

協同運算研究() – 多單體基因演算法

研究生：李漢祥

指導教授：張炳騰 博士

曾宗瑤 博士

東海大學工業工程與經營資訊研究所

摘要

在求解最佳化的過程中，精確度(Precision)、收斂率(Convergence Rate)以及計算時間(Computing Time)這三者的結果關係著演算法搜尋能力的好壞，一旦遇到高維度且複雜度高的問題時，不好的演算法便會使的精確度降低、收斂率及計算時間增加，如何解決這三大存在已久的問題便是本研究的重點所在。

因此，本研究認為要使得搜尋的效率更高，就應導入協同搜尋(Cooperative Search)的概念。本研究認為在搜尋的過程中應強化多群體(Multi-team)的關連性，建立群體間與群體內部資訊的溝通、傳遞及協調機制，使群體間能記憶彼此的資訊並藉由持續的交換及激化(stimulate)後，快速找到最佳點的資訊，並讓群體間在適當的時機進行分合動作，以快速包圍最佳點的位置，以達快速收斂及有效搜尋的目的。

為實現協同式搜尋(Cooperative Search)的概念，本研究藉由基因演算法(GA)及單體法(Simplex Method)的合併，來作為其演算法之一：混合式多單體 GA 協同搜尋法(HMSGGA-CS)。許多文獻強調基因演算法(GA)的優點是在對於整個解空間的搜尋能力(Exploration)極佳並將它廣泛的使用在求解最佳化的問題上，但基因演算法(GA)欠缺的是探勘(Exploitation)的能力，因此便將有探勘能力的局部搜尋法(Local Search)與之混合。本研究除了利用此優點外，並導入協同搜尋的概念，以使搜尋效率更佳。

在實驗驗證方面，本研究隨機選取數個範例來驗證成效。實現結果顯示 HMSGGA-CS 確實能在所有的範例中有效且快速的找到真正的最佳解。實驗結果也將與傳統的基因演算法及 Yen-Lee(Y-L)所提出的混合法比較。HMSGGA-CS 在所有的範例中測試結果顯著的優於其它的方法。

Study of Cooperative Computing Methodology() – Multi-Simplexes Genetic Algorithms

Student : Han-Hsiang Lee

Advisor : Dr. Ping-Teng Chang

Dr. Tsueng-Yao Tseng

Institute of Industrial Engineering & Enterprise Information
Tunghai University

Abstract

In the process of solution optimization, precision, convergence rate and computing time are crucial to the ability of a search methodology. Once coming across a problem with high dimensions and complexity, poor algorithms will debase the precision and convergence rate or boost up the computing time. This paper aims at these three long-existing problems.

Therefore, to improve efficiency, implementing a concept of cooperative search is significant. It is also vital to enhancing the contact among multi-teams and building up the inner communication policy between teams. In this way, individuals may catch information from each other, and through exchanging and stimulating, can hit upon the allied information of the optimal solution. All these allow the multi-teams to locate the optimal solution in a most efficient and convergent way.

This paper also combines genetic algorithm (GA), and simplex method as one of the algorithms for the realization of the concept of cooperative search, called hybrid multi-simplexes GA cooperative search(HMSGGA-CS). Many papers have emphasized that genetic algorithms have good exploration at searching all solution space and therefore GA has been extensively applied in the area of the optimal solution searching. Nevertheless, “exploitation” is what GA lacks. Therefore, we hybridized GA with a local search that has the ability of

exploitation. Above and beyond, Cooperative Search is also adopted to increase the searching competence.

To demonstrate the statement above, this paper adopts random-sampling test for 15 functions. The consequence verifies that HMSGGA-CS essentially discovers the supreme solution in each example. In addition, we also compared with traditional genetic algorithm and Yen-Lee's (Y-L) hybrid methods, it shows that HMSGGA-CS evidently reveals its priority in real implementations.

