

# 行政院國家科學委員會補助專題研究計畫成果報告

## 以基因演算法構建一般化軟體可靠度成長模型之研究

計畫類別：C 個別型計畫          整合型計畫

計畫編號：NSC 89 - 2416 - H - 032 - 047 -

執行期間：89 年 08 月 01 日至 90 年 07 月 31 日

計畫主持人：徐煥智

共同主持人：

本成果報告包括以下應繳交之附件：

赴國外出差或研習心得報告一份

赴大陸地區出差或研習心得報告一份

出席國際學術會議心得報告及發表之論文各一份

國際合作研究計畫國外研究報告書一份

執行單位：淡江大學資訊管理學系

中 華 民 國 90 年 9 月 10 日

# 行政院國家科學委員會專題研究計畫成果報告

## 國科會專題研究計畫成果報告撰寫格式說明

### Preparation of NSC Project Reports

計畫編號：NSC 89-2416-H-032-047

執行期限：89年8月1日至90年7月31日

主持人：徐煥智 淡江大學資訊管理系

計畫參與人員：廖建賀 淡江大學資訊管理系

#### 一、中文摘要

為提昇軟體之品質及提供一軟體品質的衡量指標，軟體可靠度在軟體生命週期中的許多方面都扮演著一個重要的角色。目前已有許多得的軟體可靠度成長模式根據非齊次波氏過程的假設來建構並且在軟體可靠度工程的實務上得到很高的評價，其基本之原理在於假設軟體失效的累計個數是符合非齊次波氏過程，因此只要能提出其平均值函數(mean value function)即能預測出軟體的可靠度。在此模式中有許多的參數值需要被估算，尤其在本研究中所探討之一般化的軟體可靠度成長模式，此模式考量了非完美軟體測試及轉折點的問題，所需估計的參數更形繁複。本研究中，我們考量如何利用基因演算法來估計一般化之軟體可靠度成長模式中各參數的預測值以建構更精確的軟體可靠度預測方法。

**關鍵詞：**軟體可靠度成長模式，基因演算法，非齊次波氏過程，轉折點，非完美軟體測試

#### Abstract

Software reliability growth models are derived to analyze the software reliability by using the failure data collected from software development projects. These models based on the nonhomogeneous Poisson process (NHPP) have been quite successful tools in practical software reliability engineering. Software reliability

can be estimated once the mean value function of the NHPP is determined. Model parameters are usually estimated using either the maximum likelihood estimation (MLE) or least-square estimate. For a general NHPP SRGM that incorporates with both imperfect debugging and change-point problem, lot of parameters need to be estimated at the same time. Parameters estimation has been a new challenge of the models development. Genetic algorithm appears a very promising technique to provide more accurate estimates of these parameters. In this study, we apply the modern genetic algorithm concept such as arithmetic crossover and nonuniform mutation. in our model development process.

**Keywords:** Software Reliability Growth Model, Genetic Algorithm, Nonhomogeneous Poisson Process, Change-Point, Imperfect Debugging

#### 二、緣由與目的

近年來愈來愈多的電腦被應用於諸如醫療、航空、電信、軍事、及核能等重要的領域上。因此軟體的品質愈來愈受到重視，高品質的軟體的需求也愈來愈大。然而目前許多的軟體由於隱含著許多未被偵測出來的錯誤而導致軟體執行時的沒有效率。為提昇軟體之品質及提供一軟體品質

的衡量指標，軟體可靠度在軟體的生命週期中的許多方面都扮演著一個重要的角色。例如，在軟體實際上線使用或轉交給客戶之前，正確的軟體可靠度預測在確保軟體的品質上是必須的，軟體可靠度的預測亦是決定軟體所需測試時間及何時可結束測試的重要決策依據。

