

方程式(1.15)是把方程式(1.13)依點 \mathbf{z} 的active狀況拆解成方程式(1.14)之後再轉換成典型的簡算法表格，而簡算法對方程式(1.15)的第 i 行做高斯運算(Pivot)的動作相當於把方程式(1.14)中 n 條active的拘束條件拿走一個，並換上第 i 條拘束條件當做active拘束條件，假設 $\bar{\mathbf{x}}$ 是鄰近於角解 \mathbf{x} 的角解之一，這個動作將使角解 \mathbf{x} 移到鄰近的角解 $\bar{\mathbf{x}}$ ，而 \mathbf{x} 與 $\bar{\mathbf{x}}$ 連成的線便是可行解空間 \mathcal{F} 中的一條邊界， E^i ，數學表示如方程式(1.17)：

$$E^i = \{(1 - \alpha)x + \alpha\bar{x} : 0 \leq \alpha \leq 1\} \quad (1.17)$$

如果把這兩點都透過矩陣 C 映射到目標空間的話，下式可以看出 \mathbf{x} 與 $\bar{\mathbf{x}}$ 的關係：

$$\mathbf{c}\bar{\mathbf{x}} = \mathbf{c}\mathbf{x} - \theta_i \mathbf{r}^i \quad (1.18)$$

方程式(1.18)中， \mathbf{r}^i 爲矩陣 \mathbf{R} 的行向量， θ_i 爲簡算法 [20]定義的相關參數，可以視爲兩點間的步伐大小。接著我們把 $\mathbf{y} = \mathbf{c}^T \mathbf{x}$ ， $\bar{\mathbf{y}} = \mathbf{c}^T \bar{\mathbf{x}}$ 代入重寫方程式(1.18)如下：

$$\bar{\mathbf{y}} = \mathbf{y} - \theta_i \mathbf{r}^i \quad (1.19)$$

從上式可以觀察到從 \mathbf{y} 到 $\bar{\mathbf{y}}$ 的方向就是 \mathbf{r}^i ，只是 \mathbf{y} 與 $\bar{\mathbf{y}}$ 的連線並不一定是可達空間的邊界，文獻 [15,21]提到當設計空間的維度數目比目標函數的個數多時，就會發生並非全部可行解空間

的角點直接對應到可達空間的角點，同樣的，並非全部可行解空間的邊界會直接對應到可達空間的邊界，在接下來的段落將討論可行解空間的邊界要對應到可達空間的邊界須具備什麼條件。

1.2.2 可達空間之邊界-架構(Frame)

矩陣 $\mathbf{R} = \{\mathbf{r}^1, \mathbf{r}^2, \dots, \mathbf{r}^{n-m}\}$ 表示在某個角解 \mathbf{x} 時有 $(n-m)$ 個改變 $\mathbf{c}\mathbf{x}$ 的方向，根據方程式(1.19)的概念我們在目標空間的 $\mathbf{c}\mathbf{x}$ 當起點再把所有 $(n-m)$ 個改變 $\mathbf{c}\mathbf{x}$ 的方向 $\{\mathbf{r}^1, \mathbf{r}^2, \dots, \mathbf{r}^{n-m}\}$ 畫出如圖1.3，從圖中可以看出，這些向量中只有最外邊的兩個向量 \mathbf{r}^1 與 \mathbf{r}^5 才會讓我們找到可達空間的邊界。

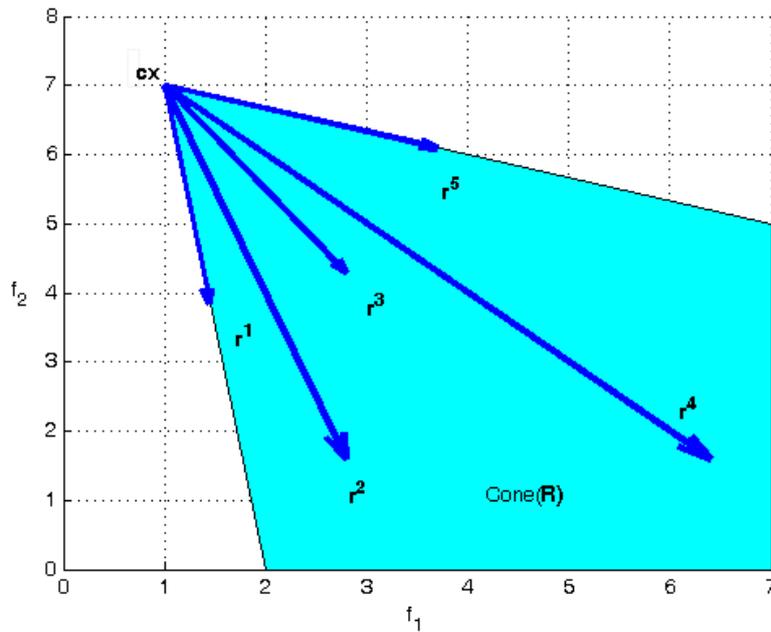


圖 1.3: \mathbf{R} 矩陣的行向量與錐體

我們定義 Id_M 是矩陣 \mathbf{M} 中所有行向量的索引所組成的集合，例如有一個 $k \times h$ 的矩陣 $\mathbf{M} = \{\mathbf{m}^1, \mathbf{m}^2, \dots, \mathbf{m}^h\}$ ，則 $Id_M = \{1, 2, \dots, h\}$ ；並定義 Id_F 是矩陣 \mathbf{M} 的架構 \mathbf{F} 所有行向量的索引所組成的集合， $Id_F \subset Id_M$ ，那我們定義一個錐體 ($Cone(\mathbf{M})$) 如方程式(1.20)：

$$cone(\mathbf{M}) = cone(\mathbf{m}^i : i \in Id_M) = \left\{ \mathbf{m} \in \mathcal{R}^k : \mathbf{m} = \sum_{i \in Id_M} \alpha_i \mathbf{m}^i, \alpha_i \geq 0 \right\} \quad (1.20)$$

方程式(1.20)中描述的錐體 $Cone$ 是一個集合，此集合是由矩陣 \mathbf{M} 中所有行向量的線性組合；如圖1.3中，深色區塊代表矩陣 \mathbf{R} 的錐體 $Cone(\mathbf{R})$ ，在深色區塊中以 $\mathbf{c}\mathbf{x}$ 為起點，一定可以找到

一組適當的係數 α 讓任意一個向量是矩陣 \mathbf{R} 中所有行向量的線性組合。另外我們可以從圖直接觀察到，要描述 R 矩陣的錐體，只要使用向量 \mathbf{r}^1 與 \mathbf{r}^5 做線性組合就可以完整描述了，接下來我們對向量 \mathbf{r}^1 與向量 \mathbf{r}^5 的特性做一個較為完整的定義。

我們定義錐體 $Cone(\mathbf{M})$ 的架構(Frame)， \mathbf{F} ，是矩陣 \mathbf{M} 某些行向量組成的集合，此集合符合特性： $cone(\mathbf{m}^i : i \in Id_{\mathbf{F}}) = cone(\mathbf{M})$ ，然後對每個 $j \in Id_{\mathbf{F}}$ 來說， $cone(\mathbf{m}^i : i \in Id_{\mathbf{F}} \setminus \{j\}) \neq cone(\mathbf{M})$ 。

上面這段錐體的定義除了說明架構 \mathbf{F} 以所有線性組合的向量形成的集合，錐體，會跟原來的矩陣 \mathbf{M} 相同之外，也說明了組成這個錐體的 \mathbf{M} 矩陣裡，必須包含架構的行向量，否則剩餘的行向量所展開的錐體必與原矩陣 \mathbf{M} 展開的錐體不同。

在文獻 [19]提出當 \mathbf{r}^i 是 $cone(\mathbf{R})$ 的架構時，設計空間的邊界 E^i 才會直接對應到目標空間的邊界，因此如何決定矩陣 R 之中哪些行向量是其架構，便是在目標空間計算線性多目標最佳化Pareto集合的重要步驟之一，而決定架構的原則描述如下：在一個由 h 個 k 維行向量所組成的矩陣 \mathbf{M} ，其架構應該會有 k 個行向量，因此在一開始，先在 \mathbf{M} 中任意挑 k 個行向量當作有可能成為架構的向量，對矩陣 \mathbf{M} 中這 k 個行向量做高斯運算(Pivot)，使矩陣 \mathbf{M} 變成 $k \times k$ 的單位矩陣與其他行向量的組合，這時候觀察其他行向量的分量值，假如有某個行向量的所有分量都大於零，則表示此行向量是其他行向量的線性組合，因此這個向量不會是架構，便將它排除，經過一連串有系統的計算，最後可以找出矩陣 \mathbf{M} 的架構 \mathbf{F} ，文獻 [22]提供一個良好的演算流程來計算出一個矩陣的架構(Frame)。

1.2.3 在目標空間計算線性Pareto集合演算法

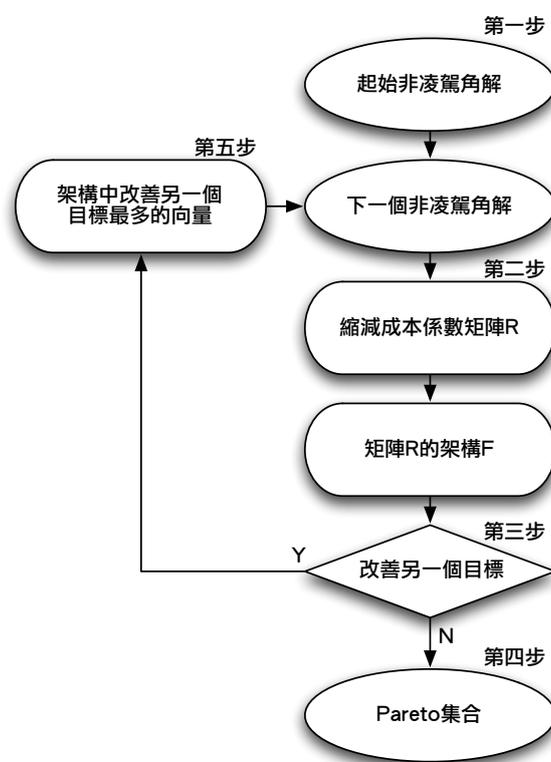


圖 1.4: 線性多目標最佳化Pareto集合演算法

圖1.4是處理線性雙目標最佳化的演算法，使用的是文獻 [19]提出在目標空間計算Pareto集合的類簡算法，從其中一個非凌駕角解移動到下一個非凌駕角解，直到所有的非凌駕角解全被找出來，下面是此演算法的細節：

演算法細節

第一步 計算任一目標的單目標最佳解：若選擇第*i*個目標，即選擇了矩陣*c*第*i*行做為目標函數的係數，則計算單目標最佳化問題(1.21)的最佳解，此解一定是Pareto集合中的其中一個非凌駕角解。

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}_i^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{a}_g^T \mathbf{x} - \mathbf{b}_g \leq 0 \end{aligned} \quad (1.21)$$

第二步 計算非凌駕解的縮減成本係數矩陣*R*，並且根據 [22]判斷出矩陣*R*的架構*F*。

$$\mathbf{R} = \mathbf{c}_z^{TD} - \mathbf{c}_z^{TB} \cdot \mathbf{a}_z^{TB^{-1}} \cdot \mathbf{a}_z^{TD}$$

第三步 檢查架構 \mathbf{F} 中的兩個向量是否能夠改善另一個目標函數(相對於第一步)，即有任意一個 p 與 q 能滿足方程式(1.22)，則進入到第五步；如果不能改善，進入第四步。

$$\begin{aligned} \mathbf{F}(p, q) &< 0 \\ p &= 1, \dots, k, \quad p \neq i \\ q &= 1, \dots, k \end{aligned} \quad (1.22)$$

第四步 結束演算法，並把所有找到的非凌駕解依序連接起來，整條線即為此最佳化問題的Pareto集合。

第五步 選擇架構 \mathbf{F} 中能夠改善另一個目標(相對於第一步)最多的向量：在本篇論文中挑選向量的方法為先將架構 \mathbf{F} 中的行向量皆轉換為長度是一的單位向量，接著觀察所有的單位向量在另一個目標(相對於第一步)的分量，分量值負值最小的向量即為架構 \mathbf{F} 中能夠改善另一個目標(相對於第一步)最多的向量；接著找出以此向量能改善的最大距離，即求解下列線性最佳化問題(1.23)

$$\begin{aligned} \min_{\theta} & -\theta \\ \text{s.t.} & \mathbf{c}_z^T \mathbf{z} + \theta \mathbf{r} = \bar{y} \\ & \mathbf{a}_z^T \mathbf{z} = \mathbf{b}_z^T \\ & \theta \geq 0, \mathbf{z} \geq 0 \end{aligned} \quad (1.23)$$

方程式(1.23)第一條拘束條件是希望目標函數可以依著第五步決定的向量 \mathbf{r} 來改變；而第二條拘束條件是限制設計點必須在可行解空間裡面；找到最大的 θ^n 值後，新的非凌駕解 $\mathbf{y}^n = \bar{\mathbf{y}} - \theta^n \mathbf{r}$ 就可以被找到，鄰近兩個非凌駕解構成的一段在目標空間的邊界 $E = \{\mathbf{y} - \theta \mathbf{r} : 0 \leq \theta \leq \theta^n\}$ 。找到鄰近的新非凌駕角解之後進入第二步。

1.2.4 範例

上面介紹了線性多目標最佳化的Pareto集合計算方法，下面將用一個數學範例 [19,23]來示範，並且使用拘束法來做檢驗。

$$\begin{aligned}
 \min \quad & \mathbf{c}\mathbf{x} = \begin{pmatrix} -x_1 - \frac{1}{9}x_3 \\ -x_2 - \frac{1}{9}x_3 \end{pmatrix} & (1.24) \\
 \text{s.t.} \quad & 9x_1 + 9x_2 + 2x_3 - 81 \leq 0 \\
 & 8x_1 + x_2 + 8x_3 - 72 \leq 0 \\
 & x_1 + 8x_2 + 8x_3 - 72 \leq 0 \\
 & -7x_1 - x_2 - x_3 + 9 \leq 0 \\
 & -x_1 - 7x_2 - x_3 + 9 \leq 0 \\
 & -x_1 - x_2 - 7x_3 + 9 \leq 0 \\
 & x_1 - 8 \leq 0 \\
 & x_2 - 8 \leq 0 \\
 & x_i \geq 0, i = 1, 2, 3
 \end{aligned}$$

