東　　海　　大　　學

工業工程與經營資訊學系

碩士論文

具資源限制及趕工計畫整合之模糊多目標非線性規劃多專案排程研究—結合遺傳演算法

研 究 生：林士戎

指導教授：張炳騰 教授

曾宗瑤 教授

中 華 民 國 一〇一 年 六 月

A Study of Multi-Project Scheduling with Fuzzy Multi-Objective Nonlinear Programming Technique, Resource Constraint and Crashing Plan integrated - Utilizing Genetic Algorithm

By
Shih-Rong Lin

Advisor: Prof. Ping-Teng Chang
Prof. Tsueng-Yao Tseng

A Thesis
Submitted to the Institute of Industrial Engineering and Enterprise Information at Tunghai University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in
Industrial Engineering and Enterprise Information

June 2012
Taichung , Taiwan , Republic of China

具資源限制及趕工計畫整合之模糊多目標非線性規劃多專案排程研究－結合遺傳演算法

學生：林士戎　　　　　　　　　　　　　　　　　指導老師：張炳騰 教授

曾宗瑤 教授

東海大學工業工程與經營資訊學系

摘　　要

本研究目標為具資源限制、現金流量、作業趕工成本與時間、多目標與模糊考慮之多專案排程模式整合之研究。有限的資源可以包括如：人力、資金、機器設備、物料等，其數量限制可以影響原本不具先後順序的作業也因資源的限制，而須排出其先後關係。此外，專案作業於時間上之成本與收入及現金流量淨現值（net present value, NPV），在專案的排程上的選擇也非常重要，因此也須考慮。同時，在專案的規劃階段，需先設定時程與成本的目標，但此兩者往往在執行過程中，受外在不確定因素影響而與規劃之目標有所落差。

當專案因進度嚴重落後，無法準時完工之情形時則需進行趕工，方可使專案仍可在預定完工日準時完工。以上因素在過去文獻已有探討，但並無文獻於多專案排程中完整的探討這些因素中，因此本研究建構出一多目標非線性規劃模式。並結合了模糊集合理論，針對參數（如：資源、成本、折現率等）利用模糊數學來進行模糊，而獲得模糊數學展開之模糊多目標非線性規劃模式，讓本研究的數學模式更接近現實情況。亦不同於過去運用模糊理論之專案排程文獻，本研究運用模糊目標的方式來將式子模糊。

最後本研究將所有的目標函數和限制式合併成為一適應函數，此模式是將所目標函數和限制式常態化後，計算出目標函數和限制式之間的關係，最大的好處是在於，在計算目標函數時，限制式不只有限制的功能，而是可以將違反的懲罰量，加諸於目標函數，形成一個新的適應函數。並利用泛用啟發式演算方法（metaheuristics）中的基因演算法找出最佳解，尋找此模式的最佳目標：最小化總專案工期與最大化總專案淨現值。

關鍵字詞：專案管理、多專案排程、資源限制、現金流量淨現值、趕工作業成本與時間、模糊數學、模糊多目標非線性規劃、限制演化最佳化-懲罰函數、遺傳演算法

A Study of Multi-Project Scheduling with Fuzzy Multi-Objective Nonlinear Programming Technique, Resource Constraint and Crashing Plan integrated - Utilizing Genetic Algorithm

Student: Shih-Rong Lin   Advisor: Dr. Ping-Teng Chang
Dr. Tseng-Tzung Yau

Department of Industrial Engineering and Enterprise Information
Tunghai University

ABSTRACT

Nowadays a project scheduling technique capable of effectively planning project agenda and resource allocations can be an extremely important foundation for successful implementation of the project. The limited resources can include, e.g. workforce, capitals, machine, materials, etc. and their capacities can influence activities that originally have no precedence relationships but now due to limited and shared resources they are having precedence relationships in scheduling. Moreover, the costs and received payments（cash in- and out-flows）over time of the projects and their new present value should be considered in scheduling the projects too. Also, an activity’s cost and correspondingly time can be a decision variable in real world too, as planning a project or multi-projects. Though the above considerations have been taken into in various literatures, yet none of them have considered all the factors completely in multi-project scheduling situations.

Therefore, in this research, we have proposed and developed an integration of multi-project scheduling with resource constraints, cash flows, operation crash cost and crash time, multi-objectives, and fuzzy consideration for multi-project scheduling situations. Inclusively, besides the factors considered, they are also considered in fuzziness with fuzzy arithmetic, and result in the fuzzy-arithmetic operated fuzzy multi-objective programming modeling, and this is different from the past literature where only the concept of fuzzy goals was applied in the project scheduling. And in order to solve this resulting fuzzy multi-objective nonlinear programming, a newly developed technique also in this project, as an effective penalty function method of constrained evolutionary optimization, with genetic algorithm is applied, and which effectively provides the solution to the problem.

Keywords: Project management, multi-project scheduling, resource constraints, cash-flow new present value, fuzzy multi-objective nonlinear programming, constrained evolutionary optimization-penalty function method, genetic algorithm

誌謝

研究所兩年了，時間過得真的很快，在這兩年裡，任是到了許多人，也學習到許多東西，過得真的很充實而且很有趣，回想起研究所的生活真的充滿了無限的感慨及懷念，也很感謝在這一路上陪伴過我的人，真的很感謝。

首先要感謝我的恩師，張炳騰老師，在這兩年的指導裡，老師給予我的東西真的太多太多了，而且不只是學業方面，對於人生的啟發也給予我許多的指導及鼓勵，雖然每次進度都未能達到老師的標準，但老師在指導完我們之後，還是會給予我們許多的鼓勵，真的很感謝老師，要不是老師，我可能學習不了這麼多東西；也很感謝 103B 的學長姐們，建中、昱宏、鴻祥、誌宏、之中學長和秀珊及郁雅學姐，感謝你們的幫忙，讓我在一開始就能很快的習慣研究所的步調，也讓我體會到研究的辛苦及快樂；感謝我在 IKS 一路上的好兄弟世念，要不是有你一起同甘共苦、患難與共，我想研究所可能會更加辛苦，感謝在這一路上讓我有更多人生體驗的映麟，感謝你讓我見識到更多人生的方向，也謝謝俊志的幫忙，真的很懷念我們那謝一起玩樂的時光，感謝研究所的好夥伴東東、邵義、貞翔、義琳、肥料、彥傑、阿牧、姿瑜、品芳、哈哈魚、怡華、煒喬、小蔡、阿楷、卜元、星星等等，讓我在研究的路上不寂寞，也感謝學弟妹小虎、子琪、富源、笨蛋、宇凡，在我們研究辛苦的時候，給予我們許多歡笑和快樂。

感謝我的高中快認識十年的好朋友們，雖然大家都各分東西，但是我們的固定聚會給予我不少的激進和目標，很喜歡跟你們在一起的時光，有目標有理想有衝勁，真的很高興認識你們。

最後要感謝我的家人，謝謝爸爸，讓我無憂無慮的過完求學的生活，感謝媽媽，讓我有更好的態度來學習成長，謝謝姐姐，給予我許多建議及方向，求學階段過了，之後就是我回饋你們的時候了。感謝在我人生中出現過的人，因為有你們讓我更加完整，謝謝。

林士戎 謹誌於

東海大學工業工程與經營資訊學系

民國一百零一年七月

目錄

圖目錄

表目錄

第一章 緒論

1.1 研究背景與動機

現今產業全球化及產品多樣化的競爭日益加劇，專案管理（project management, PM）的重要性與日俱增，已漸漸成為企業創造競爭優勢的領域之一。一般而言，專案的成功與否取決於執行專案所耗用之時間與成本。因此，一個能夠有效規劃專案時程與資源分派的專案排程技術無疑將是成就專案執行得以順利完成的重要基石。然而資源限制的問題對於專案管理來說，已經是很基本的問題，因此對於許多在有關專案排程管理的論文中，都已將資源限制納入考慮。若專案排程在資源限制的情況下，本不需要考慮前後製程順序的活動，也將需要重新的排程。

因為貨幣具有時間價值的特性，在同一作業或專案在不同時間完成會有不同的現金流量淨現值（net present value, NPV），對於專案排程上的考慮就必須更加的仔細，因為時間的關係直接影響了專案的收入，因此在專案排程的選擇上，就必須讓較重要或者利潤較高的活動先開始執行，故此考慮了時間的特性。專案趕工之決策在專案管理中，也是具有重要的參考價值，因為當活動因為某些原因而無法順利執行時，落後的部份就必須要進行趕工，這在排程等的問題中是很常見的，故此對於專案管理的決策中，就需要將趕工模式納入考量。

模糊數學運用在專案管理中，目的是在於要讓專案排程的結果，更接近實際的結果，而且不只是排程的結果，還有專案的淨現值或成本等的運算結果，都將更接近實際的狀況，因此在模糊運用在專案管理中，將是必要的趨勢。

最後本研究將所有的目標函數和限制式，合併成為一適應函數，此模式是將所目標函數和限制式常態化後，利用三角函數的形式計算出目標函數和限制式之間的關係，最大的好處是在於，在計算目標函數時，限制式不是只有限制的功能，而是可以將違反限制式的懲罰量，直接加諸在目標函數上，形成一個新的最佳化函數。

由於有限資源專案排程問題的一般型式已被證實為困難的問題，因此對於大規模問題之求解，無法在合理時間內計算出最佳的排程結果。目前的解決方案大多是以泛用啟發式演算方法（metaheuristics）為解題工具，

試圖在可接受的時間之內產生近似最佳解。在本研究中所使用基因演算法就是啟發是演算法中的其中一個，運用基因演算法找初最佳解的特性，尋找出此模式的最佳目標：最小化總專案工期與最大化總專案淨現值。

1.2 研究目的

本研究的主要目的如下：

1. 在專案管理中考慮資源限制問題，並結合現金流量淨現值問題，建構一個多目標非線性規劃模式，並且考慮多個專案問題，結合出一個多目標多專案非線性規劃的數學模式。
2. 在此數學模式中，加入趕工決策模式，目的是為了讓成本處於變動個狀態，可以讓本研究的數學模式更貼近現實的結構。
3. 將此多目標多專案非線性規劃模式結合模糊理論，將此數學模式模糊化，形成一模糊多目標多專案非線性規劃模式。
4. 利用一限制演化最佳化－懲罰函數，來結合目標式和限制式的值，形成一新的適應函數，最後使用遺傳基因演算法來演化其適應函數，用來尋找專案管理中的適應解。

1.3 研究流程與架構

本研究論文流程架構圖如下：

1. 問題探討與確立目標：探討確立研究範圍與目標。
2. 相關文獻探討：藉由相關文獻的回顧，了解目前相關研究領域上成果，作為研究過程的參考。
3. 研究方法：本研究將於第三章詳細描述本研究所提出。
4. 模式實作與結果分析。
5. 結論與未來建議：本研究將於第五章做論文結論，針對本研究之建議及未來之研究方向作一概括敘述。

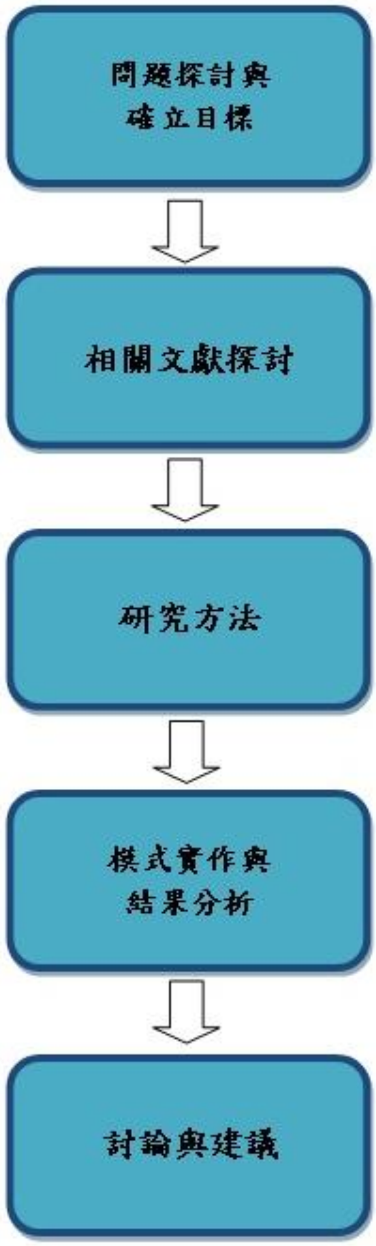

圖 1.1 研究流程圖

資料來源：本研究整理

第二章 文獻探討

2.1 有限資源專案排程問題

有許多專家學者曾為『專案』一詞下過定義。Kerzner（2001）指出專案為一個具有開始與完成時間之獨特的工作項目，必須在給定預算情形下完成各項限制與要求。Meredith and Mantel（2000）指出專案為具技術複雜性，且依賴不斷協調、控制進度、過程、成本與工作績效的一種特殊和限期內完成之任務。在綜合其他學者的看法，『專案』可定義為一組完整且具有特定目標、明確的起訖時點與成本限制的活動（稱為作業），從作業管理的角度而言，專案則是指單一訂製的個別生產。