有許多的原因會導致軟體在執行時失效，但沒有任何的失效是由於如硬體般的磨損等原因。通常軟體的失效是由於設計上的問題，如不了解需求，程式撰寫錯誤，方法錯誤，或執行環境改變等。在過去的 25 年間有許多的統計模式被提出來解決軟體可靠度預測的問題 [Goel, Okumoto, 1979][Yamada et al., 1983][Pham, 1993][Pham, 1996] 利用非齊次波氏過程所構建之軟體可靠度成長模式 (NHPP SRGMs) 已被證明在實務上的應用已是非常的成功 [Musa et al., 1987] 不同的 NHPP SRGMs 是建構在不同的假設上。傳統之 NHPP SRGMs 大部份均假設在任何時間一個軟體的失效，其失效的原因都能立即被找出來且導致失效的錯誤能被立即的更正。在除錯的過程中也不會因為程式的更動而導致新的錯誤發生。我們稱此假設為完美的除錯過程。另外一個假設被許多相關模式引用的是每一失效發生的間隔時間長度為一獨立且符合相同機率分配的隨機變數。然而在實際的軟體錯誤偵測過程中，很可能因為在除錯過程中程式的修改而導致新的錯誤發生。又由於軟體的測試環境及策略的改變，測試人員的異動，測試資料的變動等因素皆可能使得失效發生的時間可能無法具有相同的機率分配，即在收集資料上有轉折點的現象發生 [Zhao, 1993]。

本人在 88 年度國科會計劃中已提出一新得 NHPP SRGM 軟體可靠度預測模型，此模型已考量了軟體除錯過程是屬於非完美除錯且允許在測試過程中允許有轉折點的情形存在。此模式在初步的驗證下顯示出比目前所提出之 NHPP SRGMs 有更佳的軟體可靠度預測結果，尤其在具有轉折點資料的驗證上更是具有其它方法所無法達成

的預測結果。然而此一較一般化之模型所需考量的變數較多，使用一般的參數預估方法，在實用上軟體的管理者需對本模式作一較深入的了解才能在參數的預估上得到較好的結果，在目前之研究中亦發現，以傳統之 nonlinear programming 方法預估模型參數值，參數的初始值設定往往會影響整體預測結果的精確性，因此目前在建構模型過程中常需要設定數組不同之參數初始值用以比較選取一較佳之預測模型。此過程對使用者而言並非一好的解決方案。人工智慧的先趨者 John Holland [1975, 1992] 提出了動物演化的過程此種高度平行處理技術可被應用來解決各式各樣的問題本研究希望能藉由先進的基因演算法在最佳解搜尋的能力上，可提供一解決此項缺點的有效方法，並能提升軟體可靠度之預測精確性。使得軟體可靠度的研究不僅在觀念上能得以推廣，並可提高軟體可靠度工具的使用比例，更能促使國內軟體的發展重視其品質的提升，以達成國際競爭的能力。

### 三、結果與討論

本研究以本人所提出之一般化非齊次波氏過程之軟體可靠度成長模型為基礎，因為此模式是目前唯一可同時考量軟體除錯是屬於非完美除錯且允許測試資料具有轉折點的一個軟體可靠度成長預測模式。先前之研究結果顯示其在軟體可靠度的預測上較目前現有之 NHPP SRGMs 有更精確的預測結果。此軟體可靠度預測模型之累計軟體失效次數的平均值函數 (mean value function) 方程式為

$$m(t) = \begin{cases} \frac{a}{1-S_1} [1 - e^{-(1-S_1)\beta t}] & \text{when } 0 \leq t \leq \tau \\ \frac{a}{1-S_2} [1 - e^{-(1-S_1)\beta \tau - (1-S_2)\beta (t-\tau)}] + \frac{m(\tau)(S_1 - S_2)}{1 - S_2} & \text{when } t > \tau \end{cases}$$

。式中 a 為系統發展完後之原始錯誤個數，β 為除錯過程中的錯誤產生率，b 為每一錯誤之錯誤偵測率，τ 為轉折點，t 為時間。由此一函數我們可以輕易的展出相關具條件式的軟體可靠度預測模式

$$R(x|t) = \exp\{-m(t+x) - m(t)\}$$

$$= \begin{cases} \exp\left\{-\frac{a}{1-S_1} (e^{-(1-S_1)b_1 t} - e^{-(1-S_1)b_1(t+\tau)})\right\} & \text{when } 0 < t \leq \tau \\ \exp\left\{-\frac{a}{1-S_2} (e^{-(1-S_1)b_1 t} - e^{-(1-S_2)b_2(t-\tau)})(1 - e^{-(1-S_2)b_2 \tau})\right\} & \text{when } t > \tau \end{cases}$$