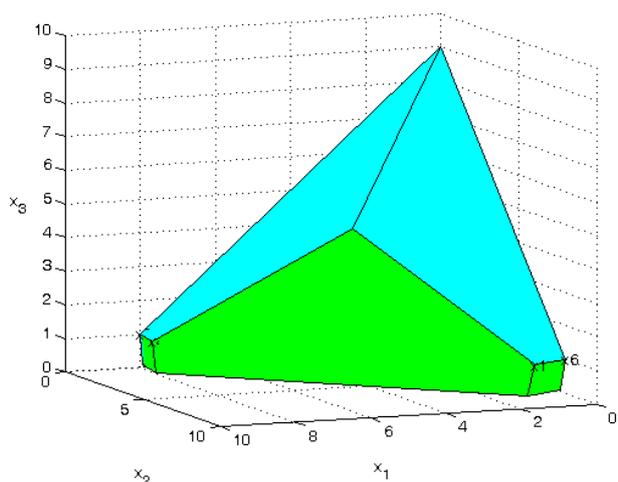


圖 1.5: 範例(1.24)的設計空間

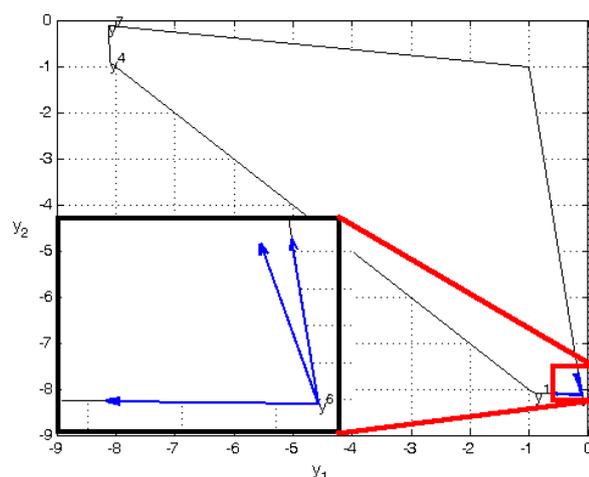


圖 1.6: 範例(1.24)的目標空間

這是一個線性雙目標最佳化問題，其中 $k = 2$ ， $n = 3$ ， $m = 8$ ，由圖1.5有更進一步的了解。根據章節1.2.3中的介紹，第一步我們假如先找第一個目標 f_2 的最佳解當作起始的非

凌駕解 $\mathbf{y}^6 = (-0.111, -8.111)^T$ ， $\mathbf{x}^6 = (0, 8, 1)^T$ ，在此非凌駕解上可以求出相對應的矩陣 \mathbf{R} 如方程式(1.25)，假如把矩陣 \mathbf{R} 中所有的行向量以單位向量的方式， \mathbf{y}^6 為起點繪製在目標函數空間如圖1.6，即可以看出這三個向量之中， $\mathbf{c}\mathbf{x}$ 沿著中間的向量不管如何做變化，都會是另外兩個向量的線性組合，因此不會是非凌駕解 [19]，更說明此問題在設計空間裡所有的角解不會直接對應到目標空間的非凌駕角解 [15,21]。在此架構 \mathbf{F} 的兩個行向量中，可以看出沿著指著左方的向量會比另一個向量使第二個目標 f_1 更容易改善，因此延著此向量計算方程式(1.23)所描述的最佳化問題，找到最大的步幅大小 θ^n 後就能夠找到下一個非凌駕解 $\mathbf{y}^1 = (-0.9 - 8.1)^T$ ， $\mathbf{x}^1 = (0.8, 8, 0.9)^T$ 。

$$R = \begin{pmatrix} 0.0141 & 0.002 & 0.8889 \\ -0.004 & -0.1434 & -0.1111 \end{pmatrix} \quad (1.25)$$

依照此演算法直到找到的架構 \mathbf{F} 不能改善 f_1 時即找到這問題的Pareto集合了，如圖1.7。計算過程的四個非凌駕角解如表1.1所示，表中列出問題1.24的每一個非凌駕角解在目標空間的位置，跟相對應的縮減成本係數矩陣 R ，還有此非凌駕角解在設計空間的位置 x_1, x_2, x_3 ，以及到下一個非凌駕角解所需要的距離 θ ；而表1.1的縮減成本係數矩陣中，中括號的行向量代表此矩陣 R 的架構 \mathbf{F} ，粗體表示的行向量則代表此非凌駕角解將由此行向量移動 θ 的距離至下一個非凌駕角解(例如 \mathbf{y}^6 沿著 R 矩陣中的粗體向量移動0.789的距離到 \mathbf{y}^1)。

圖1.7同時使用拘束法來驗證，可以看到使用拘束法計算出來的50個非凌駕解都在 [19]提出的方法計算出的Pareto集合線上，換句話說，假如使用拘束法計算無限多個非凌駕解，理論上就可以與 [19]完全相同。因為 [19]可以完整並正確的對Pareto集合中的每一個非凌駕解做描述，而且在得到完整的Pareto集合的前提下，計算量遠遠低於拘束法，因此本論文將採用此方法來計算Pareto集合。

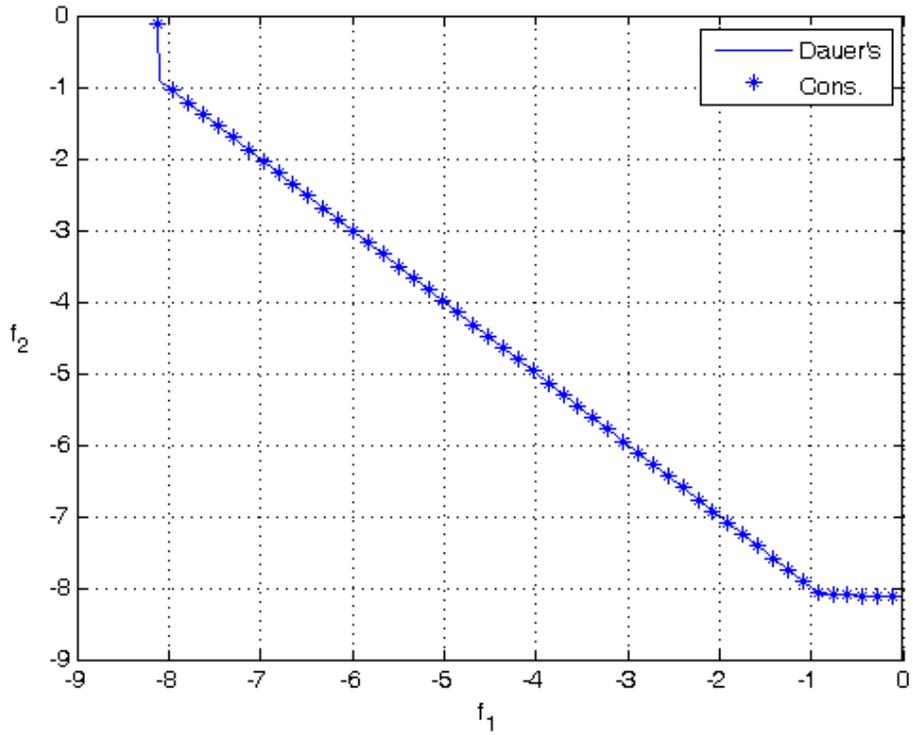


圖 1.7: Pareto 集合之Daurer's method與Constraint method比較

表 1.1: 線性Pareto集合運算過程

非凌駕角解	f_1	縮減成本係數矩陣R			θ	x_1	x_2	x_3
	f_2							
y^6	$\begin{pmatrix} 8.111 \\ 0.111 \end{pmatrix}$	$\begin{bmatrix} 0.014 \\ -0.004 \end{bmatrix}$	$\begin{bmatrix} \mathbf{0.002} \\ -\mathbf{0.143} \end{bmatrix}$	$\begin{bmatrix} 0.889 \\ -0.111 \end{bmatrix}$	0.789	8.000	0.000	1.000
y^1	$\begin{pmatrix} 8.100 \\ 0.900 \end{pmatrix}$	$\begin{bmatrix} -0.002 \\ 0.113 \end{bmatrix}$	$\begin{bmatrix} \mathbf{0.014} \\ -\mathbf{0.014} \end{bmatrix}$	$\begin{bmatrix} 0.900 \\ -0.900 \end{bmatrix}$	10.182	8.000	0.800	0.900
y^4	$\begin{pmatrix} 0.900 \\ 8.100 \end{pmatrix}$	$\begin{bmatrix} \mathbf{0.113} \\ -\mathbf{0.002} \end{bmatrix}$	$\begin{bmatrix} -0.014 \\ 0.014 \end{bmatrix}$	$\begin{bmatrix} -0.900 \\ 0.900 \end{bmatrix}$	0.789	0.800	8.000	0.900
y^7	$\begin{pmatrix} 0.111 \\ 8.111 \end{pmatrix}$	$\begin{bmatrix} -0.004 \\ 0.014 \end{bmatrix}$	$\begin{bmatrix} -0.143 \\ 0.002 \end{bmatrix}$	$\begin{bmatrix} -0.111 \\ 0.889 \end{bmatrix}$	NA	0.000	8.000	1.000

1.3 不確定因素對多目標最佳化的影響

工程上在進行設計時，爲了要求節省時間成本，會簡化許多較爲不重要影響的因素，只考慮影響較大的因素，工程上儘管我們設計時使用的模型多完善，還是無法把大自然的因素全部考慮進去。除了產品在製作過程的影響因素，像溫度變化，或是人爲誤差等造成的不確定性，還有可能因爲一開始模型的假設過於簡化，造成的誤差種種，都將使不確定因素影響我們的設計成果，使其結果與起初想像的設計成果有些許的差距，輕微的可能使產品的使用年限不穩定，嚴重的話更會影響產品生產時的良率造成公司虧損，因此在設計過程考慮不確定因素乃爲重要的事情。

不確定因素的考慮第一步便是了解到大部分的數值不是定值，而是呈現某種分佈狀況，而現在有許多分佈的模型可以描述這個世界的某些行爲：

- 指數分佈可以用來描述第一事件或相鄰兩事件發生所需要的時間，例如地震 [24]。
- 韋氏分布適合描述風的特性 [25]

而大部分的情況都可以用高斯分佈(又稱常態分佈)來描述，只要找到適當的分佈模型來描述，就可以用此模型來代替真實世界來進行分析或設計，而本篇論文使用高斯分布來描述不確定因素的分佈情況。

高斯分佈

當數值是呈現某種分佈狀態的時候，此時某個數值出現的可能性便可以用機率來表示，首先定義 X 爲隨機變數，當 X 爲連續時， f_X 定義爲機率密度函數(Probability Density Function)，是其一般形式，方程式(1.26)是對 f_X 從 a 積分到 b 得到的累積密度機率密度函數，表示爲 X 在某一區間 $[a, b]$ 內所發生的機率。

$$\int_a^b f_X(t)dt = \Pr[a \leq X \leq b] \quad (1.26)$$

採用不同的分佈當模型就會使用不同的機率密度函數；而高斯分佈是現在廣被人使用模擬不確定因素的模型，其機率密度函數如方程式(1.27)。

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x - \mu_X}{\sigma}\right)^2\right], \quad \forall x \in [-\infty, \infty] \quad (1.27)$$

但爲了計算上方便，將此高斯分佈進行標準化(Standardize)，如方程式(1.28)：

$$U = \frac{X - \mu_X}{\sigma_X} \quad (1.28)$$

其用意是把座標原點移至平均值 μ_x 上，並將標準差 σ 轉爲1，轉化後的高斯分佈可以改寫成方程式(1.29)

$$\phi(u) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right) \quad (1.29)$$

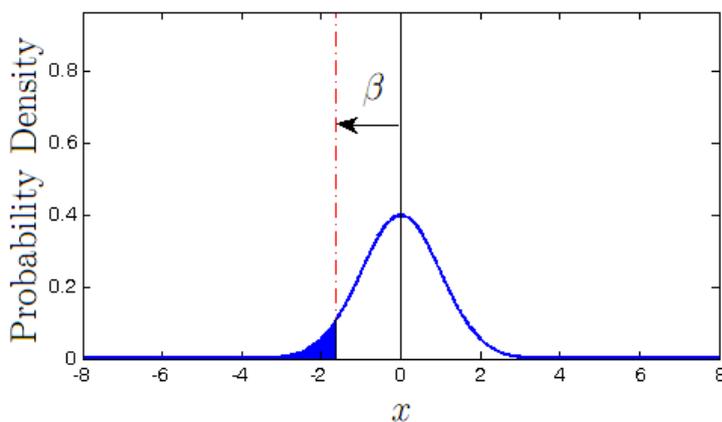


圖 1.8: 標準高斯分佈機率密度

圖1.8是平均值在0，標準差爲1的標準高斯分佈圖，x軸是量值，而y軸是出現此量值的機率密度；從圖上可以觀察到高斯分佈是個以平均值左右對稱的分佈，呈現一個鐘型曲線。而此鐘型曲線下的面積即爲此數值發生的機率，對方程式(1.29)積分即可得到累積密度函數方程式(1.30)。

$$\Phi(u) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^u \exp\left(-\frac{u^2}{2}\right) du = \Pr[U \leq u] \quad (1.30)$$

在決定使用什麼分佈來模擬不確定因素之後，就能夠利用此分佈的累積機率函數來計算可靠度；事情卻沒有這麼簡單，雖然我們有機率密度的函數，理論上我們只要對其積分就可以得到機率，但實際上方程式(1.30)沒有解析解，只能透過數值積分來得到；更甚者我們很難去獲得一個事件的機率密度函數 f_X ，因此我們下面介紹常用來估計機率的方法 [26,27]。

1.3.1 可靠度分析

市面上的電子產品大多有註明保固期，在保固期限內產品壞掉，失去它的功能時廠商願意以較低價格的方式維修，更甚者廠商願意替消費者更換新品。保固的觀念除了是消費者與廠商間的遊戲規則之外，還代表了不是每個產品都能在一段固定的時間範圍內維持正常的功能，代表了實際上的物理量值(長度，重量，溫度，濕度等等)都不是一個定值而是某些數值出現的頻率較高或較低的一個不確定性因素，導致受到這些不確定性因素影響的產品功能也不會有固定一種表現情況，因此需要以可靠的程度來提供更多資訊。

有個對可靠度的英文定義 [28]如下：

The probability of that a system, vehicle, machine, device, and so on will perform its intended function under operating conditions, for a specified period of time.