詹蕙珍在2004年針對有限資源專案排程問題，建構以總專案淨現值最大化及總專案工期最小化為目標函數之多目標混合整數非線性規劃模式，其研究限制中，設定各作業所需工期為已知且固定，但是在研究中並未考量不同人力水準帶給工期及現金流量的影響。在有限資金排程問題研究中，Doersch and Patterson（1977）最早將資金受限因素納入最大化總專案淨現值問題進行探討，提出一個0-1整數線性規劃法。Smith-Daniels等（1996）修改Doersch and Patterson（1977）的最佳解模式，將物料管理成本納入目標函數考量。Ozdamar and Dundar（1997）則探討具機率性現金流量的專案排程問題，提出一個探索解法作為排程之參考依據。

有限資源最小化總專案工期問題主要是使專案中每一作業在不違背資源及作業先後關係的限制下，以總專案工期最小為排程目標進行作業時程安排。當所有作業均完成時，整個專案即告完成。

早期單專案排程問題法則發展的成果很大，有許多學者提出了將多個專案排程問題藉由加入一個或數個虛擬作業的方式，組成一個大的單一專案來解決此問題，此種方法稱為單專案法（single-project approach）（Patterson（1973, 1976）、Mize（1964）、Knight（1966）等）。在此之後由於專案排程的問題日漸複雜，因此Kurtulus and Davis（1982）及 Kurtulus and Narula（1986）等的學者提出多專案法（multi-project approach）的排程法則。在1993年學者邱煥能與蔡登茂則針對單、多專案法進行績效評估。其他研究多專案排程問題的學者尚有Tsai and Chiu（1996）、陳弘翊（1999）、Lova（2000）等。

Liu等人在2008年針對專案排程管理提出關鍵鏈法（Critical chain）的數學法則來解決專案排程的問題，並結合資源限制與資源分配等限制，最後利用啟發式演算法（Heuristic Algorithm）來求解問題。Suwa等人在2010年提出一個新的問題框架，其問題目的是在有限資源的情況下，利用關鍵鏈法來進行專案排程。Wang等人（2010）也利用關鍵鏈法在組織專案管理，目的是為解決在專案排程的問題中，具有資源限制的問題（Resource-Constrained Project Scheduling Problem, RCPSP）。Abad等人在2011年利用專案中時間和成本之間的關係，提出一個時間-成本權衡排程法（Time-Cost Trade-off Scheduling, TCTS），並利用基因演算法求解專案排程最佳化問題。2011年Kiriklidis等人用基因演算法來求解時間限制專案排程問題（Time Constrained Project Scheduling Problem, TCPSP）。Tian等人（2012）在資源是有限的情況下，利用關鍵鏈法來進行多專案排程。

現金流量淨現值乃是專案執行期間現金流入（cash inflow）與現金流出（cash outflow）兩者之間的分佈情形。1986年Russell首先對探索解法在此類問題的績效表現進行評估，其研究結果顯示，可使總專案工期最小化的法則未必能使總專案淨現值最大化。Abbasi and Arabiat（2001）組合兩個現有排程法則形成一個新的排程法則。國內學者陳旭明（1997）也提出在資源受限的情況下，最大化淨現值專案排程問題之研究。2002年Chiu and Tsai發表有限資源多專案排程最大化總專案淨現值問題之起始性研究。

2.2 模糊理論相關文獻

2.2.1 模糊理論

1965年由柏克萊大學Zadeh教授於發表的模糊集理論，開啟了模糊理論之發展。傳統科學都是以排除研究對象的模糊性為研究前提，而模糊理論卻是承認其模糊性為前提來做研究。在現今社會裡，我們將面對的問題的結構越來越複雜，所要考慮的事項也越來越多，而不在是以前單純的線性的結構。然而許多事情都具備其模糊性，假設預定到目的地的時間約略在十分鐘到十五分鐘，這就是預估時間的模糊性，還有包括人類的知識語言因本身的主觀意識、不同時間、環境的變遷及判斷事件的角度的不同，而

具有模糊性，也就是說模糊理論相較於傳統科學，更貼近人類的思考模式，而人類的思維方式，本身就是複雜、曖昧不明且具有不確定的現象，這些現象都可以利用模糊理論來進行處理，讓傳統科學更貼近人性。

2.2.2 模糊集合

在傳統的集合中，不是屬於就是不屬於，在這樣的關係下，很多關係都變得很簡單，不是「是」就是「否」、不是「黑」就是「白」，關係分得簡單明瞭，這就是傳統數學集合中的基本定義，然而現實生活中卻不一定能將所有的事項一分為二這樣的單純，有許多答案都是似是而非的，並不是像傳統數學那樣，這就是模糊集合存在的意義，因此模糊集合承認在黑白中間，是具有許多模糊的灰色地帶。

模糊集合的主要特色，是將事件透過人類的偏好程度來決定其所屬的標準值，例如：相較於唱歌，更喜歡跳舞，因此在跳舞的喜好程度就高於唱歌，但是並不是因此就不喜歡唱歌，只是喜好的程度不同，這就是人類的思考模式，討厭並不會全然拒絕；喜歡也未必全然接受。而模糊理論的主要意義就在於，定義對於這些事項在人類的思想下，所要表達程度上的不同，用數學的方式來呈現，因此利用公式及數學模式來計算其模糊集合，下列會有更深入的說明。

2.2.3 模糊數及隸屬函數

2.2.3.1 三角形隸屬函數

假設一個模糊集合$\tilde{A}$，以三角模糊隸屬度函數（triangular membership function, TMF）則可以表示為$\tilde{A} = (a_1, a_2, a_3), \forall A \in R$。

$$\tilde{A}(x) = \begin{cases} 0 & x < a_1 \\ (x - a_1)/(a_2 - a_1), \quad and & a_1 \leq x \leq a_2 \\ (a_3 - x)/(a_3 - a_2), \quad and & a_2 \leq x \leq a_3 \\ 0 & a_3 > x \end{cases}$$

2.2.3.2 α-cut 模糊數學

α-截集模糊數學是計算模糊集合的工具，用意在於可以將一個模糊集合轉換成明確集合，其α-截集模糊數學的定義如下：

對於一個模糊集合$\tilde{A}$而言，給定一個實數$\alpha, \alpha \in [0,1]$，則對於模糊集合$\tilde{A}$，可以形成一個明確集合$A^{\alpha} = \{x | \tilde{A}(x) \geq \alpha\}$。其中，α被視為門檻值（Threshold value）或信賴標準（Confidence Level），當α越來越大時，所對應的門檻值或信賴標準相對的越來越高，相對應的數值區間也越小，反之；若α越來越小時，所對應的門檻值或信賴標準也越來越小，區間也相對得越大。

Mizumoto and Tanaka（1979）使用了Zadeh所提出的模糊運算方法，將其利用在α-截集模糊數和區間運算。α-截集模糊數學的模糊運算中，幾個模糊數的基本運算公式可以讓計算變得更快，在每個α-level區間中，使用區間運算（Kaufmann and Gupta, 1988）。

1. 加法—有$\tilde{A}$和$\tilde{B}$兩個模糊集合，其中$B_{\alpha} = \left[b_1^{(\alpha)}, b_2^{(\alpha)}\right], \alpha \in (0,1], \forall \tilde{A}, \tilde{B} \in R$。

$$\tilde{A}_{\alpha} + \tilde{B}_{\alpha} = \left[a_1^{(\alpha)}, a_2^{(\alpha)}\right] + \left[b_1^{(\alpha)}, b_2^{(\alpha)}\right] = \left[a_1^{(\alpha)} + b_1^{(\alpha)}, a_2^{(\alpha)} + b_2^{(\alpha)}\right], \forall \alpha \in (0,1]$$

2. 減法—有$\tilde{A}$和$\tilde{B}$兩個模糊集合，其中$\alpha \in (0,1], \forall \tilde{A}, \tilde{B} \in R$。

$$\tilde{A}_{\alpha} - \tilde{B}_{\alpha} = \left[a_1^{(\alpha)}, a_2^{(\alpha)}\right] - \left[b_1^{(\alpha)}, b_2^{(\alpha)}\right] = \left[a_1^{(\alpha)} - b_2^{(\alpha)}, a_2^{(\alpha)} - b_1^{(\alpha)}\right], \forall \alpha \in (0,1]$$

3. 乘法—有$\tilde{A}$和$\tilde{B}$兩個模糊集合，其中$\alpha \in (0,1], \forall \tilde{A}, \tilde{B} \in R$。

$$\tilde{A}_{\alpha} \times \tilde{B}_{\alpha} = \left[a_1^{(\alpha)}, a_2^{(\alpha)}\right] \times \left[b_1^{(\alpha)}, b_2^{(\alpha)}\right] = \left[min\left(a_1^{(\alpha)} \times b_1^{(\alpha)}, a_1^{(\alpha)} \times b_2^{(\alpha)}, a_2^{(\alpha)} \times b_1^{(\alpha)}, a_2^{(\alpha)} \times b_2^{(\alpha)}\right), max\left(a_1^{(\alpha)} \times b_1^{(\alpha)}, a_1^{(\alpha)} \times b_2^{(\alpha)}, a_2^{(\alpha)} \times b_1^{(\alpha)}, a_2^{(\alpha)} \times b_2^{(\alpha)}\right)\right]$$

4. 除法—有$\tilde{A}$和$\tilde{B}$兩個模糊集合，其中$\alpha \in (0,1], \forall \tilde{A}, \tilde{B} \in R$。

$$\tilde{A}_{\alpha} \div \tilde{B}_{\alpha} = \left[a_1^{(\alpha)}, a_2^{(\alpha)}\right] \div \left[b_1^{(\alpha)}, b_2^{(\alpha)}\right] = \left[min\left(\frac{a_1^{(\alpha)}}{b_1^{(\alpha)}}, \frac{a_1^{(\alpha)}}{b_2^{(\alpha)}}, \frac{a_2^{(\alpha)}}{b_1^{(\alpha)}}, \frac{a_2^{(\alpha)}}{b_2^{(\alpha)}}\right), max\left(\frac{a_1^{(\alpha)}}{b_1^{(\alpha)}}, \frac{a_1^{(\alpha)}}{b_2^{(\alpha)}}, \frac{a_2^{(\alpha)}}{b_1^{(\alpha)}}, \frac{a_2^{(\alpha)}}{b_2^{(\alpha)}}\right)\right], for\ b_1^{(\alpha)}, b_2^{(\alpha)} > 0$$

以上數學算式皆為α-截集模糊數學在做數學基本四則運算時，所使用

的計算公式，本研究在計算有關α-截集模糊數學的計算時，將會利用此數學法則來做α-截集模糊數學的計算。

2.2.4 模糊多目標規劃問題

Zimmermann於1976年首先將模糊集理論引入傳統之線性規劃問題中，使原本的數學模式更趨於現實的情況。之後於1978年（Zimmermann, 1978）將決策者之模糊目標予以量化，讓這些目標可以做數學上的計算，以解決多目標線性規劃之問題。以此觀念為基礎，許多學者陸續提出考量不同決策環境之模糊多目標決策模式，如：Hannan（1981）、Leberling（1981）、Luhandjula（1982）、Sakawa（1988）、Hsu and Tzeng（1990）等。這些模式主要差異在於隸屬函數及組合各模糊目標函數之運算子型態不同。

Dubois and Prade（1997）經由歸納模糊理論文獻中所提出的各種應用，指出歸屬函數值的意涵主要分為三種；相似性（Similarity）、偏好度（Preference） 與不確定性（Uncertainty）。

相似性的理論經常出現在聚類分析（Clustering analysis）與模糊控制（Fuzzy control）的方法中，進而運用在模糊分類（Fuzzy classification）、模糊影像分析（Fuzzy image analysis）、模糊系統分析與控制（Fuzzy system analysis and control）及模糊資料庫（Fuzzy databases）等領域。偏好度則出現在決策分析（Decision analysis），用以代表效用（Utility），常被使用於模糊多準則決策中的模糊優化論（Fuzzy optimization）及決策支援系統（Decision support systems）。然而不確定性經常出現在可能性理論（Possibility theory），進而運用在人工智慧（Artificial intelligence）及專家系統（Expert systems）。

Dubois and Prade（1997）提出了當歸屬函數的意涵為相似性時，其歸屬函數的建構最好採用"距離"的測度觀念;當歸屬函數的意涵為偏好度時，其歸屬函數的建構最好採用"成本"的測度觀念；當歸屬函數的意涵為不確定性時，其歸屬函數的建構最好採用"頻率"的測度觀念。

模糊理論在專案排程領域的應用在於，先行描述其專案的作業工期之不確定性，並配合要徑法等傳統專案排程法則，最後因而求得專案總工期（Chanas and Kamburowski（1981）、Nasution（1994）、Yao and Lin（2000）

及Chen and Chang（2001））。而模糊理論被使用在工程專案排程等的相關研究的問題中，始於Ayyub and Halder（1984）兩位學者，他們將模糊集理論、模糊代數運算（Fuzzy arithmetric operations）及模糊關係（Fuzzy relation）等應用於單一作業工期預估模式與量化工期之不確定性因素。在1996年Lorterpong and Moselhi提出Fuzzy network scheduling（FNET）法，以梯形或三角形模糊數來描述專案的作業工期之不確定性，並推算專案網路圖排程的總工期。