誌謝

論文得以順利完成，首先要感謝我的指導教授張炳騰老師，在他細心的指導下，給予我在寫論文時有相當大的幫助及想法，另外對於未來人生的方向與規劃，也給予我諸多的指引，亦師亦友的情誼與師恩，永誌難忘。

口試期間，承蒙曾宗瑤老師及黃良志老師在百忙之中特別撥空審閱論文，提供寶貴的意見，使我的論文得以更加完整，也讓我瞭解做研究所必須有的嚴謹態度，由衷的感謝各位老師。

在研究過程中，感謝研究室同窗至勇及家聰的幫忙及相互扶持，一起走過最艱難的時光，讓我從一個陌生的環境中成長與學習。此外，感謝學弟郁文、純行及嘉偉的鼓勵及支持，使我在低潮時依舊能擁有歡笑。

最後，特別要感謝父母及家人的關懷與照顧，使我能在無憂無慮的環境下，順利完成學業。

在此，僅以本論文獻給所有關心我及支持我的人，謝謝你們。

李漢祥 謹誌於
東海大學工業工程與經營資訊學系
民國九十一年六月

目錄

中文摘要.....	I
英文摘要.....	III
誌謝.....	V
目錄.....	VI
表目錄.....	VIII
圖目錄.....	XI
第一章 緒論.....	1
1.1 研究背景與動機.....	1
1.1.1 協同式搜尋(Cooperative Search)的原理.....	1
1.1.2 協同式搜尋預期之優點.....	2
1.2 研究目的.....	3
1.2.1 探討方向.....	3
1.2.2 基因演算法與單體法(Simplex Method)之合併.....	4
1.2.3 Y-L 混合型基因演算法[52].....	6
1.3 研究方法與步驟.....	6
1.4 研究工具.....	7
1.5 使用系統.....	8
1.6 論文架構.....	8
第二章 文獻探討.....	9
2.1 基因演算法之理論與運算.....	10
2.1.1 基因演算法之理論.....	10
2.1.2 基因演算法之運算機制.....	11
2.2 單體法(SIMPLEX METHOD)之理論與運算.....	18
2.2.1 單體法(Simplex Method)之搜尋理論.....	18
2.2.2 單體法(Simplex Method)搜尋步驟.....	18
第三章 混合式多單體 GA 協同搜尋法.....	21
3.1 混合式多單體 GA 協同搜尋法(HMSGGA-CS).....	21
3.1.1 混合式多單體 GA 協同搜尋法(HMSGGA-CS)概念.....	21
3.1.2 混合式多單體 GA 協同搜尋法(HMSGGA-CS)架構.....	22
3.1.3 混合式多單體 GA 協同搜尋法(HMSGGA-CS)運算流程.....	23
3.2 比較.....	31

第四章 實驗驗證.....	33
4.1 多變數函數.....	33
4.1.1 算例(一)：De Jong function F5	33
4.1.2 算例(二)：Sin function	39
4.1.3 算例(三)：De Jong function F1	44
4.1.4 算例(四)：De Jong function F2	61
4.1.5 算例(五)：Schaffer F6 function.....	65
4.1.6 算例(六)：Schaffer F7 function.....	70
4.1.7 算例(七)：Shubert function	75
4.1.8 算例(八)：SinCos function	80
4.1.9 算例(九)：Goldstein Price function.....	85
4.1.10 算例(十)：Bohachevsky function # 1	90
4.1.11算例(十一)：Easom function.....	95
4.1.12 算例(十二)：SQRN5 function.....	100
4.1.13 算例(十三)：SQRN7 function.....	114
4.1.14 算例(十四)：SQRN10 function.....	119
4.1.15 算例(十五)：Colville function.....	123
4.2 實驗結論.....	129
第五章 結論與未來研究方向.....	131
5.1 結論.....	131
5.2 未來研究方向.....	131
參考文獻.....	133
附錄一 變異數分析(ANOVA).....	138