這個定義說明一個系統，或是車子，機器，或是任何一個在適當操作下能夠在一段時間內擁有其功能的機率，稱為此東西的可靠度。因此可靠度有三個重要的東西組成：機率，時間與功能：

機率 由於不確定因素的影響，產品在某種機率下可以毫無故障的發揮其功能，是可靠度的度量方式。

時間 產品在不同的時間也會有不同的表現，有可能會疲勞，老化而導致產品喪失功能，因此時間可以視為一種不確定因素，嚴格來說，可靠度必須在某段固定時間內做討論才有意義；但是在考慮時間之前，必須先達到產品應有的功能，在本論文中將以在考慮時間之前，不能先破壞的目標做討論，不會考慮時間的因素。

功能 功能為判定產品是否可靠的一個標準，只要產品喪失應有的功能，就會判定此產品為破壞。

在不考慮時間的影響為前提下，可靠度是產品能達到正常功能的機率，也可以換成計算某產品不能達到正常功能的機率來得到可靠度的資訊，稱為破壞機率，可靠度 R_e 與破壞機率 P_f 為一體兩面的觀念，這兩者的關係如方程式(1.31)

$$R_e = 1 - P_f \quad (1.31)$$

工程上常用來估計機率的方法主要分為三類

- 樣本法-蒙地卡羅法
- 一階可靠度方法
 - 一階二次可靠度方法
 - 進階一階二次可靠度方法
- 二階可靠度方法

樣本法

蒙地卡羅是樣本法中最具代表性的機率估計法，其概念便是以某種分佈的樣本點在某個想要觀察機率的參考點附近大量的嘗試樣本點是破壞還是安全，最後再搜集所有樣本點的資訊來估計此參考點的機率。假設一個拘束函數 $g(\mathbf{X}) \geq 0$ 是破壞，在我們對計算機率的函數 $g(\mathbf{X})$ 不了解時，用蒙地卡羅法是最好的方法，其計算機率的方式如方程式(1.32)：

$$P_f = \frac{n_f}{n_s} \quad (1.32)$$

n_f 為樣本點中違反破壞機率的點數，也就是使 $g(\mathbf{x}) \geq 0$ 的樣本點； n_s 為所有的樣本點數，而 P_f 則為此蒙地卡羅法所估計的破壞機率。雖然蒙地卡羅法在估計可靠度時能有一定的準確度，但是要達到某種程度的準確度就需要一定的樣本數，文獻 [28]中提供了一個蒙地卡羅法樣本點數與準確度的關係方程式如(1.33)：

$$\epsilon\% = \sqrt{\frac{(1 - P_f^T)}{N \times P_f^T}} \times 200\% \quad (1.33)$$

方程式(1.33)中 P_f^T 是真實的破壞機率，說明了如果要計算1%的破壞機率，而誤差 $\epsilon\%$ 希望可以在2%以下的話，需要樣本點數為 10^6 個，如果以此數目來進行設計計算量稍嫌過大，下面介紹較為快速的方法。

一階可靠度方法

一階可靠度方法都是將拘束條件做一階泰勒展開，而形成線性的拘束條件後再考慮此線性拘束條件對設計點的破壞機率來估計可靠度；其中又以採取對拘束條件做一階泰勒展開的參考點不同而分成：一階二次可靠度方法及進階一階二次可靠度方法。

● 一階二次可靠度方法

此方法是在設計點對拘束條件進行一階泰勒展開，而且考慮可靠度的方法是使用兩個統計量 μ 與 σ 的比值來計算所需要的可靠度指標 β ，再用方程式 $P_f = \Phi(-\beta)$ 計算其破壞機率，計算步驟如下：

- 步驟一：在設計點上對某個拘束條件 $g(\mathbf{X})$ 做如方程式(1.34)的一階泰勒展開成線性拘束條件 $g'(\mathbf{X})$

$$g'(\mathbf{X}) = g(\boldsymbol{\mu}_X) + \sum_{i=1}^n \frac{\partial g}{\partial x_i} (X_i - \mu_{X_i}) \quad (1.34)$$

- 步驟二：計算 $g'(\mathbf{X})$ 的標準差

$$\sigma_{g'} = \sqrt{\sum_{i=1}^n \left(\frac{\partial g_j}{\partial x_i}\right)^2 \sigma_{X_i}^2} \quad (1.35)$$

- 步驟三：利用平均值與標準差計算可靠度指標 β

$$\beta = \frac{\mu_{g'}}{\sigma_{g'}} \quad (1.36)$$

再利用此 β 來計算破壞機率

$$P_f = \Phi(-\beta) \quad (1.37)$$

此方法可以很快將估計的破壞機率計算出來，但此方法只有在拘束條件為線性的時候最準確，其計算的破壞機率會依拘束條件的非線性程度提高而誤差也跟著提高。而且此方法只有考慮當不確定因素為高斯分佈的時候才適用，為了使其他分佈的不確定因素模擬也能使用此法，進階的一階二次可靠度方法被提出。

● 進階一階二次可靠度方法

此方法基本上也是對拘束條件進行一階泰勒展開，只是在這之前會將問題轉換至標準空間後，在拘束條件上找一個最有可能破壞的點做為展開點，而且此方法的可靠度指標 β 計算方式也有所改善。其步驟如下：

- 步驟一：如方程式(1.38)將隨機變數轉換至標準空間中，做一個把座標軸原點移到設計點上(平均值)，並對所有變數的標準差標準化。

$$\mathbf{U} = \frac{\mathbf{X} - \mu_X}{\sigma_X} \quad (1.38)$$

- 步驟二：找到拘束條件極限狀態($g(\mathbf{X}) = 0$)上的點，而此點必須與標準空間的原點距離最短，稱此點為最有可能破壞的點(MPP)，找尋此點是一個最佳化過程。

$$\begin{aligned} \min_{\mu_U} \quad & \mu_U^T \cdot \mu_U \\ \text{s.t.} \quad & g(\mu_U) = 0 \end{aligned} \quad (1.39)$$

- 步驟三：在方程式(1.39)求解最佳化的結果，其中可靠度指標 $\beta = \sqrt{\mu_U^T \cdot \mu_U}$ ，再利用方程式(1.40)計算設計點對此線性拘束條件的破壞機率

$$P_f = \Phi(-\beta) \quad (1.40)$$

這兩種一階二次可靠度方法對於處理同一個線性拘束條件，設計變數的變異量又一樣的問題時，兩種方法計算出來的效果都一樣準確，只是進階一階二次可靠度方法的計算較為繁雜，為了兼顧計算效率與準確度，本論文將採用一階二次可靠度方法來處理線性的拘束條件。

二階可靠度方法

由於一階可靠度方法在計算非線性拘束條件的時候，會因為非線性程度越大而使得計算破壞機率的誤差跟著變大，因此Fiessler等人在1979年提出二階可靠度方法 [29]，藉由對拘束條件做二次泰勒展開得到的逼近函數來改善非線性引起的誤差。由於本論文專注在線性拘束條件的可靠度估計，因此不對此做深入的探討。

1.3.2 可靠度最佳化設計

一般最佳化問題並未考慮不確定因素，而找到的最佳解卻容易因為不確定因素而使得最佳解的可靠度只有50%甚至更低(因為緊鄰拘束條件)，因此為了更加符合實際情況，也就是考慮不確定因素進入設計過程，原本的最佳化問題改寫成方程式(1.41)

$$\begin{aligned} \min_{\boldsymbol{\mu}_x} \quad & f(\boldsymbol{\mu}_x) \\ \text{s.t.} \quad & Pr[\mathbf{g}(\mathbf{X}) > \mathbf{0}] \leq P_f \end{aligned} \quad (1.41)$$

此方程式中把原本定義拘束條件 $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$ 為不會破壞，改成以機率的形式來表示：設計點對拘束條件破壞的機率要比決策者訂定可接受的破壞機率 P_f 來的小，而方程式(1.41)稱為可靠度最佳化(Reliability Based Design Optimization) [30]。基本上求解的過程與方法與不考慮可靠度的方法一樣，只是必須使用上面介紹的可靠度估計法來估計機率。

可靠度多目標最佳化設計

而現在我們可以把多目標最佳化延伸至考慮不確定因素的形式，如方程式(1.42)

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{f} = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})\} \\ \text{s.t.} \quad & Pr[\mathbf{g}(\mathbf{X}) > \mathbf{0}] \leq P_f \end{aligned} \quad (1.42)$$

這類的可靠度多目標最佳化設計問題，被許多人所關注與討論：Levi與Gobbi在2005年提出考慮不確定因素下在結構設計的多目標最佳化應用 [31]，文章中Levi等人以一階二次可靠度方法來考慮不確定因素，而其對I型梁，T型梁與O型梁做輕量與強度的雙目標設計，文章中除了說明T型梁中翼在下方的設計比翼在上方的設計來的好之外，如圖1.9，還說明了T型樑的截面的長寬比例越接近則此類設計越可靠，如圖1.10。另外Li等人在2008年在 [32]討論汽車側邊框架的設計，說了在不同公差配置下成本與敏感度的權衡得失，如圖1.11，公差配置與考慮可靠度有某種層面上的相似。

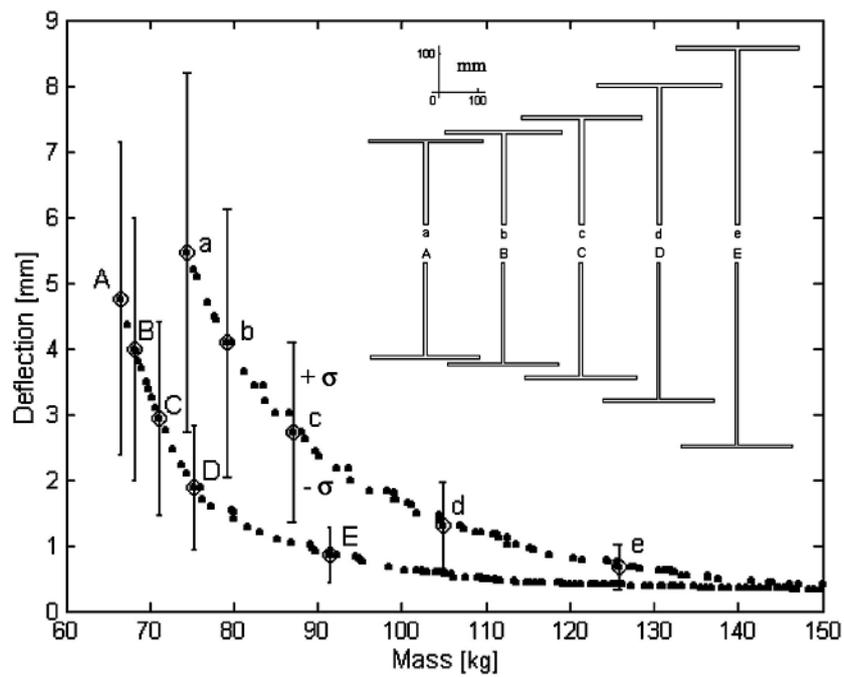


圖 1.9: 文獻[31]中T型梁上翼與下翼的比較

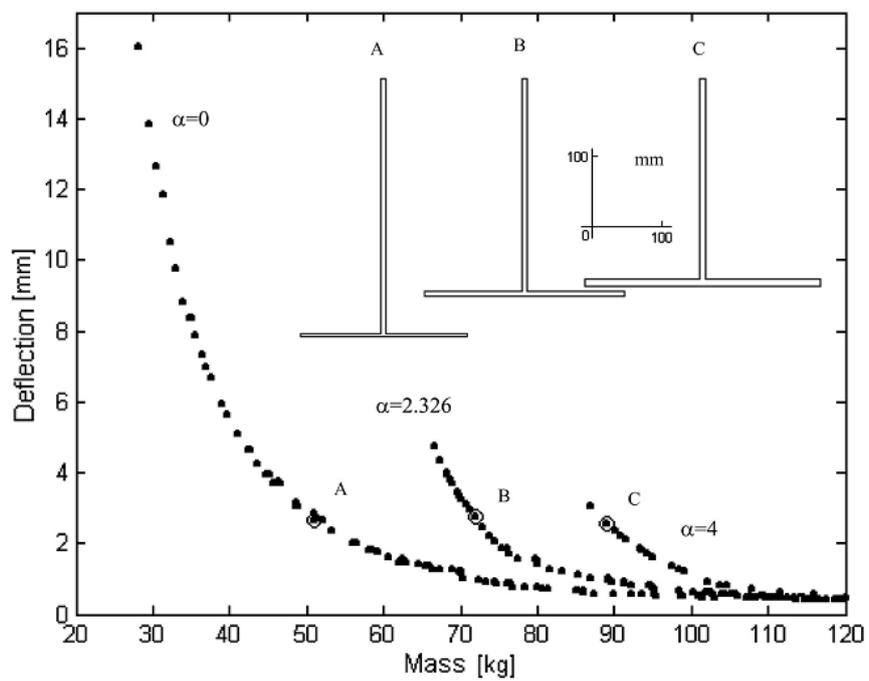


圖 1.10: 文獻[31]中T型梁長寬比例的比較

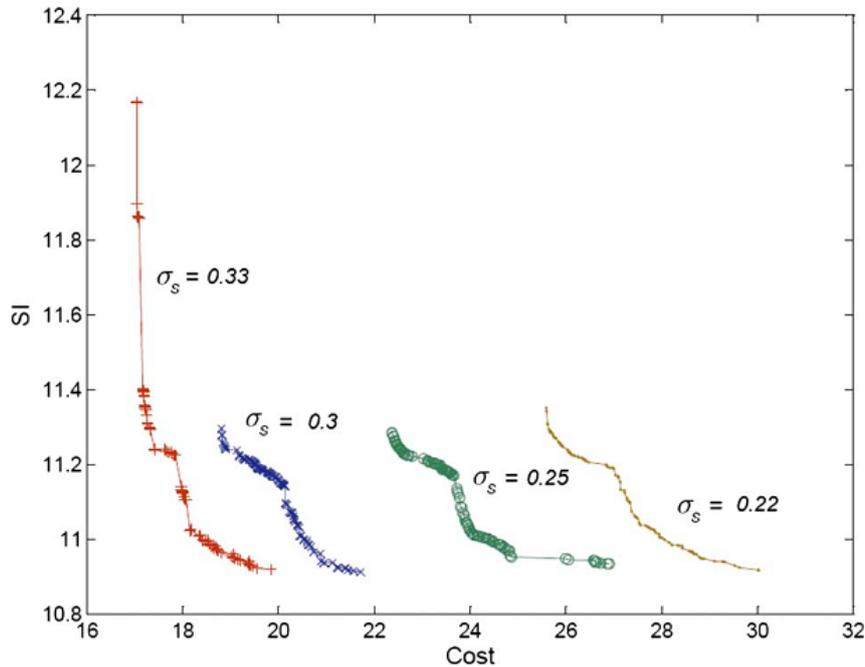


圖 1.11: 文獻[32]中在不同公差配置下的成本與敏感度關係圖

1.4 研究動機與研究方法

在做完這些文獻探討之後，觀察到最後圖1.10，圖1.11中的Pareto集合在不同可靠度的考慮下，會有移動的情形，而且圖1.11的鄰近的幾個Pareto集合形狀都很接近，似乎可以用平移再加上其他動作把較低可靠度的Pareto集合拿來預測較高可靠度的Pareto集合，這樣Pareto集合移動的情形除了 [33]之外並沒有人探討其情形。

文獻 [33]中提到，線性的多目標最佳化在考慮可靠度的情況下，假如active的拘束條件沒有改變，Pareto集合會有平移的特性，而且平移的距離與 β 成正比。本論文使用在目標空間解決線性多目標問題的方法 [19]來對此現象做討論。