Mon等人（1995）提出專案的作業時間以三角形模糊數來做為模糊數學的分佈（Fuzzy triangular distribution ,FTD）或以模糊常態的方式來做模糊數學的分佈（Fuzzy normal distribution, FND），且直接成本與間接成本為明確值的情況下，並配合模糊計畫評核術的方法，來對專案排程進行時間成本權衡分析。此模式引入α-截集（α-level cut）的概念，將其用於以描述模糊作業時間變異之風險程度，並透過決策者樂觀指數（index of optimism）將α-截集所切割的區間值轉換為相對的明確值；然而在相同的變異風險程度的情況下（即α值相同的條件下），若決策者設定之α值越大，則作業工期將會越短。在其所提出之模式中，α值具有「信賴區間」的意味；決策者可藉以觀察出特定之信賴區間內，不同之α值所推導出不同的專案總工期與直接成本的變化情形，藉以選出較經濟的趕工方案。

Leu等人（2001）則使用三角形或梯形模糊隸屬函數代表作業工期之不確定性，並使用α-截集的概念表現決策者對排程發生變異的可接受程度。在選定α值後，分別在專案中的各作業之工期分佈區間內亂數選定明確之代表值，並據以進行專案排程計算；而其作業直接成本的計算方式，視其所選定的代表值落在正常施工區間或是趕工區間而定，並且可選擇以業主或包商等不同觀點來進行推導。最後，其並以基因演算法作為時間成本權衡分析求解的工具。其模式可輸出在不同風險考量下的時間成本權衡曲線，因此決策者可依其對排程發生變異的可接受程度（α值）來選擇對應的時間成本權衡曲線，並從中找出適當的趕工方案。

Arikan and Gungor兩位學者在2001年提出一個新的模式，是以模糊目標規劃法（Fuzzy Goal Programming, FGP）為基礎的時間成本權衡模式，其目的是將決策者對其所偏好之專案總工期與直接成本滿意程度，轉化為模糊隸屬函數後，則原本的模糊多目標問題即可轉換為同時滿足專案總工期

與直接成本之最大化滿意程度之單目標問題並利用簡易的線性規劃求解。此模式可輔助決策人員，藉由調整量化的滿意程度，來觀察出作業的完工工期與直接成本的變化，並選擇滿意度較高（風險較低）的趕工排程。

Liu（2003）其模式以模糊計畫評核術為基礎，並引用Yager（1981）之解模糊方法將模糊作業工期轉換為明確值，用以求解專案排程之要徑與趕工問題。而針對專案排程趕工問題的求解，此研究以限定作業可趕工天數以及給定單日趕工成本明確值的方式，利用線性規劃求解趕工排程。此研究並指出，針對不確定性排程之時間成本權衡問題，過去多單以工期具不確定性的情況進行研究，故此研究建議未來的研究方向可進一步對工期與成本均為模糊值的情況作更深入的探討。

2.3 限制演化最佳化-懲罰函數法

Hoffmeister and Sprave 在 1996 年提出了一適應函數，此函數的目的是將目標式結合了所有限制式合成一個適應懲罰函數，讓模式可以利用此適應函數來求解。Morales and Quezada（1998）則制定出一個方法，將結果分為可行解和不可行解來做考慮，而且給予不可行解很大的懲罰因子，使得解不會偏向不可行解，這方法解決了不可行解的問題，也讓適應懲罰函數更可以代表原模式的意義。Deb（2000）設計了另一種懲罰函數，將適應函數分為兩個組：可行解及不可行解，目的是讓不可行解的適應值比可行解來的差，因而讓適應懲罰函數的解導向可行解。

Yokota 等人（1996）將 Hoffmeister and Sprave（1996）所提出的方法改良，製訂出一個乘法模式，將限制式給予的懲罰利用乘法加諸於原本的目標函數上。基於 Yokota 等人（1996）的方法，Gen and Cheng（1996）則將限制式常態化之後，利用乘法模式將所算出的懲罰值和目標函數結合，得出一個新的適應懲罰函數。Puzzi and Carpinteri（2008）則進一步修改 Yokota 等人（1996）的方法，制定了一懲罰函數，不僅適用於目標函數值，限制式也適用於適應值和限制變數之間的乘法關係。Farmani and Wright（2003）提出的另一種適應懲罰函數，將懲罰函數分成兩個部分來計算其懲罰值，結合成一新的適應懲罰函數。

Lin and Wu（2004）提出了自組織適應懲罰策略（self-organizing adaptive penalty strategy，SOAPS），此函數將目標函數和限制式各自常態化之後，最後結合並計算出及懲罰值，好處在於常態化後，所有的解在比較上可以很客觀，解決了解在不同基準下的比較。Wu and Lin（2004）進一步發展了第二代 SOAPS（SOAP-Ⅱ），不同於 SOAPS 在懲罰上將不可行解的限制懲罰因子做了新的定義。

Hu 等人（2005）制定了懲罰函數，將不可行解的懲罰值強加在目標函數值上，因而形成了此適應懲罰函數。Wang 等人（2009）制定了懲罰函數，將懲罰模式分為三種型態，只有不可行解、只有可行解及有不可行解和可行解的混合，分成這三種模式來求得其適應懲罰函數。Tessema and Yen（2009）則和 Wang 等人（2009）不同，他們將目標函數和限制式的值常態化後，並加以計算，和 Wang 等人（2009）相同的是，他們也分成了只有不可行解、只有可行解及有不可行解和可行解的混合的三種模式。

已有文獻對於上述各個適應懲罰函數的成果做比較，在求解適應解的效率及其適應解的品質中，以 Lin and Wu 在 2004 年所提出的自組織適應懲罰策略懲罰函數的成果最佳，不管是效率或是品質都優於其他適應懲罰函數，因此本研究將利用其結果，來做為本研究的適應懲罰函數，進而利用遺傳演算法來演化求解。

2.4 基因遺傳演算法

遺傳演算法是由 Holland（1975）發起的演算法，之後也有許多文獻提出遺傳演算法的使用方法和其應用。遺傳演算法基本上是一種非線性的機率程序，採用遺傳進化的概念做多點搜索最佳解的方法。遺傳演算法是以客觀為導向進行三項作業分別為交配或重組、突變及選擇或複製。下列有關此議題的討論是必要且具啟發的，已有大量的文獻改編或修改了遺傳演算法的運算子或參數，並對遺傳演算法進行運用，如 De Jong（1975）、Grefenstette（1986）、Schaffer 等人（1989）、Michalewicz（1992）、Mitchell（1996）、Myers and Hancock（1997）、Eiben 等人（1999）、Lobo（2000）、Mayer et al.（2001）及 Todoroki and Ishikawa（2004）。同時在 MT-GA，交配和突變被使用在特別的操作上，並且也有大量的文獻，如：Baker（1985）、Hansen and Ostermeier（2001）、Schmitt（2001）、Deb 等人（2002）。

同時，遺傳演算法的編碼方式在一些問題中已被發現，由這些問題的發現可以得知，實數解的編碼方式優於二進制的編碼方式（Janikow and Michalewicz, 1991）。因此，下面將本研究將所有遺傳演算法的編碼問題，將集中在實數編碼的遺傳演算法。在實數編碼的遺傳演算法中，染色體可以設計成一個向量或矩陣的方式或其他數字或符號，可以被使用在一個字串或矩陣的基因。

交配：在實數編碼的遺傳演算法，交配運算子除了已經設計用於排程或訂單等的目的（Chang and Lo, 2001; Chen 等, 2008; Murata 等, 1996），交配的方式已在許多文獻中被提出，如 simple, arithmetic, direction-based arithmetic（DBA）crossovers 等（Michalewicz, 1992; Michalewicz 等人, 1994），亦可以在一些文獻中發現新的交配方式，例如 Deb 等（2002），Elliott 等（2004），Leite and Topping（1998）。此外，交配運算子是一個交配概率（Pc）和可能產生的子代（B'），子代的產生方式是由在母代（B_1 and B_2）在一交配概率下，隨機交配後而產生，其公式如下：$B' = r（B_2 - B_1）+B_2$，其中 r 是一個隨機數在[0,1]，如果 B_2 的適應值大於等於 B_1 的適應值，產生的新的子代 B'，將取代原來母體中的 B_1，進而和原子代較好的適應值 B_2，進入下一階遺傳演算法的操作程序，然而此公式的好處在於可以讓適應解的品質偏向更好的適應解，在比較之後取代原來較不好的適應解，讓遺傳演算法更快的導向更好的方向。

突變：在實際編碼，為了再次突變，除了那些已被設計用於調度或訂購目的（Chen 等人, 2008；Georgiou, 2007；Murata 等人, 1996），一個數變異操作已經被提出，例如：均勻，非均勻變異等（Michalewicz, 1992），在許多文獻中已被討論，如 Hansen and Ostermeier（2001），Mayer 等人（2001），Leite and Topping（1998）。因此，每個基因（x_i）隨機選擇的染色體（$B =（x_1, \ldots,x_i, \ldots, x_n）, x_i \in [a_i, b_i]$）突變可能從習近平 $x_i=U（a_i, b_i）$與概率突變時 P_m,，其中 $U（a_i, b_i）$是一個均勻分佈的隨機數$[a_i, b_i]$.。此外，其他可利用變異算子在 MT-GA

選擇/複製：遺傳演算法中，選擇/複製是保持個體解或總體解上達到更好的下一代和更有效的搜索下一代。因此，一些運算方式已被提出，如 roulette-wheel or proportionate-fitness, tournament, linear ranking selections 等

（Georgiou, 2007; Schmitt, 2001）的方法已有文獻提出。在這個方向的發展，下列多種技術，如：linear scaling, power-law scaling, sigma-truncation 等已有文獻提出（Georgiou, 2007 ; Schmitt, 2001）。

本研究在遺傳演算法中編碼的部份，將使用實數編碼的方式來進行遺傳編碼；在交配的動作上，將使用上述的交配方式來產生新的子代，使得數學模式可以更快找出更好的適應解；在突變上，將使用單點突變的方式來進行遺傳突變，也就是說母體中每一個數都有可能進行突變，且所有的母體突變機率都相同，但母體的突變機率彼此都是獨立的，並不互相影響；在選擇的階段，將使用競爭法來做為選擇的依據，目的是將更好的解留下，進而使得母體有更好的適應解。

第三章 模糊多目標非線性規劃模式

本研究的研究重點在於，結合許多不同的方法及模式，建立一個模糊多目標非線性模式，並利用限制演化最佳化－懲罰函數，來計算其模式的適應值，最後使用遺傳基因演算法來進行模式的演化，找出此模式的適應解。其詳細的研究步驟如下：步驟一：建議一個多目標多專案非線性排程資源限制模式。步驟二：將建立的數學模式模糊化，並利用最大化最小邊界來解模糊。步驟三：利用限制演化最佳化－懲罰函數，將目標式及限制式合併，計算其適應值。步驟四：使用基因演算法來演化其適值，找出此模式的適應解。利用此流程來進行本研究的模式建立，進而讓本研究的數學模式得以求解。

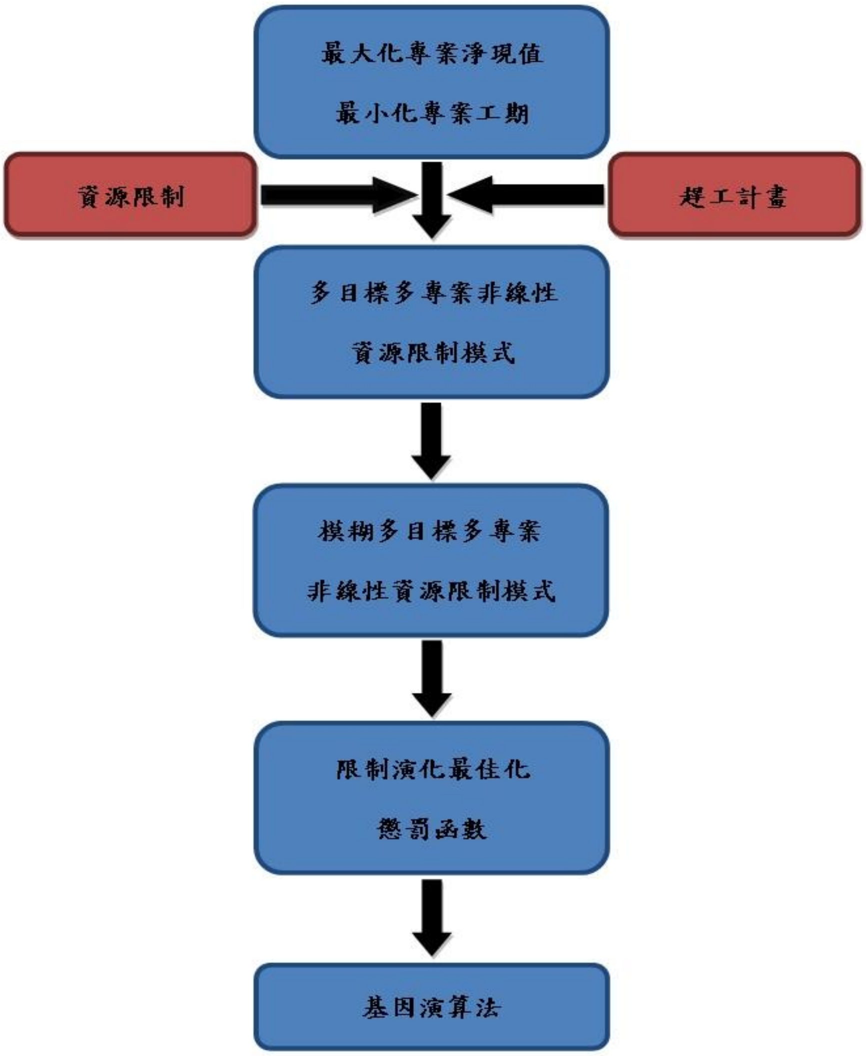

圖 3.1 模糊多目標非線性多專案規劃架構圖

資料來源：本研究整理

本章各節說明。3.1節多目標非線性規劃模式建立；3.2節模糊多目標非線性規劃模式；3.3節自組織適應懲罰策略懲罰函數；3.4節基因遺傳演算法。

3.1 多目標非線性規劃模式建立

根據詹蕙珍在2004年針對有限資源專案排程問題，本研究將延用其多目標非線性規劃數學模式，不同於其模糊數學的運用，本研究將利用模糊目標的方式來將數學模式模糊，只將需要模糊的參數來進行模糊，進而得到一資源受限之模糊多目標非線性規劃模式；在專案排程的作業中，經常會有作業工期不確定的時候，因此不同於原數學模式的計算方式，本研究在輸出成本中，加入了趕工模式的運算，此目的是為了讓本研究的多目標非線性規劃模式，更加貼近實際的生活中有可能發生的各種狀況。