表目錄

表 2.1 以二進位編碼為例的染色體.....	12
表 2.2 以實數編碼為例的染色體.....	13
表 3.1 混合式多單體 GA 協同搜尋法與混合型基因演算法的差異.....	31
表 4.1 測試結果表(算例一).....	36
表 4.2 文獻中之測試結果表(算例一).....	36
表 4.3 測試結果表(算例二).....	42
表 4.4 文獻中之測試結果表(算例二).....	42
表 4.5 測試結果表(算例三).....	46
表 4.6 參數實驗設計組合表(算例三).....	47
表 4.7 收斂代數之變異數分析表(算例三).....	54
表 4.8 單因子平均代數比較表(算例三).....	55
表 4.9 雙因子平均代數比較表(算例三).....	56
表 4.10 雙因子平均代數比較表(續).....	56
表 4.11 三因子平均代數比較表(算例三).....	57
表 4.12 收斂時間之變異數分析表(算例三).....	58
表 4.13 單因子平均收斂時間比較表(算例三).....	59
表 4.14 雙因子平均時間比較表(算例三).....	59
表 4.15 雙因子平均時間比較表(續).....	60
表 4.16 三因子平均時間比較表(算例三).....	60
表 4.17 測試結果表(算例四).....	62
表 4.18 測試結果表(算例五).....	67
表 4.19 測試結果表(算例六).....	72
表 4.20 測試結果表(算例七).....	77
表 4.21 測試結果表(算例八).....	82
表 4.22 測試結果表(算例九).....	87
表 4.23 測試結果表(算例十).....	92
表 4.24 測試結果表(算例十一).....	97
表 4.25 SQRN5 FUNCTION 之參數對照表.....	101
表 4.26 測試結果表(算例十二).....	102
表 4.27 參數實驗設計組合表(算例十二).....	103
表 4.28 收斂代數之變異數分析表(算例十二).....	111
表 4.29 單因子平均代數比較表(算例十二).....	112
表 4.30 雙因子平均代數比較表(算例十二).....	113
表 4.31 雙因子平均代數比較表(續).....	113

表 4.32 三因子平均代數比較表(算例十二).....	114
表 4.33 測試結果表(算例十三).....	116
表 4.34 測試結果表(算例十四).....	120
表 4.35 測試結果表(算例十五).....	125
表 A1. 1.....	139
表 A1. 2.....	139
表 A1. 3.....	139
表 A1. 4.....	139
表 A1. 5.....	140
表 A1. 6.....	140
表 A1. 7.....	140
表 A1. 8.....	141
表 A1. 9.....	141
表 A1. 10.....	141
表 A1. 11.....	141
表 A1. 12.....	142
表 A1. 13.....	142
表 A1. 14.....	142
表 A1. 15.....	143
表 A1. 16.....	143
表 A1. 17.....	143
表 A1. 18.....	143
表 A1. 19.....	144
表 A1. 20.....	144
表 A1. 21.....	144
表 A1. 22.....	144
表 A1. 23.....	145
表 A1. 24.....	145
表 A1. 25.....	145
表 A1. 26.....	145
表 A1. 27.....	146
表 A1. 28.....	146
表 A1. 29.....	146
表 A1. 30.....	146
表 A1. 31.....	147

表 A1. 32.....	147
表 A1. 33.....	147
表 A1. 34.....	148
表 A1. 35.....	148
表 A1. 36.....	148
表 A1. 37.....	148
表 A1. 38.....	149
表 A1. 39.....	149
表 A1. 40.....	149
表 A1. 41.....	149
表 A1. 42.....	150
表 A1. 43.....	150
表 A1. 44.....	150
表 A1. 45.....	150

圖目錄

圖 1.1 搜尋演算法的演化簡圖.....	1
圖 1.2 合併概念圖.....	5
圖 1.3 研究流程圖.....	7
圖 2.1 基因演算法之運算流程圖.....	11
圖 2.2 單點交配.....	14
圖 2.3 雙點交配.....	15
圖 2.4 字罩交配.....	15
圖 2.5 基因突變.....	16
圖 2.6 單體法運算流程圖.....	20
圖 3.1 混合式多單體 GA 協同搜尋法(HMSGGA-CS)架構圖	23
圖 3.2 演算流程圖.....	24
圖 3.3 多單體：複製前.....	26
圖 3.4 多單體：複製後.....	26
圖 3.5 交配概念圖：基因(GEN)方式.....	28
圖 3.6 交配概念圖：單體(SIMPLEX)方式.....	28
圖 3.7 單體突變概念圖.....	30
圖 4.1 DE JONG F5 之圖形.....	34
圖 4.2 收斂率比較圖(算例一).....	37
圖 4.3 平均收斂代數比較圖(算例一).....	37
圖 4.4 平均收斂代數之標準差比較圖(算例一).....	37
圖 4.5 達收斂時之平均運算時間比較圖(算例一).....	38
圖 4.6 平均運算時間之標準差比較圖(算例一).....	38
圖 4.7 與理想值之誤差比較圖(算例一).....	39
圖 4.8 代數收斂表示圖(算例一).....	39
圖 4.9 SIN 函數之圖形	40
圖 4.10 收斂率比較圖(算例二).....	42
圖 4.11 平均收斂代數比較圖(算例二).....	43
圖 4.12 平均收斂代數之標準差比較圖(算例二).....	43
圖 4.13 達收斂時之平均運算時間比較圖(算例二).....	43
圖 4.14 平均運算時間之標準差比較圖(算例二).....	44
圖 4.15 與理想值之誤差比較圖(算例二).....	44
圖 4.16 DE JONG F1 之圖形	45
圖 4.17 收斂率比較圖(算例三)	51
圖 4.18 平均收斂代數比較圖(算例三).....	52