假如能夠了解可靠度多目標最佳化的Pareto集合移動情形，那下一步很有機會的便是拿已有的Pareto集合用來預測尚未計算的Pareto集合，或者是把一個可靠度範圍內Pareto集合的變化情形描述出來，提供決策者更多資訊，關於考慮不確定因素下的影響，協助決策者做更為實際的決定。

為了達到上述目標，本論文先以相較於非線性多目標最佳化問題，較為不複雜的線性雙目標最佳化問題著手，並假設所有設計變數都是互相獨立的高斯分佈，其標準差皆為1，而

且不討論等式拘束條件。討論線性的Pareto集合在考慮不確定因素下將如何移動；再試著將此討論的成果延伸至非線性的多目標最佳化問題。

本文第二章將開始討論在不確定因素下，使用一階二次可靠度方法的線性拘束條件形式與先前介紹有何不同以及為何如此；接著討論在不確定的考量下拘束條件active的變化，並使用最佳化方法避開因為active情況改變，而無法使用平移的方式預測的困難；同時使用在目標函數下計算 β -Pareto集合的方法來再次討論線性Pareto集合具有平移可預測的特性，最後將以 β -Pareto集合圖來提供一個可靠度範圍內任意可靠度的線性Pareto集合預測演算法以及數學範例。

第三章將討論如何使用區塊三明治夾擠法延伸線性雙目標最佳化的技術，處理非線性雙目標最佳化問題，並提出一個數學範例做示範；第四章為結果與展望。



第二章 不確定因素下線性問題的Pareto集合平移

在不確定因素下，因為線性拘束條件的平移，Pareto集合會有平移的現象 [33]，文獻中也提到一個方法來預測不同可靠度的Pareto集合，但是那只能適用在預測兩個可靠度之間沒有active的拘束條件改變，因此這篇論文將提出一個方法，來處理active的拘束條件改變時的Pareto集合預測；首先介紹一下不確定因素與其造成的影響。

2.1 不確定因素對線性拘束條件的影響

在討論了最佳化與不確定因素後，假如拘束條件為線性時，如何將此條件改為機率形式，以及進行最佳化過程時該使用何種可靠度估計法最恰當，為接下來要討論的內容。

先前已經討論到線性拘束條件使用一階二次可靠度方法是很合適的，而且因為拘束條件已經是線性的，不需再計算一階二次可靠度方法步驟一的一階泰勒展開，詳細處理過程如下：假設一個一維的設計變數 X 是呈現標準高斯分佈如圖2.1，原點位於設計點的平均值 μ_X 上；而其虛線代表某一條拘束條件(2.1)的極限狀態 $g(x') = 0$ ，虛線左側代表破壞，右側代表安全，因此虛線左側的色塊面積代表此設計點對這拘束條件的破壞機率 P_f 。

$$g(x') \leq 0, \quad x' \in R^1 \quad (2.1)$$

$$P(g(X) \geq 0) \leq P_f \quad (2.2)$$

可以想像在原本不考慮可靠度時的最佳化過程是一個設計點在可行解空間中有系統的找最佳解；而考慮不確定因素之後，假設此 P_f 是設計者能容許的最大破壞機率，那麼設計點 μ_X 就不能再往左邊移動，因為一往左移破壞機率就會大於 P_f ，因此拘束條件(2.1)改寫成(2.2)的機率形式。可以視為一個半徑為 β 的球在原本的可行解空間找最佳解，而 β 計算方式如同方程式(2.3)

$$P_f = 1 - R_e = \Phi(-\beta) \quad \longrightarrow \quad \Phi^{-1}(R_e) = \Phi^{-1}(1 - P_f) = \beta \quad (2.3)$$

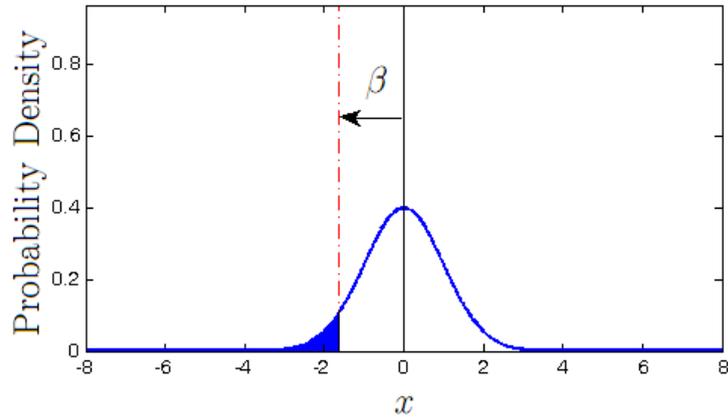


圖 2.1: 與圖1.8相同的標準高斯分佈機率密度

此時球心的活動被限制在原點以右的範圍，因此有另一種看法是限制設計點，最多只能靠近拘束條件距離 β 的位置 [26]

$$\beta = \frac{\mu_g}{\sigma_g} \longrightarrow \mu_g - \sigma_g \beta = 0 \quad (2.4)$$

$$\mu_g = g(\mu_X), \mathbf{X} \in R^n, \quad \sigma_g = \sqrt{\sum_{i=1}^n \left(\frac{\partial g}{\partial X_i}\right)^2 \sigma_{X_i}^2}$$

方程式(2.4)中，假如視 σ_{X_i} 皆為1，則 β 的計算式等於我們熟悉的距離公式，計算設計點 μ_X 與 $g(\mathbf{X}) = 0$ 的距離；因此新的問題形式可以將方程式(2.3)帶入方程式(2.4)並修改成方程式(2.5)，拘束條件經過方程式(2.5)平移後，重新進行最佳化。

$$g(\mu_X) - \sigma_g \Phi^{-1}(1 - P_f) = 0 \quad (2.5)$$

2.2 拘束條件active情況在考慮不確定因素下的改變

在考慮不確定因素下，會使得線性拘束條件平移而使得可行解空間縮小，而在某個可靠度下會導致非凌駕角解的active拘束條件改變，這會導致我們無法用直接平移現有Pareto集合的方式來預測其它可靠度下的Pareto集合 [33]，下面將用一個簡單的數學範例來說明。

方程式(2.6)為一個兩個目標函數，兩個設計變數，五個拘束條件的問題，並且同時計算可靠度50%與可靠度98%的情況：

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}\mathbf{x} = \begin{pmatrix} 4x_1 - 2x_2 \\ -2x_1 + 5x_2 \end{pmatrix} & (2.6) \\ \text{s.t.} \quad & g_1: -5x_1 + x_2 + 10 - \sigma_1\Phi^{-1}(1 - P_f) \leq 0 \\ & g_2: -3x_1 - 4x_2 + 25 - \sigma_2\Phi^{-1}(1 - P_f) \leq 0 \\ & g_3: x_1 - 5x_2 + 10 - \sigma_3\Phi^{-1}(1 - P_f) \leq 0 \\ & g_4: x_2 - 8 - \sigma_4\Phi^{-1}(1 - P_f) \leq 0 \\ & g_5: x_1 - 8 - \sigma_5\Phi^{-1}(1 - P_f) \leq 0 \\ & \sigma_{x_i} = 1, \quad i = 1, 2, \quad \sigma_j = \sqrt{\sum_{i=1}^2 \left(\frac{\partial g_j}{\partial x_i}\right)^2 \sigma_{x_i}^2}, \quad j = 1, 2, 3, 4, 5 \end{aligned}$$

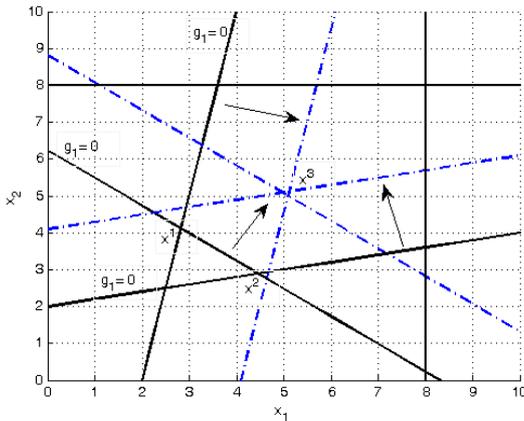


圖 2.2: 範例(2.6)設計空間

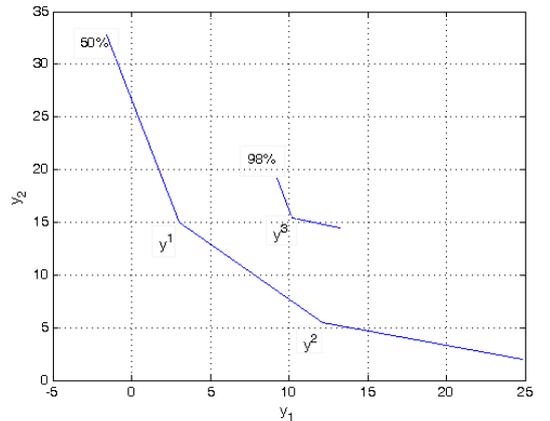


圖 2.3: 範例(2.6)目標空間

圖2.2中實線圍起來的中間區塊是可靠度50%的可行解空間，從圖2.2與圖2.3互相對照，可以知道在可靠度50%的Pareto集合中，非凌駕角解 y^1 對應到設計空間的效率角解 x^1 ，而此

效率角解active的拘束條件為 g_1 與 g_2 ，相同的可以觀察到另一個效率角解 x^2 ，active的拘束條件為 g_2 與 g_3 ；而在可靠度提升後的情形，會使得拘束條件往可行解空間的方向平移，使得可行解空間縮小，而考慮可靠度的新非凌駕角解 y^1 與 y^2 在圖2.3會越來越靠近直到碰在一起。

此情形可以觀察在可靠度98%的Pareto集合中非凌駕角解 y^3 ，其對應的效率角解 x^3 ，其active的拘束條件為平移後的 g_1' 與 g_3' ；原本 $g_1 = 0$ 與 $g_3 = 0$ 的交點在 $g_2 = 0$ 線後的非可行解空間，而在拘束條件平移時， $g_1' = 0$ 與 $g_3' = 0$ 的交點移動的比 $g_2' = 0$ 快，直到此焦點移動到 $g_2' = 0$ 線前的可行解空間裡，成為新的效率角解 x^3 ，其active的情況改變了，但是在哪種可靠度程度改變，以及如何在演算法中觀察到active的狀態，將在下面從最佳解的必要條件(KKT Conditions) [2]，來討論拘束條件active情形。

Karush-Kuhn-Tucker 必要條件(KKT)

對於一個單目標最佳化問題，其問題的一般形式如方程式(2.7)

$$\begin{aligned} \min_{\mathbf{x}} f(\mathbf{x}) \\ \text{s.t. } \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \end{aligned} \quad (2.7)$$

而當這個最佳化問題找到最佳解 x^* 的時候，這個最佳解會符合必要條件KKT [2]，分為三個部分描述如方程式(2.8)

$$g(x^*) \leq 0 \quad (2.8a)$$

$$\nabla f(x^*) + \lambda^T \nabla g(x^*) = 0^T, \quad \lambda \geq 0 \quad (2.8b)$$

$$\lambda^T g(x^*) = 0 \quad (2.8c)$$

第一部分描述的是最佳解必須在可行解空間裡面；第二部分是一階最佳解條件，當最佳解找到時，此最佳解的 $-\nabla f(x^*)$ 可以用所有active拘束條件的梯度向量做線性合成，而做線性組合時使用的係數 λ 稱之為Lagrange multiplier；第三部分則是為了只考慮active的拘束條件。若 $g(x^*)$ 值不為零則其不為active拘束條件，因此指定其 λ 值為零使它符合方程式(2.8)的第三部分，因此第三部分不是 λ 值為零就是 $g(x^*)$ 值等於零，KKT條件還有在幾何上的意義，下面將用一個簡單的範例來說明。

方程式(2.9)為文獻 [34]上提供的數學範例，而圖2.4說明了此問題的可行解空間，最佳解(2.1)及對應的函數梯度方向；首先此最佳解符合方程式(2.8)的第一部分，也就是必須

為可行解；再由圖上可以看出其active的拘束條件為 g_1 與 g_2 ，因此根據方程式(2.8)的第三部分 λ_3 與 λ_4 為零；此最佳解必須往 $\nabla f(x^*)$ 的反方向移動才能使目標函數值更好，但是若往此方向移動設計點的話，便會使得 g_1 與 g_2 的拘束條件值增大而違反拘束條件，因為此方向是 $\nabla g_1(x^*)$ 與 $\nabla g_2(x^*)$ 的線性組合，由此範例可以說明KKT條件為最佳解的必要條件。

$$\begin{aligned} \min_{\mathbf{x}} \quad & f_1(\mathbf{x}) = (x_1 - 3)^2 + (x_2 - 2)^2 & (2.9) \\ \text{s.t.} \quad & g_1 = x_1^2 + x_2^2 - 5 \leq 0 \\ & g_2 = x_1 + 2x_2 - 4 \leq 0 \\ & g_3 = -x_1 \leq 0 \\ & g_4 = -x_2 \leq 0 \end{aligned}$$

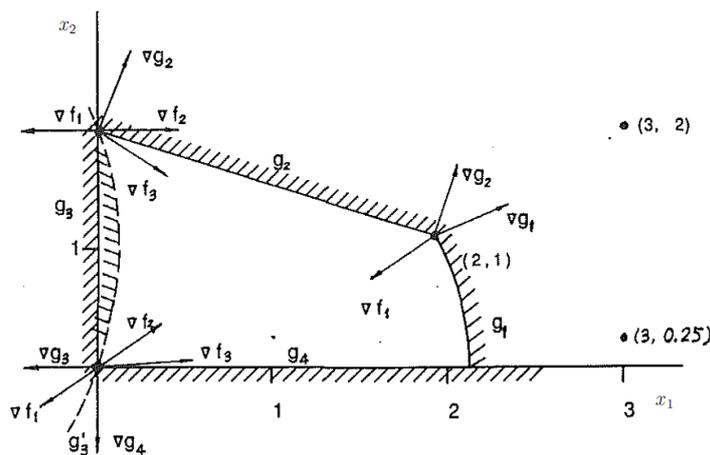


圖 2.4: 文獻[34]中描述KKT條件的幾何意義

線性單目標最佳化問題的KKT條件

在了解KKT條件的一般形式之後，在這一段落將介紹KKT條件在線性單目標最佳化的形式，一個典型的線性單目標最佳化問題如方程式(2.10)表示：

$$\begin{aligned} \min_{\mathbf{x}} \quad & f = \mathbf{c}^T \mathbf{x} & (2.10) \\ \text{s.t.} \quad & \mathbf{g} = \mathbf{a}^T \mathbf{x} - \mathbf{b} \leq \mathbf{0} \end{aligned}$$