3.1.1 數學模式符號定義

本研究根據詹蕙珍（2004）所提出的有限資源專案排程問題做為研究基礎，並且在經過數學計算上的修改之後，進而制訂出本研究的多目標非線性規劃模式，其數學模式的符號定義如下表3.1：

表 3.1 數學模式符號對照表

符號	代表意義
i	專案代號；$i=1,\cdots,I$，此處I為總專案各數。
j	作業代號；$j=1,\cdots,J_i$，J_i為專案i的作業總各數。
PL	總排程計劃長度。
t	排程時間點， $t=1,\cdots,PL$。
ω	折現率。
CP_i	專案i在無資源限制下的關鍵路徑總工期。
PD_i	專案i到期日。

符號	代表意義
G_i	專案i每期提早完工獎金。
B_i	專案i每期延遲完工成本。
FT_{ij}	專案i的作業j的完工時間點。
FT_{iJ_i}	專案i的完工時間點。
FT_{max}	資源限制下的總專案工期，此處$FT_{max} = \max_{\forall i}\left\{FT_{iJ_i}\right\}$。
INC_{ij}	專案i的作業j在其完工時點所產生的現金流入。
OUC_{ij}	專案i的作業j在其完工時點所產生的現金流出。
NVP_{ij}	專案i的作業j在其完工時點所產生的淨現金流量。$INC_{ij} - OUC_{ij}$
INC_{iJ_i}	專案i完工時的工程尾款現金流入。
c_{nij}	專案i的作業j的正常成本。
c_{kij}	專案i的作業j的趕工成本。
t_{nij}	專案i的作業j的正常工期。
t_{kij}	專案i的作業j的趕工工期。
rb_{ij}	專案i的作業j的趕工率。
BO_{ij}	專案i的作業j所有緊接先行作業的集合。
Y_{ijt}	$\begin{cases} 1, & and\text{若專案}i\text{的作業}j\text{在時點}t\text{正在進行；} \\ 0, & and\text{其他情形。} \end{cases}$
X_{ijt}	$\begin{cases} 1, & and\text{若專案}i\text{的作業}j\text{在時點}t\text{完工；} \\ 0, & and\text{其他情形。} \end{cases}$
r_{ijk}	專案i的作業j對第項資源每k期的資源需求量。
R_k	資源k每期系統可使用資源數。
TPV_i	專案i的工程總價款。
γ_i	專案i的報酬率。

符號	代表意義
Z_i	$\begin{cases} 1, & and 若 FT_{iJ_i} > PD_i；\\ 0, & and 其他情形。\end{cases}$
g	專案每期提早完工獎金率。
b	專案每期延遲罰款率。
ES_{ij}	專案i的作業j在無資源限制下的最早開始時間。
EF_{ij}	專案i的作業j在無資源限制下的最晚完成時間。

資料來源：本研究整理

3.1.2 數學模式基本假設

為了確保本研究的多目標非線性規劃模式的穩定性，將訂定幾項基本假設，目的是要讓其數學模式規則可以依循，並且不能違背其規則，基本假設如下：

1. 各專案內之作業先後關係已事先制定出且不能修改，在制定排程時需滿足作業先後關係限制。
2. 每項作業所需正常工期（t_{nij}）及趕工工期（t_{kij}）均為已知且固定，而各作業所需的資源數量（r_{ijk}），皆保持一定。
3. 每項作業只要開始執行之後，則不可中斷且不可事先預置，完全是靠排程出來的順序來執行期作業。
4. 本研究之資源為不可儲存性資源，故每期可供應的字資源總數（R_k）為已知，但若有剩餘的情況，則不可沿用至下一期或之後的作業。
5. 專案到期日（PD_i）為已知，專案完工日若超過到期日，則產生延遲罰款，專案若提早完工則有獎金，每期獎金為已知且固定。
6. 作業的現金流入（INC_{iJ}）與為已知且固定，現金流量折現率（ω）為已知且固定，且以作業的實際完工時點（FT_{ij}）作為淨現金流量折現計算之基準。
7. 現金流出（OUC_{iJ}）採用趕工模式，專案i的作業j有正常成本（c_{nij}）、趕工成本（c_{kij}）、正常工期（t_{nij}）及趕工工期（t_{kij}），以上四個參數

經過運算，就會變成專案i的作業j的作業成本。

以上各項假設必須要嚴格的遵守，不能違背，其中以第一點尤其重要，本研究的數學模式將利用以上的基本假設來進行專案排程。

3.1.3 資源限制之多目標非線性規劃模式

專案管理中，有許多指標都可以用來評估專案排程的成效，然而在進行數學模式建立時，很難將所有的目標納入考慮，因此本研究將專案排程問題的目標專注於專案的總淨現值及總專案工期，利用這兩項指標來評估專案的排程後預期的執行成效，選擇這兩項指標的原因是因為在專案管理中，現金的流量和運作的時間都必須進行控制，才能確保專案的品值，因此本研究就以最大化總專案淨現值及最小化總專案工期，兩項目標來當做本研究的目標函數。

在限制式方面，由於本研究數學模式的初始排程結果，是隨機進行排程來決定的，因此為了避免其初始排程結果違反了在開始執行專案排程前，所規劃制定的專案活動的先後順序，為此制定一個限制式，其目的是為了防止制定好的專案排程的後製程，不會比前製程要來的提早開始執行，進而阻止不合理的排程結果產生。

在本研究的數學模式中，專案活動的各項資源都是有限的，每個活動在每一期活動時間執行時，所使用的各項資源量並不是無止境的取用，因此本研究利用此限制式，將其專案執行活動每期使用的各項資源量加以限制，讓本研究的數學模式為一資源限制規劃模式。

以下為本研究的資源限制之多目標非線性規劃模式，如下列各式：

● 目標函數一：總專案淨現值最大化

$$Max\ f_1 = \sum_{i=1}^{I}\sum_{j=1}^{J_i-1} NVP_{ij}exp\ \ (-\omega FT_{ij}) + \sum_{i=1}^{I} INC_{iJ_i}\,exp\left(-\omega FT_{iJ_i}\right) + \sum_{i=1}^{I}(1-Z_i)\,G_i\ (PD_i - FT_{iJ_i})\ exp\ \ (-\omega FT_{iJ_i}) - \sum_{i=1}^{I} Z_i\,B_i\ (FT_{iJ_i} - PD_i)\ exp\ \ (-\omega FT_{iJ_i}) \tag{1}$$

每個專案活動在完成後都會有將會有現金的流入，專案的各項活動方可進行，其中NVP_{ij}為每個『專案活動』完成後的現金流入，INC_{iJ_i}則為每個『專案』在結束時，所謂的專案工程尾款的現金流入，本研究將此二個專案的現金流入，當成專案排程的收入，並加上在專案結束後，提早或延遲的獎懲金額（G,B），經過數學運算後，成為本數學模式的總專案排程淨現值，Z 值的目的是用來判斷此專案的排程結果，是否提早結束或者有延遲的狀況，其中Z 值若為零，表示專案的排程結果提早完成，因此給予一定程度得獎金，若 Z 值為 1，表示專案的排程結果有延遲的狀況，則給予罰款，$Z = 0\ or\ 1$。

由於貨幣具有時間的價值，因此在現金的流入或流出都要計算其時間的價值，尤其本研究的排程為隨機的狀態，每次專案活動結束的時間都會有所不同,因此每次排程後的淨現值都不相同,而本研究對於時間的影響，是將其折現率乘上每個專案或專案活動的完成時間，再將其指數化。

而數學模式中的指數化，就是本研究非線性方法的運用，將原本是線性關係的淨現值，利用指數化的方式來轉變為非線性關係，相較於原本的線性模式，非線性模式更符合現實的狀態。在現實世界中，專案的最後完成時間拖得越長，所能得到的相對收入也會因此減少，而指數分配就是屬於這種曲線的現象，因此本研究將利用指數分配的方式,讓『目標函數一』更能貼進現實的情況，因此在專案的個項活動的完成時間乘上折現率後，再將其指數化,『目標函數一』就可以得到時間對於淨現值的影響，最後乘

上淨現值後並做運算後，取其最大化淨現值，將可以得到第一個目標函數，『目標函數一：總專案淨現值最大化』。

- 目標函數二：總專案工期最小化

$$Min\ f_2 = FT_{max} \tag{2}$$

專案的總工期一直都是判斷專案的執行成效的指標之一，專案的總工期越短，對於公司組織的各種的彈性也越大，因此在決定專案各項活動的工期時，就希望能將其最小化，但由於本研究的數學模式的運作方式，讓專案排程的結果為隨機性的產生，因此要規定所有專案各項活動的工期，達到該工期最小化有一定程度的困難，所以本研究將直接限制其總工期最小化，利用專案活動的總工期最小化，來壓縮專案各項活動的工期，藉以達成總專案工期最小化的目標。

在本節一開始時，有提到此多目標非線性規劃模式，將由兩條限制式來限制其模式的條件，其中第一條限制式是為了要限制其專案排程的前後製程的關係；第二條的目的是為了其數學模式在資源的使用上有所限制，以下是此兩條限制式說明：

- 限制式一：前後製程的關係限制

$$FT_{iy} \geq FT_{ij} + t_{niy}, \forall j \in BO_{iy}; i = 1, \cdots, I; j = 1, \cdots, J_i; y = 1, \cdots, J_i; \tag{3}$$

$FT_{ij} \geq 0$且為整數$, i = 1, \cdots, I; j = 1, \cdots, J_i;$

式（3）表示專案排程時前後製程的關係，FT_{iy}為專案i活動y的完工時間，代表的是後製程的完工時間，FT_{ij}為FT_{iy}的前製程的完工時間，t_{niy}為專案i活動y的正常工期，此限制式所要代表的意思是，當前製程的完工時間（FT_{ij}）加上後製程的正常工期（t_{niy}），其結果不能大於後製程的完工時間（FT_{iy}），此限制式的目的是為了增加專案排程結果的穩定性，讓排程結果趨於的合理化。

● 限制式二：資源限制

$$\sum_{i=1}^{I}\sum_{j=1}^{\boldsymbol{J_i-1}} r_{ijk}\times Y_{ijt}\leq R_k, k=1,\cdots,K; t=1,\cdots,H; \tag{4}$$

其中：

$$Y_{ijt}=\sum_{v=t}^{min\{t+t_{nij}-1,PL\}} X_{ijv},\ i=1,\cdots,I; j=1,\cdots,J_i-1; t=ES_{ij},\cdots,PL; \tag{5}$$

$$\sum_{v=ES_{ij}}^{PL} vX_{ijv}=FT_{ij}, i=1,\cdots,I; j=1,\cdots,J_i \tag{6}$$

$$\sum_{v=ES_{ij}}^{PL} X_{ijv}=1\,, i=1,\cdots,I; j=1,\cdots,J_i \tag{7}$$

$$X_{ijt},Y_{ijt},Z_i=0\text{ or }1, i=1,\cdots,I; j=1,\cdots,J_i; t=1,\cdots,PL;$$

式（4）在本研究的數學模式中，扮演讓資源受限的限制式，其目的是為了要限制專案各項活動在每一期活動期間的各項資源使用量，例如：專案人員總數為十個人，因此每期專案活動可以使用的人力資源總數上限就為十個人，但每個專案活動所使用的人力資源的數目並不相同，如何找出最適當的人力資源的使用分配就變得相當的重要。