圖 4. 19 平均收斂代數之標準差比較圖(算例三)	52
圖 4. 20 達收斂時之平均運算時間比較圖(算例三).....	52
圖 4. 21 平均運算時間之標準差比較圖(算例三)	53
圖 4. 22 與理想值之誤差比較圖(算例三)	53
圖 4. 23 代數收斂表示圖(算例三).....	54
圖 4. 24 DE JONG F2 之圖形	61
圖 4. 25 收斂率比較圖(算例四)	63
圖 4. 26 平均收斂代數比較圖(算例四).....	63
圖 4. 27 平均收斂代數之標準差比較圖(算例四)	63
圖 4. 28 達收斂時之平均運算時間比較圖(算例四).....	64
圖 4. 29 平均運算時間之標準差比較圖(算例四)	64
圖 4. 30 與理想值之誤差比較圖(算例四).....	65
圖 4. 31 代數收斂表示圖(算例四).....	65
圖 4. 32 SCHAFFER F6 之圖形	66
圖 4. 33 收斂率比較圖(算例五)	68
圖 4. 34 平均收斂代數比較圖(算例五)	68
圖 4. 35 平均收斂代數之標準差比較圖(算例五)	68
圖 4. 36 達收斂時之平均運算時間比較圖(算例五).....	69
圖 4. 37 平均運算時間之標準差比較圖(算例五)	69
圖 4. 38 與理想值之誤差比較圖(算例五).....	70
圖 4. 39 代數收斂圖(算例五).....	70
圖 4. 40 SCHAFFER F7 之圖形	71
圖 4. 41 收斂率比較圖(算例六)	73
圖 4. 42 平均收斂代數比較圖(算例六)	73
圖 4. 43 平均收斂代數之標準差比較圖(算例六)	73
圖 4. 44 達收斂時之平均運算時間比較圖(算例六).....	74
圖 4. 45 平均運算時間之標準差比較圖(算例六)	74
圖 4. 46 與理想值之誤差比較圖(算例六).....	75
圖 4. 47 代數收斂表示圖(算例六).....	75
圖 4. 48 SHUBERT 之圖形	76
圖 4. 49 收斂率比較圖(算例七)	78
圖 4. 50 平均收斂代數比較圖(算例七).....	78

圖 4. 51 平均收斂代數之標準差比較圖(算例七)	78
圖 4. 52 達收斂時之平均運算時間比較圖(算例七).....	79
圖 4. 53 平均運算時間之標準差比較圖(算例七)	79
圖 4. 54 與理想值之誤差比較圖(算例七).....	80
圖 4. 55 代數收斂表示圖(算例七)	80
圖 4. 56 SINCOs 之圖形	81
圖 4. 57 收斂率比較圖(算例八)	83
圖 4. 58 平均收斂代數比較圖(算例八)	83
圖 4. 59 平均收斂代數之標準差比較圖(算例八)	83
圖 4. 60 達收斂時之平均運算時間比較圖(算例八).....	84
圖 4. 61 平均運算時間之標準差比較圖(算例八)	84
圖 4. 62 與理想值之誤差比較圖(算例八).....	85
圖 4. 63 代數收斂表示圖(算例八)	85
圖 4. 64 GOLDSTEIN-PRICE 之圖形.....	86
圖 4. 65 收斂率比較圖(算例九)	88
圖 4. 66 平均收斂代數比較圖(算例九)	88
圖 4. 67 平均收斂代數之標準差比較圖(算例九)	88
圖 4. 68 達收斂時之平均運算時間比較圖(算例九).....	89
圖 4. 69 平均運算時間之標準差比較圖(算例九)	89
圖 4. 70 與理想值之誤差比較圖(算例九).....	90
圖 4. 71 代數收斂表示圖(算例九)	90
圖 4. 72 BOHACHEVSKY # 1 之圖形	91
圖 4. 73 收斂率比較圖(算例十)	93
圖 4. 74 平均收斂代數比較圖(算例十)	93
圖 4. 75 平均收斂代數之標準差比較圖(算例十)	93
圖 4. 76 達收斂時之平均運算時間比較圖(算例十).....	94
圖 4. 77 平均運算時間之標準差比較圖(算例十)	94
圖 4. 78 與理想值之誤差比較圖(算例十).....	95
圖 4. 79 代數收斂表示圖(算例十)	95
圖 4. 80 EASOM 之圖形	96
圖 4. 81 收斂率比較圖(算例十一)	98
圖 4. 82 平均收斂代數比較圖(算例十一).....	98