把方程式(2.8)改寫成線性形式(2.11)

$$\mathbf{a}^T \mathbf{x} - \mathbf{b} \leq \mathbf{0} \quad (2.11a)$$

$$\mathbf{c}^T + \lambda^T \mathbf{a}^T = \mathbf{0}^T, \quad \lambda \geq 0 \quad (2.11b)$$

$$\lambda^T (\mathbf{a}^T \mathbf{x} - \mathbf{b}) = 0 \quad (2.11c)$$

在線性形式的KKT條件中，第一個部分確保最佳解在可行解空間中與第三部分只考慮active拘束條件的Lagrange multiplier都與方程式(2.8)相同；值得注意的第二部分，由方程式(2.11)可以觀察到，只要active的拘束條件相同，那矩陣 \mathbf{a} 便相同，計算出來的 λ^T 也會相同；相反的，只要active拘束條件改變，那麼 λ^T 值便會不同，因此比較考慮不同可靠度下的非凌駕角解的 λ^T 值，便是本篇論文演算法中判斷拘束條件active狀況的重要方法。

2.3 非凌駕角解的移動

在先前的文章討論了該如何考慮不確定因素，並且說明考慮不確定因素後會使線性拘束條件平移，平移的結果會使得非凌駕角解怎麼移動?是這段討論的主題。

假設有個效率角解 \mathbf{x}^p 與其對應的非凌駕角解 $\mathbf{y}^p = \mathbf{c}^T \mathbf{x}^p$ 存在，而其active的線性拘束條件 $\mathbf{a}^T \mathbf{x}^p - \mathbf{b} \leq 0$ 改寫成 $\mathbf{a}^T \mathbf{x}^p = \mathbf{b}$ ，效率角解 \mathbf{x}^p 可以視為其交點；在考慮可靠度下，線性拘束條件會依方程式(2.5)平移成新的拘束條件，如方程式(2.12)。

$$\mathbf{a}^T \mathbf{x}^p = \mathbf{b} \quad \longrightarrow \quad \mathbf{a}^T \mathbf{x}^p = \mathbf{b} + \sigma_A \Phi^{-1}(1 - P_f) \quad (2.12)$$

\mathbf{x}^p 是這些active拘束條件極限狀態的交點，因此平移後的 \mathbf{x}^p 可由方程式(2.12)改寫成下式(2.13)：

$$\mathbf{x}^p = \mathbf{a}^{T-1} \mathbf{b} + \mathbf{a}^{T-1} \sigma_a \Phi^{-1}(1 - P_f) \quad (2.13)$$

方程式(2.13)與平移前的 \mathbf{x}^* 相比較可以看出效率角解移動了 $\mathbf{a}^{T-1} \sigma_a \Phi^{-1}(1 - P_f)$ ，如果透過矩陣 \mathbf{c}^T 可以直接映射到目標空間的一個向量 \mathbf{z}

$$\mathbf{z} = \mathbf{c}^T \mathbf{a}^{T-1} \sigma_a \Phi^{-1}(1 - P_f) \quad (2.14)$$

假如角解的拘束條件在改變可靠度時沒有跟著改變，那麼目標空間的角點就如方程式(2.14)改變。

2.4 線性Pareto集合邊界的移動

在文獻 [33]中有討論到線性問題的Pareto集合在考慮不確定因素下會有平移的現象，而本論文在這裡將以 [19]的基礎上來討論此現象。

在方程式(1.17)中定義了鄰近兩個角解在可行解空間的某一邊界，而角解 x 在考慮可靠度時，不改變active拘束條件狀態下，其角解會依上面方程式(2.13)移到新的角解 x^p 上，因此移動之後的邊界可以表示成方程式(2.15)。

$$E^{ip} = \{(1 - \alpha)\mathbf{x}^p + \alpha\bar{\mathbf{x}}^p : 0 \leq \alpha \leq 1\} \quad (2.15)$$

而因為active狀態沒有改變，所以兩個考慮不確定因素的非凌駕角解 $\bar{\mathbf{y}}^p$ 與 \mathbf{y}^p ，都是原來的非凌駕角解經過方程式(2.14)平移的結果，如方程式(2.16)與2.17。

$$\mathbf{y}^p = \mathbf{y} + \mathbf{c}^T \mathbf{a}^{T-1} \sigma_a \Phi^{-1}(1 - P_f) \quad (2.16)$$

$$\bar{\mathbf{y}}^p = \bar{\mathbf{y}} + \mathbf{c}^T \bar{\mathbf{a}}^{T-1} \sigma_a \Phi^{-1}(1 - P_f) \quad (2.17)$$

根據章節1.2.2與文獻 [19]的說明，只要 \mathbf{r}^i 是矩陣 R 架構中的其中一行向量，方程式(1.19)就是Pareto集合的其中一條邊界；而在不改變active狀態的前提下，矩陣 R 就不會改變，因此方程式(2.18)依然是Pareto集合的其中一條邊界，而且 $\mathbf{r}^{ip} = \mathbf{r}^i$ ，因此兩條Pareto集合在同一個超平面上，可以使用線段平移的方式來預測Pareto集合 [33]。

$$\bar{\mathbf{y}}^p = \mathbf{y}^* - \theta_i^p \mathbf{r}^{ip} \quad (2.18)$$

2.5 改變active拘束條件的可靠度尋找

在上一章我們可以產生出線性多目標最佳化Pareto集合的角點，再經由方程式(2.14)可以知道角點的移動，至於移動距離最大為多少時拘束條件active情形還未改變？我們透過最佳化搜尋的方式求解。

在KKT條件之中，每一個角解都有他專屬的Lagrange multiplier，只要角解在考慮不同可靠度程度下active情形沒有改變，那麼它的Lagrange multiplier也不會改變，換句話說，我們只要觀察在Lagrange multiplier不改變的情況下，能預測的最大移動距離為多少？即能預測的最大的可靠度是多少，數學式子如下列最佳化問題：

$$\begin{aligned} \min_{P_f} \quad & -|z(P_f)| \\ \text{s.t.} \quad & \lambda(P_f) - \lambda' = 0 \end{aligned} \quad (2.19)$$

如方程式(2.14)所表示，方程式(2.19)中的目標函數 $-|z(P_f)|$ 是個單調遞增的函數，因此可以簡化為方程式(2.20)：

$$\begin{aligned} \min_{P_f} \quad & P_f \\ \text{s.t.} \quad & \lambda(P_f) - \lambda' = 0 \end{aligned} \quad (2.20)$$

問題(2.20)中的 $\lambda(P_f)$ 是每一個角點在改變 P_f 的情況下用拘束法計算出來的Lagrange multiplier，計算方式如方程式(2.21)，而 λ' 則是一開始的參考Lagrange multiplier：

$$\begin{aligned} \min_{\mathbf{x}} \quad & f_l(\mathbf{x}) \\ \lambda(P_f) : \quad & \text{s.t.} \quad \mathbf{g}(\mathbf{x}) \leq 0 \\ & f_j(\mathbf{x}) \leq \varepsilon_j + z_j(P_f), \quad \forall j = 1, \dots, k, \quad j \neq l \end{aligned} \quad (2.21)$$

如此一來每一個角點都計算出在active情形不改變下能進行多少可靠度的預測，再選擇較小可靠度的進行新的計算，直到我們想要觀察的可靠度為止。附帶一提，因為此最佳化問題(2.20)的拘束條件並不具有梯度資訊，因此沒法使用梯度方法當基礎的演算法來求解，必須使用類似DIRECT [35]的全域演算法才能求解。

至此，我們可有系統的找到改變active拘束條件的臨界可靠度，因此任兩個active拘束條件不改變的Pareto集合均可用文獻 [33]的方法進行預測，如此一來我們就可以解決線性多目標最佳化問題。

2.6 考慮不確定因素下之線性多目標最佳化的Pareto集合演算法

根據上面的介紹，在本篇論文提出一套演算法如圖2.5，讓決策者可以得到從起始可靠度到最終可靠度(最小破壞機率)中任一可靠度的Pareto集合資訊。首先以起始可靠度計算出一組Pareto集合後，再計算此集合上每一個非凌駕角解在考慮可靠度的情況下可能的移動向量，並計算可靠度提高多少的時候任一個非凌駕角解的拘束條件active情況改變，接著以此可靠度計算出新的Pareto集合，一直持續計算直至得到最終可靠度的Pareto集合資訊。下面是此演算法的細節。

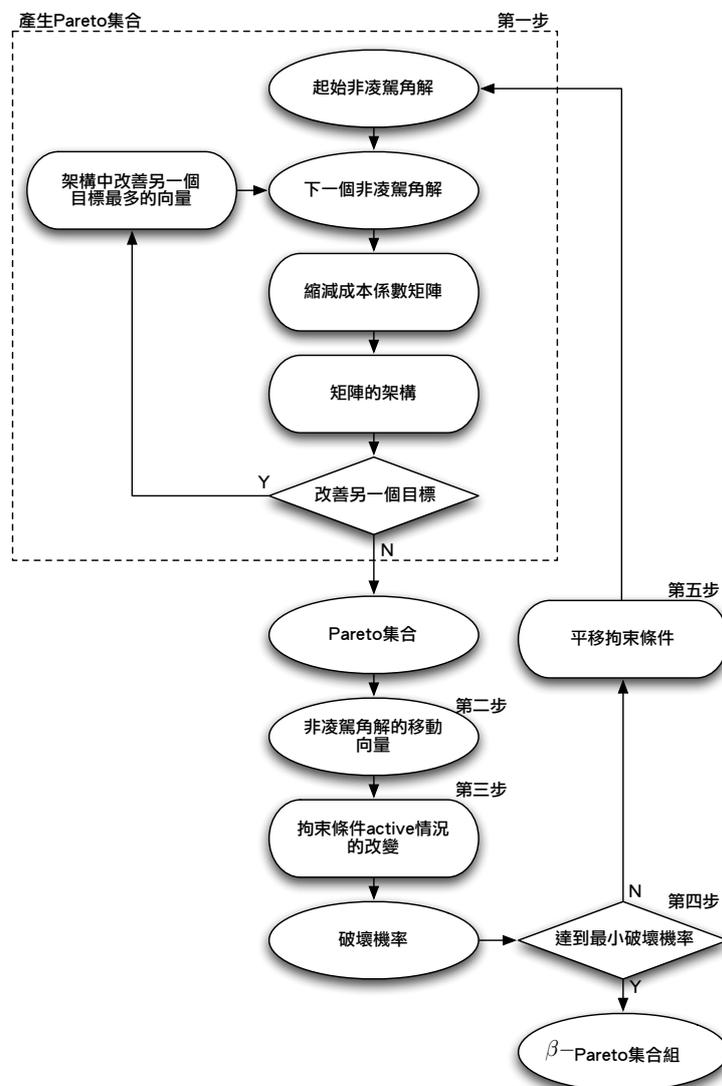


圖 2.5: 考慮不確定因素下之線性多目標最佳化Pareto集合演算法

演算法細節

第零步 在一開始讓決策者決定一個可靠度範圍。基本上，低於50%可靠度的平均設計點已經違反了拘束條件，因此不需考慮；起始可靠度通常都是設定50%以上，而最終可靠度必須比起始可靠度來的高。決定可靠度範圍後以起始可靠度進入第一步。

第一步 根據此可靠度讓原本不考慮不確定因素的拘束條件做平移的動作，產生一組新的拘束條件

$$\mathbf{a}^T \boldsymbol{\mu}_X - \sigma_a \Phi^{-1}(1 - P_f) - \mathbf{b} \leq 0$$

$$\sigma_a = \sqrt{\sum_{i=1}^n (\mathbf{a}_i)^2 \sigma_{X_i}^2}$$

接著使用在目標空間計算線性Pareto集合演算法來計算此可靠度的Pareto集合，進入第二步。

第二步 計算在Pareto集合上每一個非凌駕角解在不確定因素下的變化方向 z

$$\mathbf{z} = \mathbf{c}^T \mathbf{a}^{T-1} \sigma_a \Phi^{-1}(1 - P_f)$$

並找出每一個非凌駕角解在拘束法時的Lagrange multiplier， λ' ，來代表每一個非凌駕角解的active情形，進入第三步。

$$\mathbf{c}^T + \lambda'^T \mathbf{a}^T = \mathbf{0}^T$$

第三步 爲了知道每一個非凌駕解沿著第二步所計算出來的變化方向移動距離最遠多少，因此計算一個最佳化問題如下，來得知某一個非凌駕角解的拘束條件active情形改變時的可靠度：

$$\begin{aligned} \min_{P_f} \quad & P_f \\ \text{s.t.} \quad & \lambda(P_f) - \lambda' = 0 \end{aligned}$$

其中 $\lambda(P_f)$ 則是使用拘束法計算單目標最佳化問題而得到相對應的Lagrange multiplier，舉雙目標爲例：

$$\begin{aligned} \min_{\mathbf{x}} \quad & f_2(\mathbf{x}) \\ \lambda(P_f) : \quad & \text{s.t.} \quad \mathbf{g}(\mathbf{x}) \leq 0 \\ & f_1(\mathbf{x}) \leq \varepsilon_1 + z_1(P_f) \end{aligned}$$

而每一個非凌駕角解都可以計算出在不改變拘束條件active情況下能提高的可靠度，整理所有非凌駕角解相對應的最大可靠度中，挑選最小的一個可靠度進入第四步。

第四步 檢查 $\lambda(P_f)$ 與最終可靠度(最小破壞機率)計算出來的Lagrange multiplier是否相同，相同就代表現在此可靠度的active情形與最終可靠度的active情形已經相同，不需再繼續計算下去，結束演算法；如果其Lagrange multiplier不同，則進入第五步。

第五步 此可靠度的某個非凌駕角解active情形剛好發生改變，以此可靠度平移拘束條件後進入第一步。

在演算法結束之後我們得到一張在active情形一有改變就繪製Pareto集合的圖，在鄰近兩個可靠度的Pareto集合間並沒有active情形的變化，因此可以使用線性預測的方式來取得Pareto集合 [33]。



2.7 範例

延續 [19]的範例，在我們考慮不確定因素之後，使用FOSM的概念平移拘束條件，從方程式(1.24)改成方程式(2.22)：

$$\begin{aligned}
 \min_{\mathbf{x}} \quad & \mathbf{c}\mathbf{x} = \begin{pmatrix} x_1 + \frac{1}{9}x_3 \\ x_2 + \frac{1}{9}x_3 \end{pmatrix} & (2.22) \\
 \text{s.t.} \quad & g_1 : 9x_1 + 9x_2 + 2x_3 - 81 - \sigma_1\Phi^{-1}(1 - P_f) \leq 0 \\
 & g_2 : 8x_1 + x_2 + 8x_3 - 72 - \sigma_2\Phi^{-1}(1 - P_f) \leq 0 \\
 & g_3 : x_1 + 8x_2 + 8x_3 - 72 - \sigma_3\Phi^{-1}(1 - P_f) \leq 0 \\
 & g_4 : -7x_1 - x_2 - x_3 + 9 - \sigma_4\Phi^{-1}(1 - P_f) \leq 0 \\
 & g_5 : -x_1 - 7x_2 - x_3 + 9 - \sigma_5\Phi^{-1}(1 - P_f) \leq 0 \\
 & g_6 : -x_1 - x_2 - 7x_3 + 9 - \sigma_6\Phi^{-1}(1 - P_f) \leq 0 \\
 & g_7 : x_1 - 8 - \sigma_7\Phi^{-1}(1 - P_f) \leq 0 \\
 & g_8 : x_2 - 8 - \sigma_8\Phi^{-1}(1 - P_f) \leq 0 \\
 & x_i \geq 0, \quad \sigma_{x_i} = 1, \quad i = 1, 2, 3, \quad \sigma_j = \sqrt{\sum_{i=1}^3 \left(\frac{\partial g_j}{\partial x_i}\right)^2 \sigma_{x_i}^2}, \quad j = 1, 2, \dots, 8
 \end{aligned}$$