在之前的模式中我們僅假設所有的  $a$  值與  $b$  值是未知值而錯誤產生率  $\beta$  與轉折點  $\tau$  為已知，而在本研究中已放寬此一限制，即令  $\beta$  與  $\tau$  亦為未知值。為估計這些參數首先需收集測試資料，並利用最大似估計法或最小平方法來估計之，為避免在估計的過程中產生錯誤而導致可靠度預測精準度的降低與繁雜的步驟，本研究以基因演算法(Genetic Algorithm) 作為估計參數的工具。基因遺傳演算法是一種模擬生物基因演化的 heuristic 求解模型。此新的方法在複雜的最佳化求解問題中展現了富潛力的能力(Goldberg 1989)。在這種模型中，問題的解答被適當的安排成一串數值的組合，用以模擬基因中的一串染色體基因。大量的基因經過突變(mutation)，交配(crossover)，與複製(reproduction)等運算過程不斷地產生新的子代基因，且不良的基因在篩選過程中將逐漸被淘汰。最後演化出所求問題的最佳解答。使用基因演算法必需定義出相關的 Fitness 函數方程式，作為子代挑選的依據。本研究所使用的 Fitness 函數方程式為一概似函數，且包含其 penalty function 以避免不可行解的產生。令  $S$  代表所有參數的集合：

$$f(S) = L(a, b_1, b_2, S_1, S_2, \tau) = \prod_{j=1}^n \frac{[m(t_j) - m(t_{j-1})]^{y(t_j) - y(t_{j-1})} \cdot e^{-[m(t_j) - m(t_{j-1})]}}{(y(t_j) - y(t_{j-1}))!}$$

式中  $y(t_j)$  為在測試時間  $t_j$  前之累計 Failure 發生次數， $n$  是觀測總次數。

$$F(S) = f(S) - p(S).$$

penalty function  $p(S)=0$  當以下條件成立時否則為一很大的數目字

$$a \geq 0$$

$$0 \leq b_1, b_2 \leq 1$$

$$0 \leq S_1, S_2$$

為保此研究的可行性，我們應用了已

成功地被使用的 genetic operators 如 Gorberg (1989) 的 roulette wheel selection, Michalewicz (1994) 的 arithmetic crossover 及 Gen 和 Cheng (1997) 的 non-uniform mutation. 此外我們設定了不同的 population size, crossover rate 與 mutation rate. 以瞭解以上設定值對本模型求解的影響。在本研究中所設定的最多子代數為 10000, 若在連續 1000 個子代內所估計最佳解的值變化小於 1% 亦停止搜尋。

實作中是以 Musa [1979] 對 T1 系統測試所得到的資料為研究對象，其中共有 136 筆記錄，我們將其中的 130 筆記錄作為 Modeling Data 用以執行基因演算法估計參數以取得最佳解。並利用最後 6 筆記錄作為 Testing Data，以瞭解此模型的實際預測能力。模型與預測能力的好壞是以 MSE 值作為評斷依據。MSE 值是根據  $m(t)$  與  $y(t)$  值的差異計算而得，愈小代表估計值與實際值誤差愈小。在研究中有以下之發現：

1. 使用基因演算法，在不同參數設定下均能快速收斂至接近最佳解之可行解。
2. Population size 的大小，在相同條件下對於模型的最終結果(預測能力)並無明顯之關係存在。見表一

Population Size	MSE-Modeling	MSE-Prediction
50	10.14	10.90
60	9.15	10.56
70	9.78	5.92
80	10.10	11.84
90	9.13	10.35
100	8.28	10.95
200	8.50	4.09
500	9.07	12.42
1000	9.63	7.28

Crossover rate: 0.5, Mutation rate: 0.1

表一

3. Crossover rate 的大小，在其餘條件相同下與模型的最終結果並無明顯之關係存在。見表二。

Crossover Rate	MSE-Modeling	MSE-Prediction
----------------	--------------	----------------

0.2	8.25	8.47
0.3	8.55	5.65
0.4	8.25	8.14
0.5	8.50	4.09
0.6	8.54	5.74
Population size:200, Mutation Rate: 0.1		