在式（4）中，R_k為第k種資源在每期專案活動中可以使用資源上限，而r_{ijk}則為專案i活動j在資源k的使用量，而Y_{ijt}則表示專案i活動j在作業工期t時正在進行活動，Y_{ijt}是由X_{ijt}計算得來，由式（6）中可以得知FT_{ij}代表專案i活動j的完工時點，當tX_{ijt}等於FT_{ij}時，就可知道X_{ijt}專案i活動j在工期t時完工，因此在式（4）中的$\sum_{i=1}^{I}\sum_{j=1}^{\boldsymbol{J_i-1}} r_{ijk}\times Y_{ijt}$代表資源（$r_{ijk}$）在各期專案活動使用量的總和，且各期專案活動的資源使用量總和，不能超過資源的上限（R_k）。

在現實情況中，任何工作都會有趕工的情況，因此本研究的數學模式

多加了趕工模式，而在趕工的過程當中，成本會有增加的現象，本研究的數學模式利用正常成本（c_{nij}）、趕工成本（c_{kij}）、正常工期（t_{nij}）及趕工工期（t_{kij}），來計算趕工的狀態。

- 趕工模式:

$$OUC_{ij} = \frac{(c_{kij} - c_{nij})}{(t_{nij} - t_{kij})} \times rb_{ij} + c_{nij} \tag{8}$$

在式（8）中，分式所計算出的是一個趕工斜率的計算式，如圖3.1，可以計算出從正常狀態下工作的情況到趕工狀態工作情況的變化，而式中的rb_{ij}則代表專案i活動j是否需要趕工的參數（$rb_{ij} \epsilon [0,1]$），當$rb_{ij} = 1$時，代表的是專案i活動j需要進行趕工的動作；相反的，當$rb_{ij} = 0$時，則代表不需要進行趕工的動作，而藉由0到1之間的數值，則代表不同的趕工情況，而在本研究中rb_{ij}是藉由0到1之間隨機產生的值，代表著每個專案活動在開始時，並不知道是否要進行趕工。

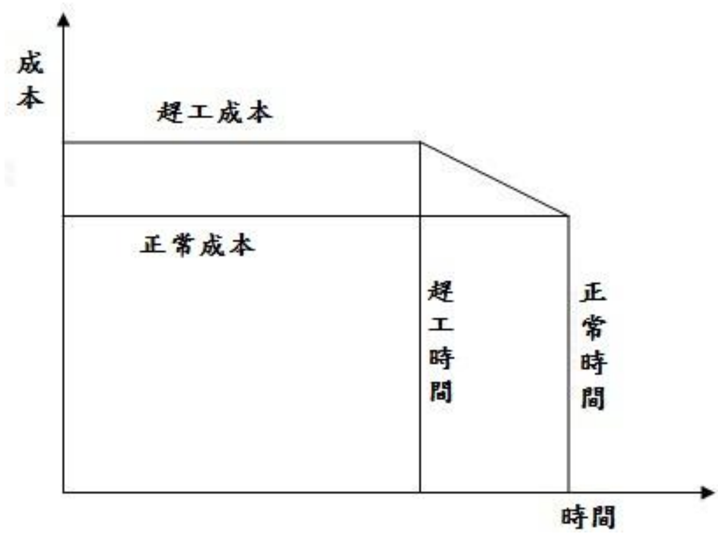

圖 3.2 趕工模式圖

資料來源：本研究整理

式(8)所要代表的是專案活動的輸出值，也就是所謂的專案活動成本，然而在原模式中，專案活動成本就等於正常成本（$OUC_{ij} = c_{nij}$），而在本

研究中加入了趕工模式，讓專案活動成本處於跳動的狀態，也更加符合現實的情況。

- 專案總價：

$$TPV_i = \left\{ \sum_{j=1}^{J_i - 1} OUC_{ij} \exp\left(\omega\left(CP_i - EF_{ij}\right)\right) \right\}\left(1 + \gamma_i\right) \tag{9}$$

專案總價（TPV_i），是由專案活動成本（OUC_{ij}）乘上專案活動的報酬率（γ_i）計算而得，計算專案總價的目的是為了計算各專案完工時的工程尾款現金流入（INC_{iJ_i}），計算公式如下：

$$INC_{iJ_i} = TPV_i - \sum_{j=1}^{J_i - 1} INC_{ij} \exp\left(\omega\left(CP_i - EF_{ij}\right)\right) \tag{10}$$

而專案結束時都有所謂的獎金（G_i）或罰款（B_i），其計算方式都是由專案總價（TPV_i）乘以一定的比例（$gand\ b$）來做計算，其計算方式如下：

$$G_i = TPV_i \times g \tag{11}$$

$$B_i = TPV_i \times b \tag{12}$$

3.2 模糊多目標非線性規劃模式

在現實世界的情況，明確值只存在理論中，因此本研究使用了模糊數學，其目的是為了要讓本研究的數學模式更加貼近現實，和原模式不同的是，本研究所使用的模糊方式，是將變動可能性較大的參數來進行模糊，而不是將所有的參數或變數模糊，故此在計算上較為簡潔，也更能看出模糊後的差異。而本節的重點在於將原本的多目標非線性規劃模式進行模糊，進而變成模糊多目標非線性規劃模式。

3.2.1 三角模糊數

本研究將使用對稱三角模糊數（triangular fuzzy numbers, TFNs）來做為模糊的方式，對稱三角模糊數是將一個數值分成三個，並以原來的點作為中心點，形成一個等腰三角形，而在這範圍內的數值稱為一個模糊集合，在此集合內的數值都會有一個相對應的隸屬程度，進而形成一個隸屬函數，在這模糊集合內的數值，可以藉由此隸屬函數找到屬於自己的隸屬程度。

假設$\tilde{x}_{ki} =$ （x_{ki}, a_{ki}） 為三角模糊數（*TFNs*），x_{ki}和a_{ki}進行展開，下列為此模式的隸屬函數：

$$\tilde{x}_{ki} = (x_{ki} - a_{ki}, x_{ki} x_{ki} + a_{ki}) \tag{13}$$

$$0 \le a_{ki} \le x_{ki}, 1 \le i \le |I|, k \in P$$

在式（13）中，x_{ki}代表的是三角型的中心點，a_{ki}代表的是此三角模糊數的延伸值，$\tilde{x}_{ki}$代表是一個模糊集合，其範圍由模糊右界$x_{ki} + a_{ki}$，到模糊左界$x_{ki} - a_{ki}$，有此可以找出將對應於模糊集合的隸屬函數$\mu_{\tilde{x}_{ki}}(x)$，其隸屬函數如下：

$$\mu_{\tilde{x}_{ki}}(x) = \begin{cases} 0, & x < x_{ki} - a_{ki} \\ (x - x_{ki} + a_{ki}) / (a_{ki}), & and \quad x_{ki} - a_{ki} \le x \le x_{ki} \\ (x_{ki} + a_{ki} - x) / (a_{ki}), & x_{ki} \le x \le x_{ki} + a_{ki} \\ 0, & x > x_{ki} + a_{ki} \end{cases} \tag{14}$$

由式（14）可以看出，當超出模糊集合的範圍時，其隸屬函數$\mu_{\tilde{x}_{ki}}(x)$就會等於零，而其他模糊集合的數值都由其相對應的隸屬函數值。

- **定義 1**：給定兩個對稱的三角模糊數$Z_1 =（z_1, w_1）$和$Z_2 =（z_2, w_2）$的關係可以定義為以下不等式：

$$\begin{aligned} z_1 - (1-h)w_1 &\le z_2 - (1-h)w_2 \\ z_1 + (1-h)w_1 &\le z_2 + (1-h)w_2 \end{aligned} \qquad (15)$$

上述兩式，可以看出$Z_2 =（z_2, w_2）$恆大於$Z_1 =（z_1, w_1）$，本研究將利用此二式的觀念運用在計算限制式上，右式要恆大於左式，在模糊數學的計算上也要如此，因此在限制式計算上可以利用式（15）的觀念，讓模糊後的限制式，右式會恆大於左式，而不改變其限制式的本意。

參照**定義 1**，一個最大化的三角模糊數$Z =（z, w）$可以解釋為最大限度地發揮其$z - (1-h)w$和$z + (1-h)w$兩式的功能，因此有了一個加權函數$\lambda_1(z-(1-h)w) + \lambda_2(z+(1-h)w)$可用於某些妥協的解決方案，$\lambda_1 \ge 0, \lambda_2 \ge 0$為權重且 $\lambda_1 + \lambda_2 = 1$，用於左右為$h$水準的$Z$模糊函數。$\lambda_1 = 1$在最大化的目標函數中被認為是悲觀的解，而$\lambda_2 = 1$是一個樂觀的解，被認為是最好的情況。在本文中則採用$\lambda_1 = 1$，即最大化最小值，如下式：

$$Max\, z - (1-h)w \qquad (16)$$

本研究的目標式利用上式的方法，來計算得到模糊目標式，最後結合了限制式的模糊，而成為本研究的模糊多目標非線性規劃模式。

本研究將成本（OUC_{ij}）、折現率（ω）、報酬率（γ_i）還有將作業中使用的資源（r_{ijt}）及每期作業的總限制量（R_k）進行模糊，由於本模式有最大化總專案淨現值及最小化總專案工期，兩個目標函數，因此最大化目標函數如（16）式所示，最小化目標函數則反之。

3.2.2 模糊多目標非線性規劃模式

以下數學模式將以上述兩種模糊運算方法，個別將目標式及限制式來進行模糊，首先將目標函數一進行模糊，其模糊結果如下：

- 目標函數一：總專案淨現值最大化

$$\begin{aligned} Max\ f_1 = \sum_{i=1}^{I}\sum_{j=1}^{J_i-1} NVP_{ij} \\ \times exp\ (-(\omega-(1-h)\times\omega')\times FT_{ij}) \\ +\sum_{i=1}^{I} INC_{iJ_i}^{1}\times exp\big(-(\omega-(1-h)\times\omega')\times FT_{ij}\big) \\ +\sum_{i=1}^{I}(1-Z_i)\times G_i\times(PD_i-FT_{iJ_i}) \\ \times exp\ (-(\omega-(1-h)\times\omega')\times FT_{ij})-\sum_{i=1}^{I} Z_i \\ \times B_i\times(FT_{iJ_i}-PD_i) \\ \times exp\ (-(\omega+(1-h)\times\omega')\times FT_{ij}) \end{aligned} \qquad (17)$$

在目標函數一中，本研究利用式（16）中最大化最小值的模糊目標式的方法，將此目標式進行模糊，而模糊的對象為折現率（ω），然而在數學的計算上必須注意其正負號的使用方式，若在模糊化遇到負號時，要將最小化的模式轉換成最大化的模式，確保數學模式在計算上的正確性。

- 目標函數二：總專案工期最小化

$$Min\ f_2 = FT_{max} \qquad (18)$$

由於專案的各個活動的工期不在本研究的模糊範圍內，因此目標函數二將不進行模糊。

● 限制式一：前後製程的關係限制

$$FT_{iy} \geq FT_{ij} + t_{niy}, \forall j \in BO_{iy}; i = 1, \cdots, I; j = 1, \cdots, J_i; y = 1, \cdots, J_i; \quad (19)$$

$FT_{ij} \geq 0$且為整數$, i = 1, \cdots, I; j = 1, \cdots, J_i;$

同目標函數二的解釋，限制式一為計算專案活動工期的先後次序，因此不用將其模糊。

● 限制式二：資源限制

$$\sum_{i=1}^{I} \sum_{j=1}^{\boldsymbol{J_i-1}} \left(r_{ijk} + (1-h) \times r'\right) + Y_{ijt} \leq (R_k + (1-h) \times R'), k = 1, \cdots, K; t = 1, \cdots, H; \quad (20)$$

$$\sum_{i=1}^{I} \sum_{j=1}^{\boldsymbol{J_i-1}} \left(r_{ijk} - (1-h) \times r'\right) + Y_{ijt} \leq (R_k - (1-h) \times R'), k = 1, \cdots, K; t = 1, \cdots, H; \quad (21)$$

$$Y_{ijt} = \sum_{v=t}^{min\{t+t_{nij}-1, PL\}} X_{ijv},\ i = 1, \cdots, I; j = 1, \cdots, J_i - 1; t = ES_{ij}, \cdots, PL; \quad (22)$$

$$\sum_{v=ES_{ij}}^{PL} vX_{ijv} = FT_{ij}, i = 1, \cdots, I; j = 1, \cdots, J_i \quad (23)$$

$$\sum_{v=ES_{ij}}^{PL} X_{ijt} = 1, i = 1, \cdots, I; j = 1, \cdots, J_i \quad (24)$$

$X_{ijt}, Y_{ijt}, Z_i = 0\ or\ 1, i = 1, \cdots, I; j = 1, \cdots, J_i; t = 1, \cdots, PL;$

在資源限制的限制式裡，利用式（15）的方法，右式恆大於左式的方法，將原本的限制式一分為二，確保其限制式的合理性。

● 趕工模式:

$$OUC_{ij}=\frac{(c_{kij}-c_{nij})}{(t_{nij}-t_{kij})}\times rb_{ij}+c_{nij} \quad (25)$$