可靠度 $P_r = (1 - P_f)$ ，在 $P_r = 50\%$ 時因為 $\Phi^{-1}(0.5) = 0$ 可以看出此問題與方程式(1.24)完全一樣，因此下列討論皆會以 $P_r = 50\%$ 的描述來代替*Deterministic*；可靠度提高會使得拘束條件平移，造成可行解空間縮小，進一步的使Pareto集合平移。使用本章節介紹的演算法可以繪製出如圖2.6的Pareto集合組，圖2.6的x軸是第一個目標函數值，y軸是第二個目標函數值，首先在圖2.6中 $P_r = (50\%)$ 的每個非凌駕角解都有此角解在考慮不確定因素下的變化向量，非凌駕角解將沿著向量移動直到拘束條件的active拘束條件改變。根據本章節提出的演算法計算在 $P_r = (69.3\%)$ 時有某非凌駕角解的active拘束條件改變，如圖2.6上表示， $P_r = (69.3\%)$ 的非凌駕角解很理想的出現在向量的交點上；另外圖2.6上說明演算法推算在可靠度 $P_r = (98.3\%)$ Pareto集合接近空集合，說明此時的可行解空間縮小至趨近於消失。

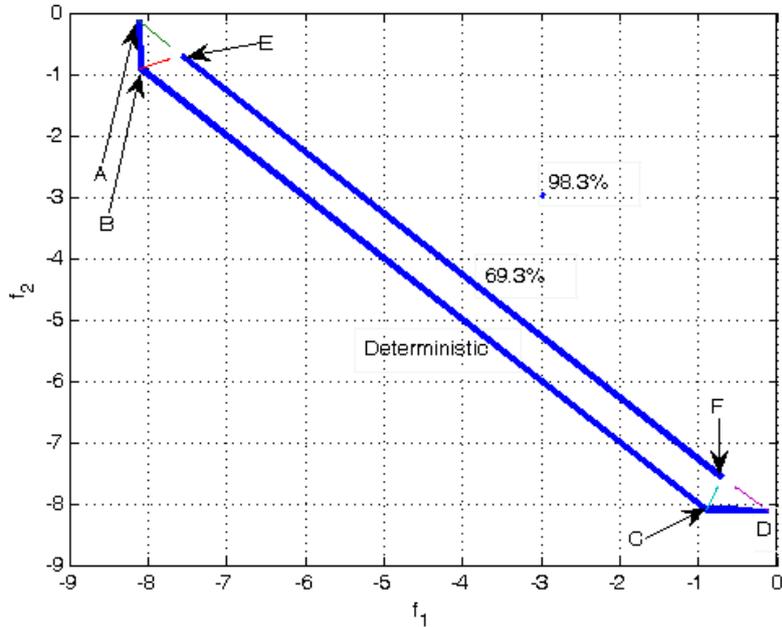


圖 2.6: 考慮不確定因素下的線性Pareto集合組

表 2.1: 非凌駕角解的相關資訊

非凌駕角解	A	B	C	D	E	F
$\mathbf{ca}^{-1}\sigma_a$ 在 f_1 的分量	1.064	1.042	0.390	-1.181	2.834	-1.403
$\mathbf{ca}^{-1}\sigma_a$ 在 f_2 的分量	-1.181	0.390	1.042	1.064	-1.403	2.834
λ_{g_1}	0.000	0.111	0.111	0.000	0.111	0.111
λ_{g_2}	1.000	0.000	0.000	0.000	0.000	0.000
λ_{g_3}	0.000	0.000	0.000	0.014	0.000	0.000
λ_{g_4}	0.000	0.000	0.000	0.002	0.000	0.000
λ_{g_5}	0.000	0.000	0.000	0.000	0.000	0.000
λ_{g_6}	0.000	0.000	0.000	0.000	0.000	0.000
λ_{g_7}	63.000	0.000	0.000	0.000	0.000	0.000
λ_{g_8}	0.000	0.000	0.000	0.889	0.000	0.000
λ_{f_1}	71.000	1.000	1.000	0.000	1.000	1.000
最小適用可靠度	50.0%	50.0%	50.0%	50.0%	69.3%	69.3%
最大適用可靠度	69.3%	69.3%	69.3%	69.3%	98.3%	98.3%

表2.1是演算法在計算時必要的計算結果，表中提供了圖2.6中每一個非凌駕角解的相關資訊，包括移動向量，以及每個拘束條件的Lagrange multiplier量值來代表active情況跟適用的可靠度範圍，在這可靠度範圍內都是一樣的Lagrange multiplier，代表此非凌駕角解active的拘束條件都相同。

第三章 不確定因素下非線性問題的Pareto集合的改變

前面討論了在目標空間計算Pareto集合，以及在目標空間計算Pareto集合的方法為基礎，延伸至考慮不確定因素的Pareto集合計算，都必須在目標函數與拘束條件都是線性的前提下才能使用，因此現在我們要把上一段處理線性多目標最佳化的技術衍伸到非線性問題上，就必須要找一個適當的線性問題來代替非線性問題處理，問題就出在於什麼是“適當”的線性問題，在這一章將討論對一個非線性多目標最佳化問題線性化的方法。

須注意一點，下面的段落為了討論對目標空間的Pareto集合線性化，因此符號上x軸是自變數 f 而y軸是應變數 $p(f)$ ，與平常符號使用意義不同，在此強調。

3.1 Pareto集合的線性化

在使用逼近法來解決多目標最佳化的回顧文獻 [36]中，文獻 [37]的方法是使用區塊三明治夾擠法，因為它並不需要計算函數的梯度資訊，因此計算速度快，又不失準確性，這篇論文將使用此方法。

3.1.1 區塊三明治夾擠法

區塊三明治夾擠法是使用一個上界與下界的線性函數來夾擠原函數，直到上界與下界的誤差值小於我們所希望的誤差值為止。假如我們將Pareto集合視為一個凸函數 $p(f)$ 來處理，也就是將拘束法中改變不同的 ϵ 當作自變數 f ，而計算出非凌駕解中另一個目標函數的值為應變數 $p(f)$ ，即可用此法來產生出一組上界與下界的線性函數，如圖3.1。拘束法只有在適當的區間內可以使用，而我們假設在一個區間 $[f, \bar{f}]$ 之間有 b 個切點可以將Pareto集合切成 $(b + 1)$ 個區塊來討論， $S_b = \{f_i, i = 0, 1, 2, \dots, b\}$ ($f_0 = f$ ，且此處的 f_i 與第 i 個目標函數沒有關係)，則上界函數 $u_i(f)$ 可以寫成方程式(3.1)並如圖3.2所表示

$$u(f) : u_i(f) = p(f_i) + \frac{p(f_{i+1}) - p(f_i)}{f_{i+1} - f_i} (f - f_i), \quad (3.1)$$
$$f \in [f_i, f_{i+1}]$$

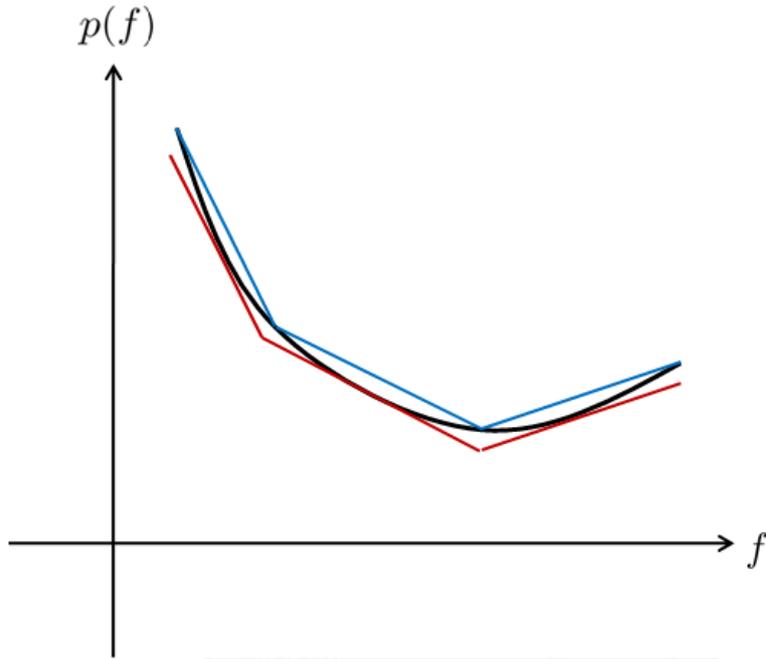


圖 3.1: 區間三明治夾擠法示意圖

而把上界函數往下移動，直到此上界函數與原非線性函數切於一點 f'_i ，找 f'_i 的過程為一個單目標最佳化問題：

$$\begin{aligned} \min_f \quad & p(f) - e_i f & (3.2) \\ \text{s.t.} \quad & f_i \leq f \leq f_{i+1} & \text{where } e_i = \frac{p(f_{i+1}) - p(f_i)}{f_{i+1} - f_i} \end{aligned}$$

假設 f'_i 是使問題(3.2)有最佳解的最佳點如圖3.2，下界函數 $l_i(f)$ 可以寫成方程式(3.3)

$$l(f) : l_i(f) = p(f'_i) + \frac{p(f_{i+1}) - p(f_i)}{f_{i+1} - f_i} (f - f'_i), \quad (3.3)$$

因為下界函數 l_i 的範圍必須考慮後一段 l_{i+1} 與前一段 l_{i-1} 才能得到，如圖3.3所示，分別計算 l_i 與 l_{i+1} 的交點 f'_i 來得到 l_i 的上界如方程式(3.4)，以及 l_i 與 l_{i-1} 的交點 f'_{i+1} 來得到 l_i 的下界如方程式(3.5)

$$\begin{aligned} f'_{i+1} = & \left(\frac{p(f_{i+1}) - p(f_i)}{f_{i+1} - f_i} - \frac{p(f_{i+2}) - p(f_{i+1})}{f_{i+2} - f_{i+1}} \right)^{-1} \\ & \left(\frac{p(f_{i+1}) - p(f_i)}{f_{i+1} - f_i} f'_i - \frac{p(f_{i+2}) - p(f_{i+1})}{f_{i+2} - f_{i+1}} f'_{i+1} - p(f'_i) + p(f'_{i+1}) \right) \end{aligned} \quad (3.4)$$

$$\begin{aligned} f'_i = & \left(\frac{p(f_i) - p(f_{i-1})}{f_i - f_{i-1}} - \frac{p(f_{i+1}) - p(f_i)}{f_{i+1} - f_i} \right)^{-1} \\ & \left(\frac{p(f_i) - p(f_{i-1})}{f_i - f_{i-1}} f'_{i-1} - \frac{p(f_{i+1}) - p(f_i)}{f_{i+1} - f_i} f'_i - p(f'_{i-1}) + p(f'_i) \right) \end{aligned} \quad (3.5)$$

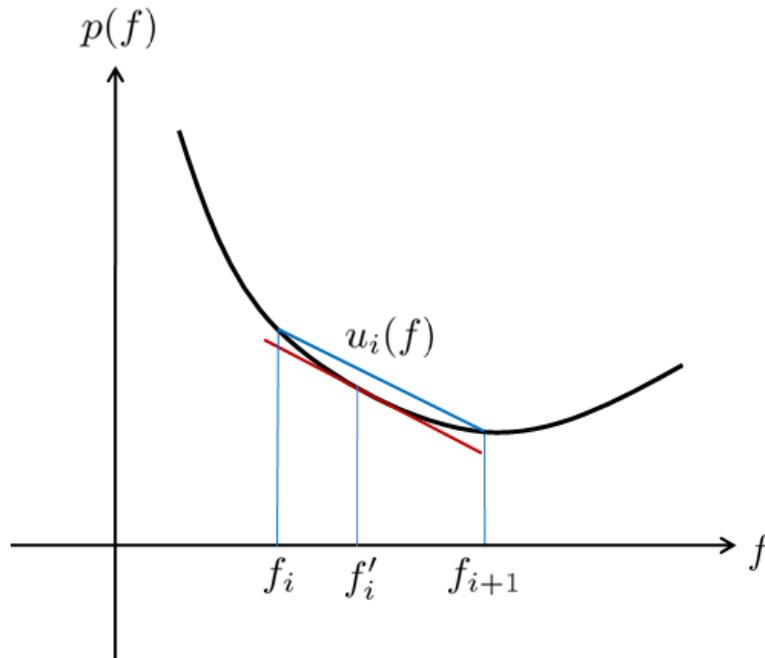


圖 3.2: 由上界函數 $u_i(f)$ 尋找下界函數 $l_i(f)$ 示意圖

如此一來只要有適當的切點，我們就可以使用上界與下界線性函數來逼近原函數： $l(f) \leq p(f) \leq u(f)$, $f \in [\underline{f}, \bar{f}]$

對第 i 段的上界與下界函數而言，兩個函數互相平行，文獻 [37]中定義上界與下界之間間距定義為此逼近法的誤差，假如誤差大於使用者所要求的值，則把 f'_i 做為新的切點加入原來的切點中 $S_b = \{f_0, f_1, f_2, \dots, f_b\} \rightarrow \{f_0, f_1, f_2, \dots, f_b, f'_i\}$ ，把原來的函數 $p(f)$ 用更多的切點分更多段去討論，用更多線段去逼近。由此法產生出的線性函數有個特性，曲率較高的部分將會用較多段線性函數去逼近，曲率較低的部分相對的會用較少段線性函數逼近。

3.1.2 非線性拘束條件的線性化

雖然產生出一組在目標空間的線性函數，但由於考慮不確定因素必須使用設計空間的拘束條件作平移的計算，設計空間須如何設計才能達到我們所算出的線性函數？剛剛提到三明治夾擠法會有個特性是曲率較高的部分會有較多的線性函數去逼近，而每一個線性函數可以視為某個在曲線上的點作切線，而這些點對曲線來說是較為重要的“特徵點”，每一個特徵點我們都可以找到在設計空間中相對應的點位置，接著我們只要在設計空間中這些點上對active的非線性拘束條件作一階泰勒展開就可以獲得一組線性拘束條件，接著再使用文獻 [19]的方法來計算相對應的Pareto集合。值得一提，用此方法計算出來的Pareto集合與一開始在設計空