圖 4. 83 平均收斂代數之標準差比較圖(算例十一).....	98
圖 4. 84 達收斂時之平均運算時間比較圖(算例十一)	99
圖 4. 85 平均運算時間之標準差比較圖(算例十一).....	99
圖 4. 86 與理想值之誤差比較圖(算例十一)	100
圖 4. 87 代數收斂表示圖(算例十一).....	100
圖 4. 88 收斂率比較圖(算例十二).....	108
圖 4. 89 平均收斂代數表示圖(算例十二).....	108
圖 4. 90 平均收斂代數之標準差比較圖(算例十二).....	108
圖 4. 91 達收斂時之平均運算時間比較圖(算例十二)	109
圖 4. 92 平均運算時間之標準差比較圖(算例十二).....	109
圖 4. 93 與理想值之誤差比較圖(算例十二)	110
圖 4. 94 代數收斂表示圖(算例十二).....	110
圖 4. 95 收斂率比較圖(算例十三).....	116
圖 4. 96 平均收斂代數比較圖(算例十三).....	117
圖 4. 97 平均收斂代數之標準差比較圖(算例十三).....	117
圖 4. 98 達收斂時之平均運算時間比較圖(算例十三)	117
圖 4. 99 平均運算時間之標準差比較圖(算例十三).....	118
圖 4. 100 與理想值之誤差比較圖(算例十三)	118
圖 4. 101 代數收斂表示圖(算例十三).....	119
圖 4. 102 收斂率比較圖(算例十四)	121
圖 4. 103 平均收斂代數比較圖(算例十四).....	121
圖 4. 104 平均收斂代數之標準差比較圖(算例十四)	122
圖 4. 105 達收斂時之平均運算時間比較圖(算例十四).....	122
圖 4. 106 平均運算時間之標準差比較圖(算例十四)	122
圖 4. 107 與理想值之誤差比較圖(算例十四)	123
圖 4. 108 代數收斂表示圖(算例十四).....	123
圖 4. 109 收斂率比較圖(算例十五)	126
圖 4. 110 平均收斂代數比較圖(算例十五)	126
圖 4. 111 平均收斂代數之標準差比較圖(算例十五)	126
圖 4. 112 達收斂時之平均運算時間比較圖(算例十五).....	127
圖 4. 113 平均運算時間之標準差比較圖(算例十五)	127
圖 4. 114 與理想值之誤差比較圖(算例十五).....	128

圖 4.115 代數收斂表示圖(算例十五).....128

第一章 緒論

1.1 研究背景與動機

本研究試圖提出一套新的搜尋法概念，主張演化時應將協同的概念導入其中，以擺脫個別獨立或欠缺關連性的運算機制。藉由協同的機制使演算法能建立起資訊的傳遞與溝通的能力後，並藉此來觸發與激化(Stimulate)所搜尋的群體，使群體間或群體內能因為某些因素而適時的產生變化，最終使演算法搜尋的效率更佳。

1.1.1 協同式搜尋(Cooperative Search)的原理

協同式搜尋(Cooperative Search)主要在主張多群體(Multi-teams)之間本應能自我演化並存在著資訊的交流、溝通與激化(Stimulation)的機制，告別以往的單點式或多點式的個別獨立運作方式，藉由這樣的機制使多群體在搜尋時能快速且有效的得到最佳點所在位置的資訊並將它附近的區域快速的包圍住，以達到快速收斂的目的。