在趕工模式中，並沒有需要模糊的參數，但現金流出（OUC_{ij}），在計算時需要將其模糊化後來做運算，藉以使此參數可以更貼近真實的情況。

● 專案總價：

$$TPV_i^1=\left\{\sum_{j=1}^{J_i=1}(OUC_{ij}-(1-\mathrm{h})\times OUC')\times \exp((\omega-(1-\mathrm{h})\omega')\times(CP_i-EF_{ij}))\right\}\times(1+(\gamma_i-(1-\mathrm{h})\times\gamma')) \quad (26)$$

$$TPV_i^2=\left\{\sum_{j=1}^{J_i=1}(OUC_{ij}+(1-\hbar)\times OUC')\times exp((\omega+(1-\hbar)\times\omega')\times(CP_i-EF_{ij}))\right\}\times(1+(\gamma_i+(1-\hbar)\times\gamma')) \quad (27)$$

由於將現金流出（OUC_{ij}）及折現率（ω）模糊，利用式（15）的計算方式，將專案總價（TPV）計算成兩種不同的情況，一為TPV_i^1的模糊左界，而TPV_i^2則為模糊右界，此後使用到該參數，在依該式中的正負號來判斷該式適合使用何種情況。

● 各專案完工時的工程尾款現金流入：

$$INC_{iJ_i}^1 = TPV_i^1 - \sum_{j=1}^{\boldsymbol{J_i-1}} INC_{ij} \times exp\left(\left(\omega + (1-h) \times \omega'\right) \times (CP_i - EF_{ij})\right) \tag{28}$$

$$INC_{iJ_i}^2 = TPV_i^2 - \sum_{j=1}^{\boldsymbol{J_i-1}} INC_{ij} \times \exp\left(\left(\omega - (1-\mathrm{h}) \times \omega'\right) \times (CP_i - EF_{ij})\right) \tag{29}$$

各專案完成時的工程尾款（INC_{iJ_i}）是利用專案總價（TPV）減去專案各個活動的現金流入（INC_{ij}），因此專案完成的現金流入模糊左右界，就必須分別利用專案總價的模糊左右界來做計算。

● 獎金（G_i）與罰款（B_i）：

$$G_i = TPV_i^1 \times g \tag{30}$$

$$B_i = TPV_i^2 \times b \tag{31}$$

由於獎金是對於『目標函數一』的計算是加法，因此必須使用最悲觀的情況來做計算，因此利用專案總價的左界來做為獎金的計算值；而罰款則反之，對於『目標函數一』為減法，因此就必須要利用專案總價的右界來做計算。

以上為本研究模糊多目標非線性規劃模式的模式建立，利用此數學模式來進行專案排程，並根據不同程度的 α-截集的運算，比較出在不同程度下的 α-截集是否影響其排程的結果。

3.3 自組織適應懲罰策略（SOAPS）懲罰函數

本研究利用適應懲罰函數來將目標式及限制式進行模糊，在文獻中有提到許多適應懲罰函數的方法，其中以 Lin and Wu 在 2004 年所提出的自組織適應懲罰策略（SOAPS）懲罰函數為較好的適應懲罰函數，已有文獻比較出其在解決類似問題時，求解適應解的效率及求解的品質都較其他的適應懲罰函數要來的佳，因此本研究選擇了 SOAPS 法來做為本研究的適應懲罰函數。

Lin and Wu 提出了提出所謂的自組織適應懲罰策略（SOAPS）懲罰函數定義了常態化的目標函數值和常態化限制式違反的總合適應值，其公式如下：

$$F(\vec{x}) = f(\vec{x})/f_c + P(\vec{x},q)$$

$$P(\vec{x},q) = \frac{100+q}{100} \times \frac{1}{m} \times \sum_{j=1}^{m} r_j^q \times \frac{g_j(\vec{x})}{b_j},$$

where
$$r_j^q = \begin{cases} r_j^{q-1} \times \left[1 - \frac{\left(f_j^q - 0.5\right)}{5}\right], & and q \geq 1 \\ r_j^0 = \frac{QR_{\mathrm{obj}}^1}{QR_{\mathrm{con}j}^1}, & and q = 0 \end{cases} \tag{32}$$

其中r_j^q是懲罰因子，會隨著代數改變而有所不同，在初始代數時$r_j^0 = QR_{\mathrm{obj}}^1/QR_{\mathrm{con}j}^1$，其中$QR_{\mathrm{obj}}^1$為初始母體中目標函數值四分位差，而$QR_{\mathrm{con}j}^1$則代表初始母體中，第$j$個限制式的四分位差，這種方法主張最好的懲罰函數是藉由環繞最佳母體中所有個體的重心，當代數增加時，r_j^q也會有所改變。

f_j^q為第j個限制式的可行率，若在母體中，第j個限制式並沒有任何違反的情況出現，則可行率f_j^q為百分之百；若限制式都違反，可行率f_j^q則為零，這種方法可以監控每一代的每一個限制式的可行解率，並且引導母體至可行解或不可行解。

在限制式$g_j(\vec{x})$當中，$g_j(x) - b_j \leq 0$，此為一限制式的通式，其中$g_j(x)$

代表的是限制式的左式，而限制式的右式則是b_j，因此根據通式的計算，可以得知$g_j(x)/b_j \leq 1$，利用此方式將限制式和目標式合併。

m為限制式總數，（$100+q$）$/100$的計算方式可以有效的增強不可行解的懲罰，然而，這種懲罰方法被一固定值 f_c所影響，f_c為目標函數中最差的目標值，因此必須嘗試找出最合適的f_c值來做為模式的運算。

本研究利用 SOAPS 法來將目標式及限制式進行合併，其目的在於在尋找最佳化目標值的同時，限制式的限制因素也可以納入考量，而不是單純的計算目標函數，而且可以計算多個目標函數，只要設定好各個目標函數所佔的權重，就可以將多個目標函數合併計算，而本研究兩個目標函數的權重將各自設定為 0.5 來做計算。最後利用遺傳演算法來進行適應懲罰函數的求解。

3.4 基因遺傳演算法

本研究採用了遺傳演算法來進行數值比較，遺傳演算法的基本思路是以人工的方式模擬自然的演化過程。該演算法是基於套演化的方式來解決一組解，此解則稱之為母體。通常母體都是由隨機產生的，藉由遺傳運算子的演算，進而得到下一代母體，遺傳運算子包括選擇、交配和突變，母體經過了選擇，在經過交配和突變則可得到新的子代。在遺傳演算法中，可分為二進制編碼-遺傳演算法和實數編碼-遺傳演算法。由於遺傳演算法受到歡迎，且經常被使用，已經被證實可以應用在各種最佳化問題，並可以增加新的功能或結合其他演算法。

本研究在遺傳演算法的選擇階段，使用了競爭法，其方法是從母體隨機抽出個解，並比較其解的適應值，若為最大化問題則適應值取大，最小化問題則反之。比較之後，較好的解會被留下來，較差的則會被取代，直到得到新的母體，則進行下一個步驟：交配。在本研究中使用了啟發式交配法，其方法也是從母體抽取兩個解並進行比較，較佳的解會被留下，較差的則被取代，取代的方式如下：

$$\tilde{x}_3 = \lambda(\tilde{x}_2 - \tilde{x}_1) + \tilde{x}_2 \quad (32)$$

其中$\tilde{x}_2$為較佳的解，$\tilde{x}_1$為較差的解，其觀念是基於利用較佳的解$\tilde{x}_2$並找出更好的解，λ是介於 0 到 1 的隨機亂數。最後一個步驟是突變，本研究的方法是將每個母體解給予一個突變率，並對母體的每一個因子進行個別突變並且得到新的母體，重複上述作業直到基因代數達到最大代數則停止此遺傳演算法的程序。

本研究中所使的交配率有 50%，而突變率則設為 35%，但因為本研究中每個母體的因子有 22，故每個因子的突變率變 1.6%而總母體有 20 個，最大的基因代數有 500 代，而本研究使用 Matlab7.0 軟體來求解。

第四章 範例討論

4.1 問題背景

本研究的數學模式是在探討多專案的各個活動間的排程問題，利用此建立好的模式計算出專案內的排程順序，以下我們以範例說明並證明本研究的數學模式的可行性。

- 本範例考慮了三個專案，共二十七個作業，如下列三圖

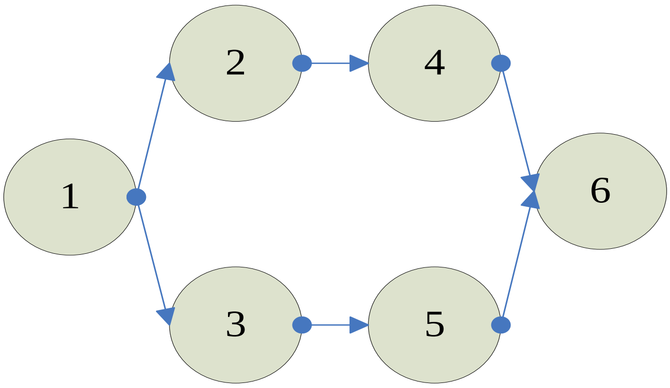

圖 4.1 專案一網路圖

資料來源：本研究整理

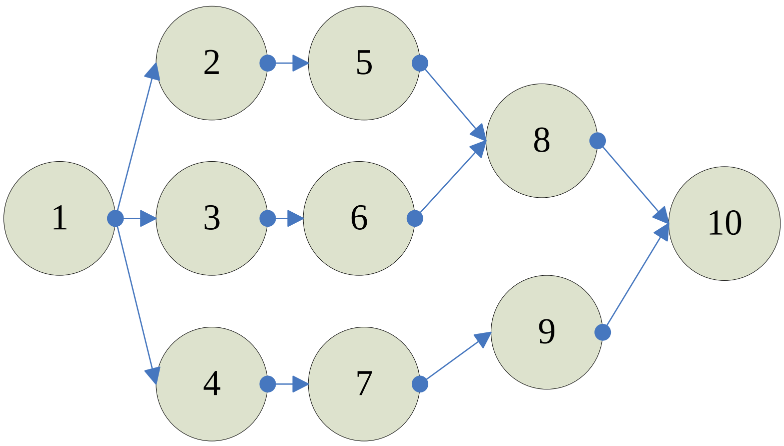

圖 4.2 專案二網路圖

資料來源：本研究整理

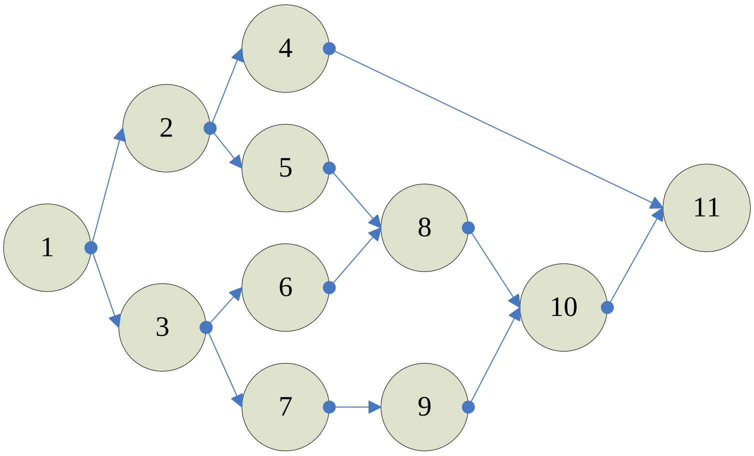

圖 4.3 專案三網路圖

資料來源：本研究整理

下列各表為各專案的各個作業所需輸入的參數值有，正常工期、趕工工期、正常成本、趕工成本、現金流入、無資源限制下最早開始作業時間、無資源限制下最晚完工作業時間及各個作業對於三種資源的需求使用量。如下列各表：

表 4.1 正常工期（天）

專案 \ 作業	1	2	3	4	5	6	7	8	9	10	11
1	5	4	3	6	5	5					
2	3	3	5	6	5	5	7	8	8	3	
3	3	4	5	6	7	7	5	5	4	6	5

資料來源：本研究整理

表 4.2 趕工工期（天）

專案 \ 作業	1	2	3	4	5	6	7	8	9	10	11
1	3	2	2	3	3	2					
2	2	1	4	5	4	3	4	7	6	2	
3	2	3	3	4	5	4	4	4	3	4	4

資料來源：本研究整理

表 4.3 正常成本（萬元）

專案＼作業	1	2	3	4	5	6	7	8	9	10	11
1	5	2	3	5	3	4					
2	3	4	4	4	2	5	3	3	4	5	
3	4	4	1	4	3	2	2	6	3	2	3

資料來源：本研究整理

表 4.4 趕工成本（萬元）

專案＼作業	1	2	3	4	5	6	7	8	9	10	11
1	7	5	6	8	4	5					
2	5	6	7	5	3	7	4	4	6	6	
3	7	7	3	5	5	4	3	8	6	6	4