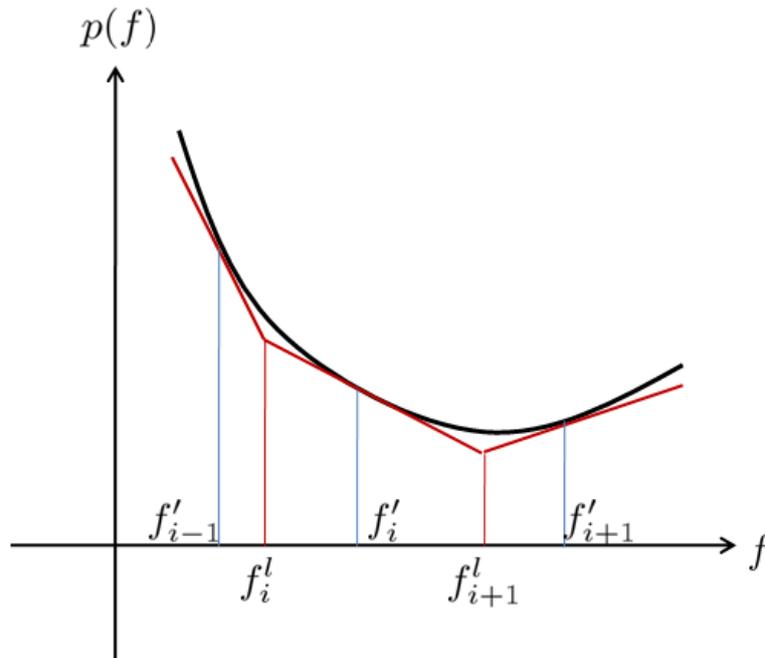


圖 3.3: 下界函數 $l_i(f)$ 的適用範圍

間找到的下界線性函數有些許誤差，不過這誤差會隨著決策者希望使用較多線性函數逼近而減小。

如此一來我們已經找到適當的線性問題來逼近原本的非線性問題，處理線性問題比處理非線性問題來的簡單而且計算速度快。接著就使用此線性問題來預測非線性問題Pareto集合的移動情形。而線性最佳化問題Pareto集合的移動與非線性最佳化問題Pareto集合的移動一定會有誤差，在這篇論文所使用計算誤差的方式為計算單目標最佳解，非線性問題與線性問題的差距，因為這些值是在運算時一定會計算的，假如兩者的值相差大到決策者不容許，那麼表示一開始的線性最佳化問題不夠適當，因為理論上線性函數的個數趨近於無限多的時候，其線性函數的特性就可以代表非線性函數的特性，因此試著用較多線性函數來逼近原問題可以縮小這個誤差。

3.2 不確定因素下之非線性多目標最佳化的Pareto集合演算法

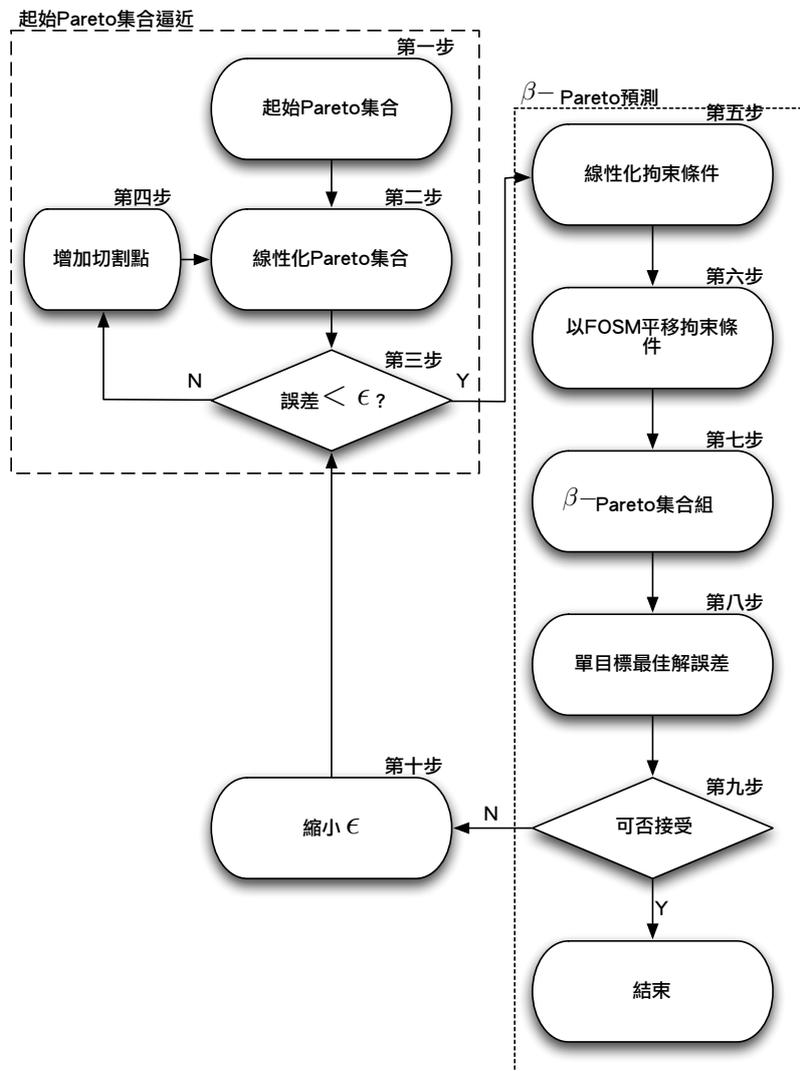


圖 3.4: 非線性Pareto集合演算流程

我們在本篇論文針對非線性問題提出的Pareto集合演算法如圖3.4，此演算法可以概分為兩部分；第五步之前是第一部分：起始Pareto集合逼近，是為了找到一個線性的問題來逼近非線性問題，找到之後才能夠以此線性問題進入第二部分進行： β -Pareto預測。而第一部分的逼近方法採用適當的線段數量來近似原本的非線性Pareto集合，直到逼近的誤差小於決策者可以接受的範圍後，再將逼近時產生的“特徵點”進入第二部分線性化非線性拘束條件，產生一組適當的線性拘束條件。在第二部分則使用前面章節提到線性問題在不確定因素下的Pareto集合計算技巧，得到 β -Pareto集合組。下面介紹此演算法的細節：

演算法細節

第一步 把不考慮可靠度的非線性Pareto集合當成起始Pareto集合開始演算法，假如取第 j 個目標函數值當因變數，另一個函數值便是應變數，先計算 f_j 的最佳解 $f_j(\mathbf{x}_j^*)$ 當作因變數的下界 \underline{f}

$$\begin{aligned} \min_{\mathbf{x}} \quad & f_j(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{x}) \leq 0 \end{aligned} \quad (3.6)$$

接下來計算另一個目標函數的最佳解 $f_l(\mathbf{x}_l^*)$ 後，將此最佳解帶入目標函數 $f_j(\mathbf{x}_l^*)$ 做為因變數的上界 \bar{f} ，如此一來因變數的上界與下界計算完成後，以 $\underline{f} \leq \varepsilon_j \leq \bar{f}$ 得到Pareto集合函數 $p(f)$ ，並以上界與下界做為起始切點，進入第二步。

$$p(f) = p(\varepsilon_j) : \left[\begin{array}{l} \min_{\mathbf{x}} \quad f_l(\mathbf{x}) \\ \text{s.t.} \quad \mathbf{g}(\mathbf{x}) \leq 0 \\ f_j(\mathbf{x}) \leq \varepsilon_j, \quad \forall j = 1, 2, \quad j \neq l \end{array} \right] \quad (3.7)$$

第二步 如方程式(3.1)計算上界函數，方程式(3.3)計算下界函數之後，計算上下界函數之間的誤差，進入第三步。

第三步 檢查由上一步得到的最大誤差是否比決策者可接受的誤差來的小，如果是，則第一部分演算完成，進入第五步；如果不能接受則進入第四步。

第四步 以第二步計算下界函數時的 f' 當作新增的切點，增加對起始Pareto集合的切割點來使用較多的線段逼近，根據 [37]提出的改良三明治夾擠法，會在Pareto集合曲率較大的地方有較多的切割點。進入第二步。

第五步 在目標空間對Pareto集合找到適當的線性化模式之後，要在設計空間找一組相對應的拘束條件，即把切割點當作“特徵點”並在設計空間把相對應的解找出來，並且對此特徵點上active的拘束條件做一階泰勒函數展開的線性化，得到一組拘束條件進入第六步。

第六步 以決策者決定的起始可靠度平移拘束條件後進入第七步。

第七步 根據如圖2.5所描述的線性問題Pareto集合演算法來計算繪製從起始可靠度到最終可靠度範圍內的Pareto集合組，進入第八步。

第八步 線性Pareto集合的平移與非線性Pareto集合的移動間一定有誤差存在，此誤差為非線性程度造成。這裡檢查兩者誤差的方式乃使用檢查兩問題中同一個單目標最佳點在目標空間的連線距離差距。計算完進入第九步。

第九步 檢查第八步所計算的誤差是否可以被決策者所接受，如果可以接受則演算法結束；如果不能接受則進入第十步。

第十步 理論上使用無限多個線段來逼近非線性函數時，就能有此非線性函數的特性，因此這裡縮小第三步裡決策者可接受的誤差 ϵ 使之成為原來的一半並進入第一部分重新開始演算法。

在此演算法計算結束之後，可以得到非線性問題的 β -Pareto集合組，鄰近兩個可靠度的Pareto集合間可以使用平移的方式預測之間可靠度的Pareto集合情形。



3.3 範例

這裡使用一個簡單的非線性多目標最佳化問題來示範。

$$\begin{aligned}
 \min \quad & \mathbf{c}\mathbf{x} = \begin{pmatrix} x_2 + x_3 \\ -x_1 - x_3 \end{pmatrix} & (3.8) \\
 \text{s.t.} \quad & x_1^2 + 0.1x_2^2 - x_3 - 50 - \sigma_1\Phi^{-1}(1 - P_f) \leq 0 \\
 & 0.1x_1^2 + 0.4x_2^2 + x_3 - \sigma_2\Phi^{-1}(1 - P_f) \leq 0 \\
 & \sigma_{x_i} = 1, \quad i = 1, 2, 3, \quad \sigma_j = \sqrt{\sum_{i=1}^3 \left(\frac{\partial g_j}{\partial x_i}\right)^2 \sigma_{x_i}^2}, \quad j = 1, 2
 \end{aligned}$$

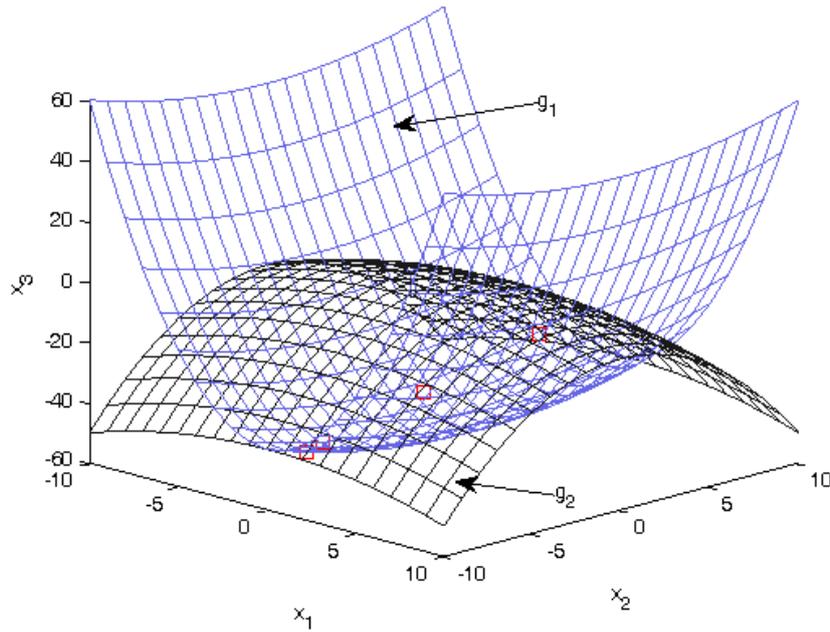


圖 3.5: 非線性問題的設計空間

從圖3.5可以看出此問題的拘束條件切割設計空間成一個凸集合的可行解空間，由於目標函數是線性函數，因此Pareto集合也是凸函數。此範例的計算根據演算法必須先對不考慮可靠度的Pareto集合線性化，如圖3.7，得到此Pareto集合的特徵點之後尋找其在設計空間相對應的位置如圖3.5，圖上的方框則是在進行演算法第五步時需要的特徵點，在這些特徵點上對active的拘束條件做如圖3.8的線性展開，對所有的特徵點都進行完此線性化的動作後而得到一組線性的拘束條件，接著計算此線性多目標最佳化的Pareto集合如圖3.7，再進行第二部分的演算法計算從可靠度50%到可靠度99.5%間的Pareto集合，得到如圖3.6的 β -Pareto集

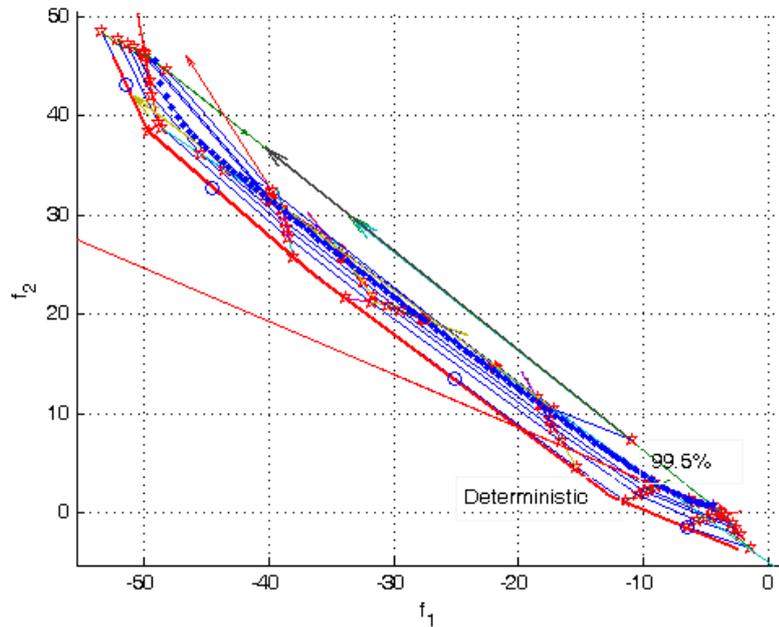


圖 3.6: 非線性問題的 β -Pareto集合

合組。演算法的運算到這裡停止，而圖3.6中鄰近兩條Pareto集合間都可使用平移的方式預測 [33]。

圖3.6的x軸是第一個目標函數值，y軸是第二個目標函數值，此圖描繪出問題(3.8)從可靠度50%到99.5%的Pareto集合情形，並且在可靠度99.5%用拘束法檢驗，而圖上的向量是非凌駕角解的移動向量，只要向量改變就代表非凌駕角解的active情況改變；圖3.6中注意到在可靠度99.5%的地方有些許預測的Pareto集合邊界與拘束法得到的Pareto集合相差甚大，在仔細觀察後，因為演算法第五步所計算出來的線性多目標問題有可能不是良好界定(Well Bounded)，而導致計算線性最佳化問題的Pareto集合出現問題而無法繼續，此時作者加入在可靠度50%的Pareto集合中非active的線性拘束條件，來使此線性多目標問題成為良好界定的問題，但在提高可靠度時此“不該影響Pareto集合”的拘束條件卻加入某些非凌駕角解的active集合中，是此方法的缺點。