圖 1.1 為本研究提出的搜尋演算法之演化簡圖。早期的演算法強調的是單點式的方式，在搜尋時最佳解時會先將周圍的資訊先比較過後再決定其路徑。到了下一代的演算法時則是以多點來同時搜尋最佳解的資訊。接下來則是以多群體的方式來個別獨立搜尋最佳點的資訊，而群體間並無任何的關連性存在。最後是由協同式搜尋的方式將不同的群體間有關連的個體重新分合與激化(Stimulate)後，將最佳點的資訊迅速的找出來。接下來將說明以往演算法的缺點或不足之處。

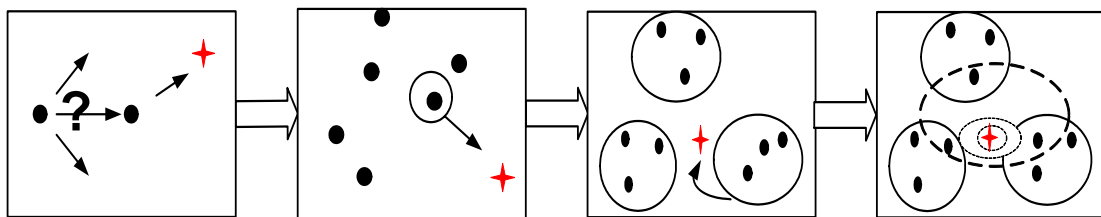


圖 1.1 搜尋演算法的演化簡圖

對於以點為主的演算法而言，找尋最佳解的效率的好壞可能決定於母體數量的多寡，如果母體數量少，它所搜尋的範圍就小，且容易提早收斂，因此不容易將真正的最佳解給挖掘出來。但如果數量一多，因為在搜尋時彼此獨立、互相缺乏訊息的緣故，即使曾經很靠近最佳值了，但總是缺臨門一腳，始終無法將最佳解給找出來，使的搜尋時間增長，效率不佳。

在搜尋的過程中，點跟點之間的互動關係著搜尋後的最適值結果以及能不能快速收斂。我們必須瞭解當某一點或多點同時在演算時，而其他的點的是否也在演算而互相影響著。因此如果點跟點之間真的有互動關係的話，或許不需要那麼多的初始點，則能快速收斂到最佳解。不過，這樣的機制其實還不夠，如果再加上群體的概念，就能使的運算的機制更趨完善。

不同的群體，雖然都各有其自主性，也就是各有其演化方式，但果只是各自獨立演化的話，就如同多點式的運作方式，只不過是將它們用框框包圍起來而已。也因此，求解時它們應該存在一些溝通的機制或資訊的傳遞以達到互相交換的目的。藉由這樣的動作，相關連的個體能一起運作，原本沒有關係的個體，也能因空間中的因子而產生關連性，便能使的演化後的解更好。另外，不同的群體之間，在演算的過程中，有可能會被干擾而分開，也有可能因為某種因素而重新組合成新的群體。若能將這些會影響演化結果的因子找出後並予以控制的話，相信這樣的演算機制必能快速且有效率的找到最佳值。

1.1.2 協同式搜尋預期之優點

協同式搜尋預期的優點可以歸納為下列幾點：

1.創造群體之間資訊的傳遞與溝通：

藉由協同式搜尋的演化機制，使群體與群體之間及群體內部能互相比較，產生並交換彼此的資訊，以建立起資訊的傳遞與溝通的機制。

2.記憶資訊與持續交換以及相互激化(Stimulation)：

協同式搜尋能幫助演算法在演算時記憶以往群體的資訊，並在適當時機使群體能重新組合與交換進而激化新群體，以避免重複搜

尋的現象。

3.快速包圍最佳點所在的區域：

藉由彼此資訊的交換與溝通，使不同的群體在必要時能重新組合與交換，進而包圍最佳點所在的區域。

1.2 研究目的

1.2.1 探討方向

本研究主要是要探討在演算的過程中如何使群體跟群體或點跟點之間存在溝通機制或關連性，並能夠在不同的群體之中進行交換，使的即使在前一代時它是不好的但因為存在關連性而能被保留下來，以達到有效率的計算與快速的收斂。本研究的探討方向可歸納為下列幾項：

1.多群體(Multi-teams)與多點的機制對其求解效率的影響：

本研究將探討以群體為主的演算機制是否比以點為主的演算機制來的好。在此，並非多群體內點的數量一定比多點的數量來的多。本研究將藉由單體法將母體分為多群體的方式來演算，而群體內點的數量將依不同的範例類型而有所不同，進而探討其求解效率。