資料來源：本研究整理

表 4.5 現金流入（萬元）

專案＼作業	1	2	3	4	5	6	7	8	9	10	11
1	10	6	8	9	6	7					
2	6	8	9	7	5	9	7	6	8	9	
3	9	8	4	6	7	6	5	9	8	9	6

資料來源：本研究整理

表 4.6 無資源限制下最早開始作業時間（天）

專案＼作業	1	2	3	4	5	6	7	8	9	10	11
1	0	5	5	9	8	15					
2	0	3	3	3	6	8	9	13	12	21	
3	0	3	3	7	7	8	8	15	13	19	25

資料來源：本研究整理

表 4.7 無資源限制下最晚完工作業時間（天）

作業 專案	1	2	3	4	5	6	7	8	9	10	11
1	5	9	10	15	15	20					
2	3	8	8	10	13	13	13	21	21	24	
3	3	8	8	25	15	15	15	19	19	25	30

資料來源：本研究整理

表 4.8 資源 A（使用量）

作業 專案	1	2	3	4	5	6	7	8	9	10	11
1	7	10	9	6	12	14					
2	8	5	13	9	14	15	16	18	19	5	
3	7	8	8	12	13	8	11	19	8	7	8

資料來源：本研究整理

表 4.9 資源 B（使用量）

作業 專案	1	2	3	4	5	6	7	8	9	10	11
1	8	6	11	10	8	15					
2	10	18	19	15	7	10	20	16	20	16	
3	7	12	8	15	10	12	12	17	22	15	10

資料來源：本研究整理

表 4.10 資源 H（使用量）

作業 專案	1	2	3	4	5	6	7	8	9	10	11
1	6	4	4	3	7	4					
2	5	9	7	6	8	6	10	12	7	7	
3	4	5	7	5	6	9	9	11	11	8	9

資料來源：本研究整理

以上資料為範例所給的資料，有了這些資料，我們就可以將這些資料帶入本研究的模式中，經過遺傳演算法的運算之後，我們可以得到專案的排程結果。而本研究有關遺傳演算法的參數設定如下表：

表 4.11 遺傳演算法參數

參數	
競爭大小	2
交配率	0.5
突變率	0.35
母體數	20
基因代數	500

資料來源：本研究整理

4.2 排程結果

經過遺傳演算法的計算後，我們可以分類成三種模式：參數未模糊模式、參數模糊 50%模式及參數模糊 20%模式，這三種模式來運用本研究的數學模式求解。所得到的解如下列各表：

表 4.12 未模糊模式之適應值、總淨現值及總工期

適應質	0.107849
Z1	1191.098
Z2	75

資料來源：本研究整理

表 4.13 未模糊模式之作業開工時點（單位：天）

作業 / 專案	1	2	3	4	5	6	7	8	9	10	11
1	1	13	9	50	25	72	0	0	0	0	0
2	14	35	27	33	38	42	49	47	62	74	0
3	1	13	20	32	14	29	30	56	68	72	75

資料來源：本研究整理

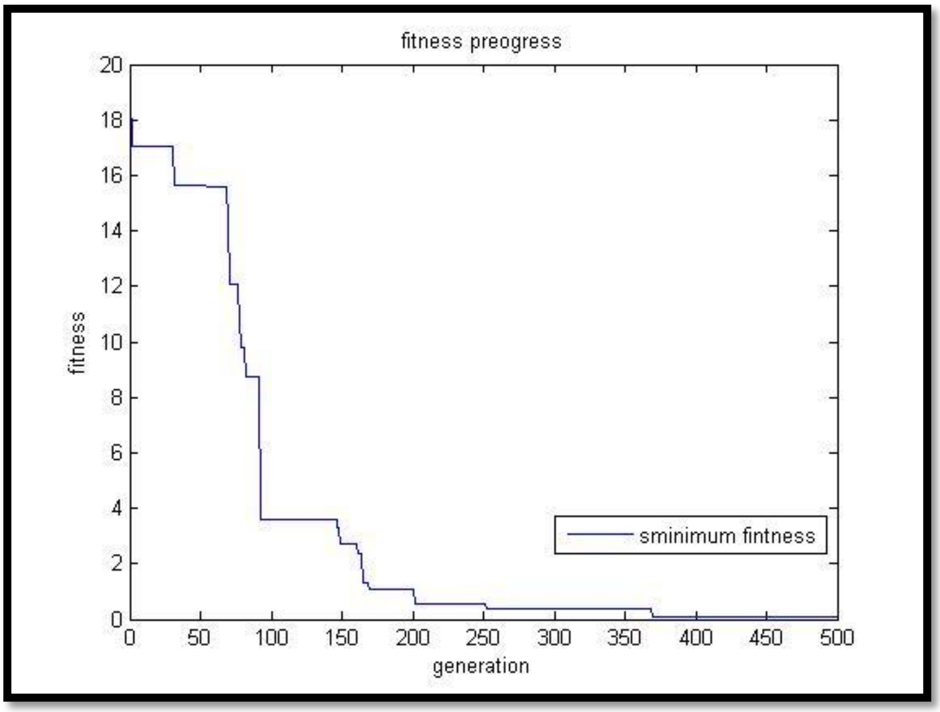

圖 4.4 基因演算法未模糊的迭代情況

資料來源：本研究整理

表 4.14 參數模糊 50%模式之適應值、總淨現值及總工期

適應質	0.107704
Z1	578.1983
Z2	76

資料來源：本研究整理

表 4.15 參數模糊 50%模式之作業開工時點（單位：天）

作業 / 專案	1	2	3	4	5	6	7	8	9	10	11
1	9	55	22	60	40	69	0	0	0	0	0
2	6	26	28	11	50	37	22	45	55	66	0
3	6	42	9	60	51	48	15	59	31	64	76

資料來源：本研究整理

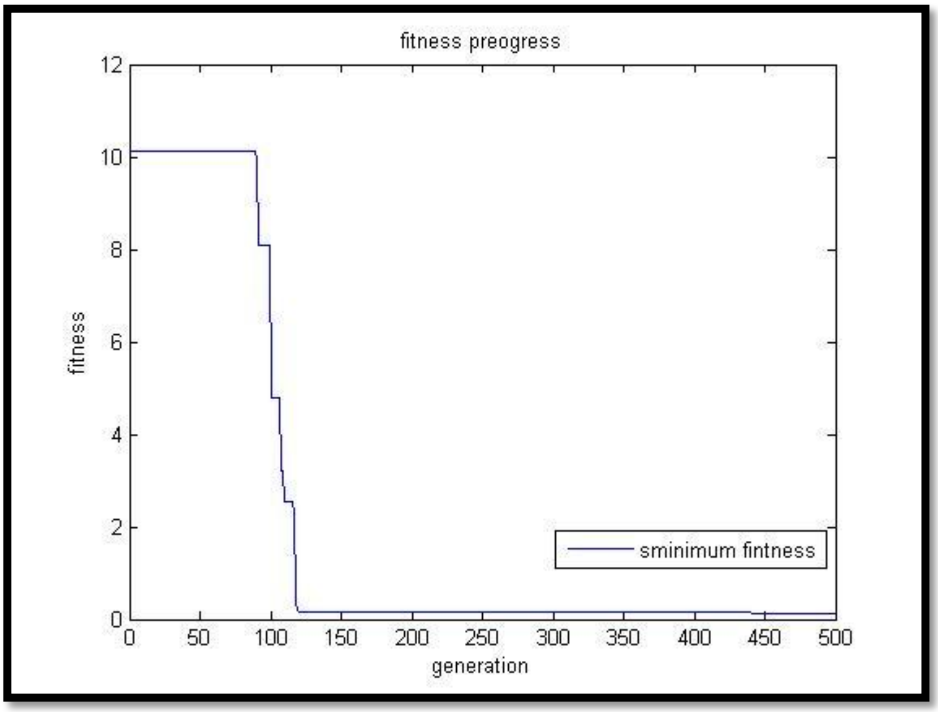

圖 4.5 基因演算法在模糊值 0.5 的迭代情況

資料來源：本研究整理

表 4.16 參數模糊 20%模式之適應值、總淨現值及總工期

適應質	0.109207
Z1	319.9379
Z2	75

資料來源：本研究整理

表 4.17 參數模糊 20%模式之作業開工時點（單位：天）

作業 / 專案	1	2	3	4	5	6	7	8	9	10	11
1	12	17	52	32	57	54	0	0	0	0	0
2	12	31	18	15	43	53	32	67	67	75	0
3	7	12	25	17	27	44	47	54	59	66	69

資料來源：本研究整理

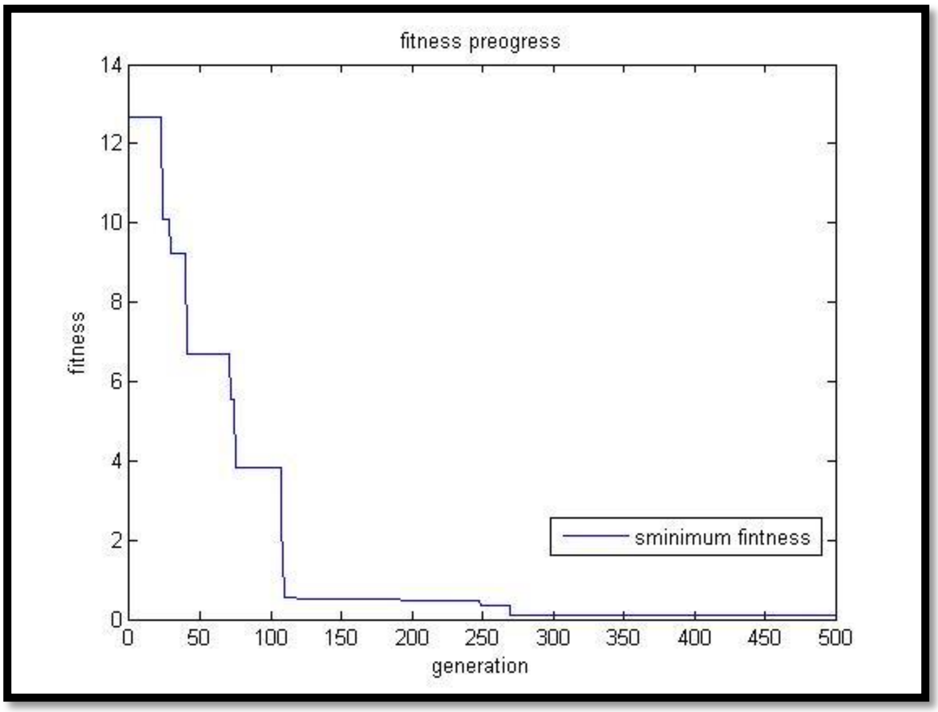

圖 4.6 基因演算法在模糊值 0.2 的迭代情況

資料來源：本研究整理

由以上解可以得知，本研究的數學模式適用於各個種不同的模糊數，並且都可以得到合理的解，固驗證了此模式是可以被運用在求解最大化總專案淨現值及最小化總專案工期的問題。

第五章 結論與建議

5.1 研究成果

本研究的目的是為了建構一模糊多目標多專案非線性規劃模式，並結合自組織適應懲罰策略懲罰函數，將目標式及限制式合併成一個適應函數，並交由遺傳基因演算法來求解。而本研究的研究成果就是此模式可以用在各種專案排程管理的問題上，可以調整其專案或活動的數目，並且適用於各種不同的量性資源，可以根據專案的需求來確定哪些資源需要被限制，由於專案數及活動數可以增加或減少，因此可以處裡各種大型的專案排程。不同於別的專案排程模式，本研究模式加入了趕工決策、資源限制、非線性規劃、模糊數學及多目標多準則規劃。在排程上，可以適應各種不同的專案型態，也可以增加或減少資源的限制種類，可以因應不同的專案問題。

5.2 未來研究建議

以下幾點為研究方向之建議：

1. 利用不同演算法，例如：蜂群演算法、螞蟻演算法等，取代基因演算法來進行演化，並比對其排程結果，求解效率等比較性研究。
2. 使用不同的懲罰函數，例如：SOAP-Ⅱ法、TAPA法等並比對其排程結果和求解效率。

參考文獻

中文文獻

邱煥能、蔡登茂（1993）。有限資源多專案排程單專案法與多專案法之比較研究。**中國工業工程學刊**，第十卷，第三期，第171-179頁。

陳弘翊（1999）。**多重專案規劃與控制:系統動態分析法暨流程建模技術的應用**（碩士論文）。國立中央大學，桃園縣。

陳旭明，(1997)。**有限資源最大化淨現值專案排程問題之研究**（碩士論文）。國立台灣科技大學，台北市。

詹蕙珍（2004）。**模糊多目標非線性規劃在有限資源多專案排成問題之應用**（碩士論文）。屏東科技大學，屏東縣。

英文文獻

Ayyub, B. M. and Halder, A. (1984). Project scheduling using fuzzy set concepts. *J. Constr. Engrg. and Mgmt., ASCE, 110*(2), 189-203.

Abbasi, G. Y., and Arabiat, Y. A. (2001). A heuristic to maximize the net present value for resource-constrained project-scheduling problems. *Project Management Journal*, *32*(2), 17-24.