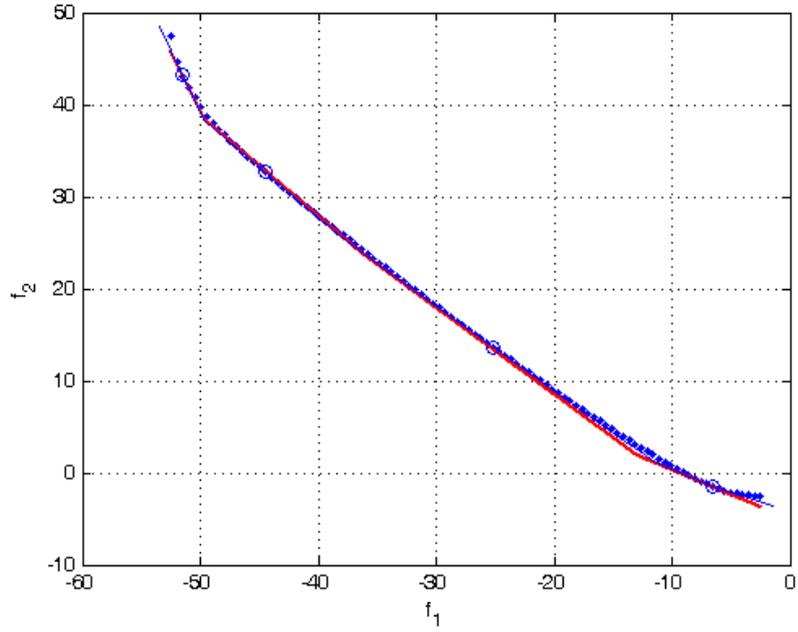


圖 3.7: 線性化不考慮可靠度的Pareto集合

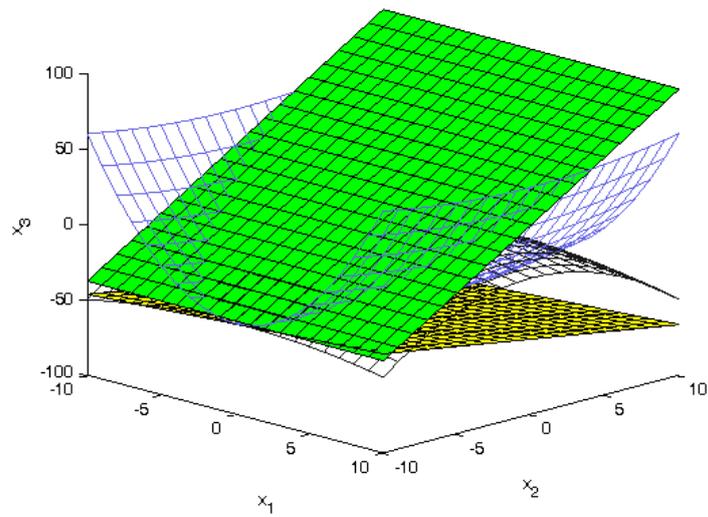


圖 3.8: 對特徵點上active的拘束條件線性化示意圖

第四章 結果與展望

4.1 結論

在本篇論文中使用文獻 [19]中計算線性多目標最佳化Pareto集合的演算法，並且搭配將二維非線性凸函數線性化的文獻 [37]提出一套演算法，來逼近非線性雙目標最佳化的Pareto集合，並且可以提供一個可靠度範圍內的Pareto集合變化情形，解決文獻 [33]中無法處理active拘束條件改變的狀況。

在章節2.3中討論到計算每個角點的active情形不改變下能提高的可靠度為多少時，必須採用類似DIRECT的全域演算法，但此演算法演算停止的條件是疊代次數到達使用者指定的次數，或者是函數的計算次數到達要求，因此有可能在到達要求次數之後還未收斂到正確的 P_F ，關於此問題作者建議可以提高指定次數來改善情況。

處理非線性問題的部分，在計算出一組適當的線性拘束條件時，有可能因為線性多目標問題不是良好界定(Well Bounded)而導致計算線性最佳化問題的Pareto集合出現問題而無法繼續，這裡作者建議可以自行加入“不影響”Pareto集合的線性拘束條件來解決此問題，也就是新加入的線性拘束條件不能是Pareto集合中active的拘束條件。

4.2 研究貢獻

相對於本論文中討論到線性Pareto集合在考慮不確定因素下會有平移的特性，在前人的文章 [33]中也有討論此特性，只是本論文是使用文獻 [19]提出在目標空間計算Pareto集合的方法來對此特性做討論；另外討論到如何由數值得知非凌駕角解的拘束條件active情形，進而可以計算在哪個可靠度下“剛好”會使Pareto集合中某個非凌駕角解的拘束條件active情形改變，只要在決策者想要了解的可靠度範圍中將非凌駕角解active情況改變時可靠度相對應的Pareto集合繪製出來，那鄰近兩條Pareto集合之間便可以平移的方式來進行預測；再來本論文使用文獻 [37]提出的方法將預測的技巧從線性問題延伸至非線性問題，研究貢獻條列如下：

- 證明線性多目標最佳化的Pareto集合在考慮不確定因素下會有平移的特性。

- 在線性多目標最佳化的問題，可以列出一段可靠度範圍內的所有Pareto集合。
- 將某一可靠度範圍內所有Pareto集合的計算技巧延伸至非線性問題。

4.3 未來工作與展望

在可靠度提高，也就是 P_f 減小的情況，會造成線性拘束條件的平移，連帶使得設計空間縮小，這種狀況不可避免的會遇到拘束條件active情形的改變，在改變的那一個 P_f 會使得部份的角解有退化(degeneracy)的情形，也就是在某個角解上active的拘束條件個數比設計空間的維度高；本論文遇到此情形處理的方式是採用再減小 P_f 一個很小的值來避開處理此問題，當然幸運的是有作者討論當退化時的情形該如何處理 [23]，本論文的未來可以此做修正。

另外如果想要增加目標函數的維度，也就是處理不只兩個目標函數的問題時，作者建議採用其他種線性化Pareto集合的方法來產生適當的線性拘束條件，因為本論文所採用的區塊三明治夾擠法 [37]僅適合使用在二維的問題。因此只要找到適當的線性拘束條件，皆可以使用本論文中計算線性問題的 β -Pareto集合組的方法逼近非線性問題。

因為實際工程問題常常設計空間不是凸集合，而且目標函數也有可能為非線性函數，兩個延伸方向為處理凹集合的Pareto集合，還有非線性目標函數的處理，這兩個方向都是本論文尚未接觸討論的部分；再進行較為完善的討論後，就能將 β -Pareto預測法應用在較為實際的問題上。

參 考 文 獻

- [1] R. Fletcher, *Practical Methods of Optimization*. Wiley, 1980.
- [2] P. Y. Papalambros and D. J. Wilde, *Principle of Optimal Design*. Cambridge, 2 ed., 2000.
- [3] G. B. Dantzig and M. N. Thapa, *Linear Programming*. Springer-Verlag, 1997.
- [4] I. Maros, *Computational Techniques of The Simplex Method*. Kluwer Academic Publishers, 2003.
- [5] K. M. Miettinen, *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, 1998.
- [6] V. Pareto, *Manuel d'Economic Politique*. Girard, 2 ed., 1927.
- [7] M. Zeleny, *Compromise programming*. University of South Carolina Press, 1973.
- [8] S. Gass and T. Saaty, "The computational algorithm for the parametric objective function," *Naval Research Logistics Quarterly*, vol. 2, p. 39, 1955.
- [9] L. Zadeh, "Optimality and non-scalar-valued performance criteria," *IEEE Transactions on Automatic Control*, vol. 8, pp. 59–60, 1955.
- [10] I. Kim, "Adaptive weighted sum method for biobjective optimization: Pareto front generation," *Structural Multidisciplinary Optimization*, vol. 29, pp. 149–158, 2005.
- [11] S. Marglin, *Public Investment Criteria*. MIT Press, 1967.
- [12] H. W. Corley, "A new scalar equivalence for pareto optimization," *IEEE Transactions on Automatic Control*, vol. 25, pp. 829–830, 1980.
- [13] R. E. Wendell and D. N. Lee, "Efficiency in multiple objective optimization problems," *Mathematical Programming*, vol. 12, pp. 406–414, 1977.
- [14] I. Das and J. Dennis, "Normal-boundary intersection: An alternate method for generating pareto optimal points in multicriteria optimization problems," *Institute for Computer Applications in Science and Engineering*, 1996.

- [15] J. P. Dauer, "An equivalence result for solutions of multiobjective linear programs," *Computer and Operations Research*, vol. 7, pp. 33–39, 1980.
- [16] J. G. Ecker and I. A. Kouada, "Finding all efficient extreme points for multiple objective linear programs," *Mathematical Programming*, vol. 14, pp. 249–261, 1978.
- [17] J. G. Ecker, N. S. Hegner, and I. A. Kouada, "Generating all maximal efficient faces for multiple objective linear programs," *Journal of Optimization Theory and Applications*, vol. 30, pp. 353–381, 1980.
- [18] J. P. Dauer and R. J. Krueger, "A multiobjective optimization model for water resources planning," *Applied Mathematical Modelling*, vol. 4, pp. 171–175, 1980.
- [19] J. P. Dauer and Y.-H. Liu, "Solving multiple objective linear programs in objective space," *European Journal of Operational Research*, vol. 46, pp. 350–357, 1990.
- [20] D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*. Addison-Wesley, 1973.
- [21] J. P. Dauer, "Analysis of the objective space in multiple objective linear programming," *Journal of Mathematical Analysis and Applications*, vol. 126, pp. 579–593, 1987.
- [22] R. J. B. Wets and C. Witzgall, "Algorithm for frames and lineality spaces of cones," *Journal of Research of the National Bureau of Standards-B. Mathematics and Mathematical Physics*, vol. 71B, pp. 1–7, 1967.
- [23] J. P. Dauer, "On degeneracy and collapsing in the construction of the set of objective values in a multiple objective linear program," *Annals of Operations Research*, vol. 47, pp. 279–292, 1993.
- [24] Y. Y. Kagan, "Magnitude-frequency distribution in the european-mediterranean earthquake regions-comment," *Tectonophysics*, vol. 245, pp. 101–105, 1995.
- [25] G. C. Larsen and K. S. Hansen, "Database on wind characteristics-analysis of wind turbine design loads," *Riso National Laboratory Roskilde*, vol. Riso-R-1473(EN), 2004.
- [26] R. E. Melchers, *Structural Reliability-Analysis and Prediction*. Wiley, 2 ed., 2002.

- [27] A. Haldar and S. Mahadevan, *Probability, Reliability, and Statistical Methods in Engineering Design*. Wiley, 2000.
- [28] A. Haldar and S. Mahadevan, *Probability, Reliability, and Statistical Methods in Engineering Design*. John Wiley and Sons, 2000.
- [29] B. Fiessler, H.-J. Neumann, and R. Rackwitz, “Quadratic limit states in structural reliability,” *ASCE J Eng Mech Div*, vol. 105, no. 4, pp. 661–676, 1979.
- [30] J. Tu, K. K. Choi, and Y. H. Park, “A new study on reliability-based design optimization,” *Journal of Mechanical Design*, vol. 121, pp. 557–564, 1999.
- [31] F. Levi, M. Gobbi, and G. Mastinu, “An application of multi-objective stochastic optimisation to structural design,” *Structural Multidisciplinary Optimization*, vol. 29, pp. 272–284, 2005.
- [32] Z. Li, L. E. Izquierdo, M. Kokkolaras, S. J. Hu, and P. Y. Papalambros, “Multiobjective optimization for integrated tolerance allocation and fixture layout design in multistation assembly,” *Journal of Manufacturing Science and Engineering*, vol. 130, 2008.
- [33] D.-J. Yu, “多目標最佳化之解集合在不確定因素下之分析及預測：analyses and predictions on the shift of pareto set of multi-objective optimization problems under uncertainty,” Master’s thesis, National Cheng Kung University, Tainan, Taiwan, June 2008.
- [34] P. Y. Papalambros, “Remarks on sufficiency of constraint-bound solutions in optimal design,” *ASME Journal of Mechanical Design*, vol. 115, pp. 374–379, 1993.
- [35] D. R. Jones, C. D. Perttunen, and B. E. Stuckman, “Lipschitzian optimization without the lipschitz constant,” *Journal of Optimization Theory and Applization*, vol. 79, pp. 157–181, 1993.
- [36] S. Ruzika and M. M. Wiecek, “Suvey paper : approximation methods in multiobjective programming,” *Journal of Optimization Theory and applications*, vol. 126, no. 3, pp. 473–501, 2005.
- [37] X. Q. Yang and C. J. Goh, “A method for convex curve approximation,” *European Journal of Operational Research*, vol. 97, pp. 205–212, 1997.

附錄一：專有名詞中英文對照表

Advanced First Order Second Moment,AFOSM 進階一階二次可靠度方法

Attainable Set 可達空間

Block Sandwich Approximation 區塊三明治夾擠法

Cone 錐體

Extreme Point 角解

Efficient solution 效率解

Frame 架構

First Order Reliability Method,FORM 一階可靠度方法

First Order Second Moment,FOSM 一階二次可靠度方法

Nondominated solution 非凌駕解

Pivot 高斯運算

Reduced Cost Coefficients 縮減成本係數

Second Order Reliability Method,SORM 二階可靠度方法

Simplex 簡算法

Slack Variable 差額變數

Trade-Off 權衡得失

Vertex 角點

自 傳

姓名：林東信

籍貫：台中市

出生日期：民國74年8月9日

學經歷：

- 私立明道高級中學 (89.9~ 92.6)
- 國立清華大學動力機械工程學系畢業 (92.9~ 96.6)
- 國立成功大學機械工程學系碩士班畢業 (96.9~ 98.6)

通訊處：408台中市南屯區黎明路一段173巷2弄34號

聯絡電話：(04)-2479-0775