表二

4. Crossover rate 的大小，在其餘條件相同下與模型的最終結果並無明顯之關係存在。見表三。

Mutation Rate	MSE-Modeling	MSE-Prediction
0.05	8.91	8.884
0.1	8.50	4.09
0.2	9.08	12.73
0.25	9.36	7.39
Population Size: 200, Crossover Rate: 0.5		

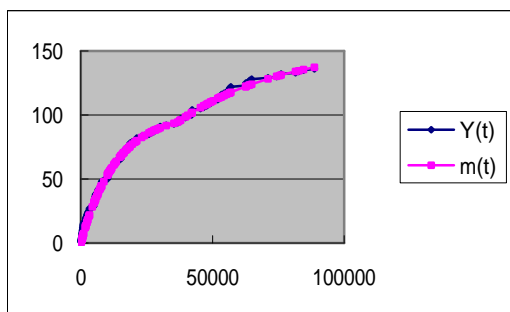
表三

5. 若放寬最多子代數的限制，可得到相當精確的軟體可靠度預估值。本研究所得之最佳解為當  $a=91$ ,  $b_1=8.25E-5$ ,  $b_2=2.43E-4$ ,  $\beta_1=0.099$ ,  $\beta_2=0.8988$ ,  $\tau=36453$ 。其 MSE 值分別如下：

MSE-Modeling = 8.01,

MSE-Prediction = 0.74。

此最佳解的取得為當最大子代數設為 30000, Crossover Rate = 0.5, Mutation Rate = 0.1, Population Size = 200。圖一顯示最終之結果。此結果的表現較傳統作業研究方法所得之模型提供了更精確的預測。



圖一

#### 四、計劃結果自評

本研究提供目前所有軟體可靠度模型建立之新方法，並使得軟體可靠度的預測能藉由新的參數預估方法而達到更精確的地

步。本研究已完成

- (1) 求解一般化之軟體成長模式之參數方程式的建立
- (2) 基因演算法於估計軟體成長模式之各參數的可行性分析
- (3) 基因演算法於估計軟體成長模式中參數的敏感度分析

傳統作業研究方法所帶來的限制，使的在構建軟體可靠度時因為相關係數之估計不易與不正確，而導至整個軟體可靠度的預估並非如此的精確，使的在軟體的發展過程中的許多決策發生了錯誤。基因演算法已經應用在許多不同的領域，如作業排程，市場分析，物件辨識等，而本研究結果顯示基因演算法應用於軟體可靠度之預估上亦可產生良好的結果。

#### 五、參考文獻

- Goel, Amrit L. and Okumoto, Kazu (1979), "Time-Dependent Error Detection Rate Model for Software Reliability and Other Performance Measures," IEEE Trans. On Reliability, Vol. R-28, No.3, pp. 206-211
- Yamada, S., Ohba, M., and Osaki, S. (1983), "S-Shaped Reliability Growth Modeling for Software Error Detection," IEEE Trans. On Reliability, Vol R-32, PP. 428-443
- Pham, H. (1993), "Software Reliability Assessment: Imperfect Debugging and Multiple Failure Types in Software Development," EG&G-RAAM-10737, Idaho National Engineering Laboratory
- Pkam, H. (1996), "A Software Cost Model with Imperfect Debugging, Random Life Cycle and Penalty Cost," Int'l J. Systems Science, Vol. 27, No. 5, pp. 455-463
- Musa, J.D. (1979), "Software Reliability Measures Applied to System Engineering," Proceedings COMPCON
- Musa, J.D., Iannino, A. and Okumoto, K. (1987), "Software Reliability: Measurement, Prediction, Application," McGrawHill Book Company, New York, NY
- Zhao, M. (1993), "Change-point problems in software and hardware reliability," Commun. Statist.-Theory Meth., Vol. 22, pp. 757-768
- Holland, J.H. (1975), "Adaptation in Natural and Artificial Systems," Ann Arbor, MI:

University of Michigan Press

Goldberg, D. (1989), "Genetic Algorithms in Search, Optimization, and Machine Learning," Addison-Wesley Publication

Michalewicz, Z. (1994), "Evolutionary Computation Techniques for Nonlinear Programming Problems," International Transactions of Operational Research, 1(2), pp. 223-240

Gen, M. and Cheng, R. (1997), "Genetic Algorithms and Engineering Optimization," Wiley-Vch