Arikan, F. and Gungor, Z. (2001). An application of fuzzy goal programming to a multi objective project network problem. *Fuzzy sets and Sys., 119*, 49-58.

Abadi, S. N., Aghassi, H., and Roghanian, E. (2011). A multi-attribute GA for piecewise linear Time-Cost Trade-off Scheduling optimization. *Information and Communication Technologies*, 1187-1192.

Baker, J. E. (1985). Adaptive selection methods for genetic algorithms. *J. J. Grefenstette (Ed.), Proceedings of the First International Conference on Genetic Algorithms*, 101-111.

Chiu, H. N., and Tsai, D. M. (2002). An efficient search procedure for the resource-constrained multi-project scheduling problem with discounted cash flows. *Construction Management and Economics, 20*(1), 55-66.

Chanas, S. and Kamburowski, J. (1981). The use of fuzzy variables in PERT. *Fuzzy sets and Sys, 5,* 11-19.

Chen, S. M. and Chang, T. H. (2001). Finding multiple possible critical paths using Fuzzy PERT. *IEEE Trans. on Sys., Man, and Cyb. Part B, 31*(6), 930-937.

Chang, P.-T., and Lo, Y.-T. (2001). Modeling of job-shop scheduling with multiple quantitative and qualitative objectives and a GA/TS mixture approach. *International Journal of Computer Integrated Manufacturing, 14*, 367-384.

Chen, J.-S., Pan, J. C.-H., and Lin, C.-M. (2008). A hybrid genetic algorithm for the re-entrant flow-shop scheduling problem. *Expert Systems with Applications, 34*, 570-577.

Deb, K., Anand, A., and Joshi, D. (2002). A computationally efficient evolutionary algorithm for real-parameter optimization. *Evolutionary Computation*, *10*(4), 371-395.

De Jong, K. A. (1975). An analysis of the behavior of a class of genetic adaptive systems. Ph.D. Dissertation, University of Michigan. *Dissertation Abstracts International, 36*(10), 5140B.

Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering, 186*(2), 311-338.

Dubois, D. and Prade, H. (1997). The three semantics of fuzzy sets. *Fuzzy Sets and Systems, 49(*1), 65-74.

Doersch, R. H., and Patterson, J. H. (1977). Scheduling a project to maximize its present value: a zore-one programming approach. *Management Scienc, 23*(8), 882-889.

Eiben, A. E., Hinterding, R., and Michalewicz, Z. (1999). *Parameter control in evolutionary algorithms. IEEE Transactions on Evolutionary Computation*, *3*(2), 124-141.

Elliott, L., Ingham, D. B., Kyne, A. G., Mera, N. S., Pourkashanian, M. and Wilson, C. W. (2004). Genetic algorithms for optimization of chemical kinetics reaction mechanisms. P*rogress in Energy and Combustion Science, 30*, 297-328.

Farmani, R. and Wright, J.A. (2003). Self-adaptive fitness formulation for constrained optimization. *IEEE Transactions on Evolutionary*

Computation, *7*(5) 445-455.

Gen, M. and Cheng, R. (1996). Interval programming using genetic algorithms. *Proceedings of the Sixth International Symposium on Robotics and Manufacturing.*

Grefenstette, J. J. (1986). Optimized control of parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics, 16,* 122-128.

Georgiou, S. D. (2007). New two-variable full orthogonal designs and related experiments with linear regression models. *Statistics and Probability Letters*, *77*, 25-31.

Hansen, N., and Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation, 9*(2), 159-195.

Holland, J. H. (1975). Adaptation in natural and artificial systems. *Ann Arbor, MI: University of Michigan Press; Extended new Edition, 1992. Cambridge: MIT Press.*

Hu, Y.-B., Wang, Y-P. and Guo, F.-Y. (2005). A new penalty based genetic algorithm for constrained optimization problems. *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*, 18-21.

Hoffmeister, F. and Sprave, J. (1996). Problem-independent handling of constraints by use of metric penalty functions. *Proceedings of the Fifth Annual Conference on Evolutionary Programming*, 289-294.

Hsu, G. J. Y. and Tzeng, Y. R. (1990). A new algorithm of fuzzy multiobjective programming: the compromise factor approach. *Proceedings: The First International Symposium on Uncertainty Modeling and Analysis*. *Maryland University, Dec*, 3-5.

Hannan, E. L. (1981). Linear programming with multiple fuzzy goals. *Fuzzy Sets and Systems*, *6*, 235-248.

Janikow, C. Z., and Michalewicz, Z. (1991). An experimental comparison of binary and floating point representation in genetic algorithms. *Proceedings of the Fourth International Conference on Genetic Algorithms*, 31-36.

Kerzner, H. (2001). Project Management: A system Approach to Planning, Scheduling, and Controlling. *Seventh edition. New York: Van Nostrand Reinhold.*

Knight, R. M. (1966). Resource allocation and multi-project scheduling in a research and development environment. *Unpublished M.S. Thesis. Massachusetts Institute of technology.*

Kurtulus, I., and Davis, E. W. (1982). *Multi-project scheduling: categorization of heuristic rules performance. Management Science, 28*(2), 161-172.

Kurtulus, I., and Narula, S. C. (1986). Multi-project scheduling: analysis of project performance. *IIE Transactions, 17*(1), 58-66.

Kaufmann, A. and Gupta, M. M. (1988). Fuzzy Mathematical Models in Engineering and Management Science. *New York: Elsevier Science Publishing Company Inc.*

Kiriklidis, C., Kirytopoulos, K., and Rokou, E. (2011). Exploring close-optimal solutions for the Time Constrained Scheduling Prolem in project management. *Industrial Engineering and Engineering Management,* 844-847.

Liu, J. B., Hong, R., and Xie, J. H. (2008). Critical Chain Project Management Based Heuristic Algorithm for Multiple Resources-Constrained Project. *Business and Information Management, 1,* 330-335.

Leite, J. P. B. and Topping, B. V. T. (1998). Improved genetic operators for structural, engineering optimization. *Advances in Engineering Software, 29,* 529–62.

Lobo, F. G. (2000). The parameter-less genetic algorithm: Rational and automated parameter selection for simplified genetic algorithm operation. Ph.D. Thesis, University of Lisbon, Portugal.

Lin, C.-Y. and Wu, W.-H. (2004). Self-organizing adaptive penalty strategy in constrained genetic search. *Structural and Multidisciplinary Optimization, 26*(6), 417-428.

Leu, S. S., Chen, A. T. and Yang, C. H. (2001). A GA-based fuzzy optimal model for construction time-cost trade-off. *Int. J. Project Mgmt., 19,* 47-58.

Liu, S. T. (2003). Fuzzy activity times in critical path and project crashing problems. *Cyb. and Sys., 34*, 161-172.

Lorterpong, P. and Moselhi, O. (1996). Project-network analysis using fuzzy sets theory. *J. Constr. Engrg. and Mgmt., ASCE, 122*(4), 308-318.

Luhandjula, M. K, (1982). Compensatory operators in fuzzy programming withmultiple objectives. *Fuzzy Sets and Systems, 8,* 245-252.

Leberling, H. (1981). On finding compromise solutions in multicriteria problems using the fuzzy min-operator. *Fuzzy Sets and Systems, 6,* 108-118.

Lova, A., Maroto, C. and Tormos, P. (2000). A multicriteria heuristic method to improve resource allocation in multiproject scheduling. *European Journal of Operational Research, 127,* 408-424.

Meredith, J.R. and Mantel, S.J. Jr. (2000). Project Management: A Managerial Approach. Fourth edition. *New York: John Wiley and Sons.*

Mize, J. H.(1964). A heuristic scheduling model for milti-project organization. Unpublished Ph.D. Thesis. Purdue University.

Mon, D. L., Cheng, C. H. and Lu, H. C. (1995). Application of fuzzy distributions on project management. *Fuzzy sets and Sys., 73,* 227-234.

Morales, A. K. and Quezada, C.V. (1998). A universal eclectic genetic algorithm for constrained optimization. *Proceedings of the 6th European Congress on Intelligent Techniques and Soft Computing,* 518-522.

Michalewicz, Z. (1992). Genetic algorithms + data structures = evolution programs. *1st Edition. Berlin: Springer-Verlag. 2nd Edition, 1994. 3rd Edition, 1996.*

Michalewicz, Z., Logan, T., and Swaminathan, S. (1994). Evolutionary operators for continuous convex parameter spaces. *Proceedings of the Third Annual Conference on Evolutionary Programming*, 84-97.

Myers, R., and Hancock, E. R. (1997). Genetic algorithm parameter sets for line labeling. *Pattern Recognition Letters, 18,* 1363-1371.

Mayer, D. G., Belward, J. A., and Burrage, K. (2001). Robust parameter settings of evolutionary algorithms for the optimisation of agricultural systems models. *Agricultural Systems, 69,* 199-213.

Murata, T., Ishibuchi, H., and Tanaka, H. (1996). Genetic algorithms for flows shop scheduling problems. *Computers and Industrial Engineering, 30,* 1061–1071.

Mizumoto, M. and Tanaka, K. (1976). The four operations of arithmetic on fuzzy numbers. *Systems-Computers-Controls*, *7*, 73-81.

Nasution, S. H. (1994). Fuzzy critical path method. *IEEE Trans. on Sys., Man, and Cyb, 24*(1), 48-57.

Ozdamar, L., and Dundar, H. (1997).A flexible heuristic for a multi-mode capital constrained project scheduling problem with probabilistic cash inflows. *Computers and Operations Research*, *24*, 1187-1200.

Puzzi, S. and Carpinteri, A. (2008). A double-multiplicative dynamic penalty approach for constrained evolutionary optimization. *Structural and Multidisciplinary Optimization, 35*(5), 431-445.

Patterson, J. H. (1973). Alternative methods of project scheduling with limited resources. *Naval Research Logistics Quarterly*, *20*(4), 767-784.

Patterson, J. H. (1976). Project scheduling: the effects of problem structure on heuristic performance. *Naval Research Logistics Quarterly, 23*(1), 95-122.

Russell, R. A. (1986). A comparison of heuristics for scheduling projects with cash flows and resource restrictions. *Management Science, 32*(10), 1291-1300.

Sakawa, M. (1988). An interactive fuzzy satisficing method for multiobjective linear fractional programming problems. *Fuzzy Sets and Systems*, *28,* 129-144.

Schaffer, J. D., Caruna, R. A., Eshelman, L. J., and Das, R. (1989). A study of control parameters affecting online performance of genetic algorithms for function optimization. *Proceedings of Third International Conference on Genetic Algorithm,* 51-60.

Schmitt, L. M. (2001). Theory of genetic algorithms. *Theoretical Computer Science, 259,* 1-61.

Suwa, H., Morita,D., and Sandoh, H. (2010). A new framework of critical chain scheduling in project management. *Engineering Systems Management and Its Applications,* 1-6.

Smith-Daniels, D. E., Padman, R., and Smith-Daniels, V.L. (1996). Heuristic scheduling of capital constrained projects. *Journal of operations*

Management, 14, 241-254.

Tian, Z., Zhang, Z. and Pen, W. (2012). A Critical Chain Based Multi-project Management Plan Scheduling Method. *Industrial and Information System, 2010 2sd*, 304-308.

Tessema, B. and Yen, G.G. (2009). An adaptive penalty formulation for constrained evolutionary optimization. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 39*(3), 565-578.

Todoroki, A., and Ishikawa, T. (2004). Design of experiments for stacking sequence optimizations with genetic algorithms using response surface approximation. *Composite Structures, 64*, 349-357.

Tsai, D. M., and Chiu, H. N. (1996). Two heuristics for scheduling multiple projects with resource constraints. *Construction Management and Economics, 14*(6), 325-340.

Vose, M. D. (1999). The simple genetic algorithm: Foundations and theory. *Cambridge, MA: MIT Press.*

Wu, W.-H. and Lin, C.-Y. (2004). The second generation of self-organizing adaptive penalty strategy for constrained genetic search. *Advances in Engineering Software, 35*(2), 815-825.

Wang, Y., Cai, Z., Zhou, Y. and Fan, Z. (2009). Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. *Structural and Multidisciplinary Optimization, 37*(4), 395-413.

Wang, W., Bai, Y., Cai, C., and Yan, X. (2010). Application of Resource-Constrained Project Scheduling with a Critical Chain Method on organizational Project Management. *Computer Design and Applications, 2*, 51-54.

Yao, J. S. and Lin, F. T. (2000). Fuzzy critical path method based on signed distance ranking of fuzzy numbers. *IEEE Trans. on Sys., Man, and Cyb. Part A, 30*(1), 76-82.

Yager, R. R. (1981). A procedure for ordering fuzzy subsets of the unit interval. *Information science, 24*, 143-161.

Yokota, T., Gen, M., Ida, K. and Taguchi, T. (1996). Optimal design of system reliability by an improved genetic algorithm. *Electronics and*

Communications in Japan, *79*(2) 41-51.

Zimmermann, H. J. (1976). Description and optimization of fuzzy systems. *International Journal of General Systems*, *2*, 209-215.

Zimmermann, H. J. (1978). Fuzzy programming and linear programming with several objective functions. *Fuzzy Sets and Systems, 1,* 45-56.