2.群體與群體之間如何傳遞訊息：

群體間傳遞訊息的方式有如化學反應中的觸發現象或以隨機的機制來進行訊息的溝通與傳遞。本研究將以隨機的機制來探討對效能的影響。本研究將藉由單體法單體本身內部的比較機制與基因演算法中對整個解空間的比較機制來使的群體間能夠互相傳遞訊息。

3.分合現象：

群體間或群體內何時該分合、分合是否有意義以及分合機制都是本研究探討的方向之一，而其可能的機制包括有隨機的分合現象、化學反應中分子結構的分合現象等。本研究以隨機的機制來探討對效能的影響。本研究藉由基因演算法中的運算子，如複製、交配和突變等，使群體與群體之間存在著分合現象。

本研究為了演算協同式搜尋(Cooperative Search)的機制及實現本研究所提出的論點，因此本研究將利用基因演算法與單體法(Simplex

Method)合併使用來實現協同式搜尋(Cooperative Search)機制。

1.2.2 基因演算法與單體法(Simplex Method)之合併

在最佳化問題的求解過程中，要搜尋到全域解並不是一件容易的事，基因演算法是採用隨機搜尋的方式，尤其在初始解的產生是以隨機搜尋的方式，雖然可以很快的做全域的搜尋，找到極值附近的區域解，但想從最佳解附近逼近到最佳解上時，卻需要耗費很多的時間去演化最佳的解。另一方面，由於它在演化的過程中會淘汰不適合的母體，其淘汰的方式是比較適應函數值的大小，但適應函數較差的並不代表不好，這樣的資訊反而有可能在下幾代時需要被用到或在下幾代時反而是關鍵點，進而造成搜尋的時間增加，效率不佳等；另外，基因演算法是針對單一群體來運算，一旦遇到多維度的問題時，母體如果不夠多便無法有效的搜尋最佳解，且單一群體對包圍最佳解的能力有限。

單體法(Simplex Method)在搜尋時能藉由群體內的資訊來往最佳點的方向移動，也因此，如果它選擇的路徑是正確的，它便能將最佳點包圍起來，一旦包圍住便能達到快速收斂的效果，不過，如果遇到的問題區域解太多的話，它很難找到真正的最佳解，且很容易陷入區域最佳解，這也是它最大的缺點之一。

基於以上的缺點，本研究期望藉由單體法(Simplex Method)的搜尋方式的優點，它能保留在演化過程中輸入參數的關連性[15]，來避免因為基因演算法的演化機制的關係而將母體存在有關連性的點被淘汰。另一方面，藉由單體法(Simplex Method)群圍的特性，以及基因演算法本身群體內溝通的機制，來達到我們協同式運算的目的。下列為此種方法合併的優點：

1. 由於基因演算法計算所需的時間過長以及搜尋軌跡不易掌控，因此，藉由單體法(Simplex Method)有方向的搜尋，以及單體形狀的變化，可以改善基因演算法的不足之處。
2. 基因演算法雖然搜尋整體最佳解的能力極佳，但最佳解的值並不能如傳統演算法(如單體法)應用搜尋規則所找到之整體最佳解來得好，而可能只在最佳解的附近。因此，可藉由單體法(Simplex Method)

- 的特性，由單體來逼近最佳點，以迅速找出最佳解。
3. 基因演算法在搜尋時會將不好的母體剔除，保留好的母體，但被剔除的母體有可能會影響搜尋的結果，如果就這樣把那些次等的母體刪除，就有可能浪費更多的搜尋時間或找到不好的解；因此，如果藉由保留單體的方式，在下一代時將完整的單體複製或取代掉部分不好的母體，使的有用的資訊能被保留而沒用的資訊能被創造成有用的資訊。
 4. 基因演算法在搜尋時並無特定的方向，如果將其散亂的母體變為多單體(multi-simplexes)的群體方式來演算並擁有方向性，且由單體將解空間給包圍住的話，必能快速達到收斂的目標。

圖 1.2 為合併概念圖。藉由多單體(Multi-simplexes)的演化機制，將其納入基因演算法的各因子中，使其運作機制不再是點對點的方式，而是以群體對群體的方式來運作並能保留住母體間的關連性，以及藉由彼此的交換與激化(Stimulation)後以實現協同式搜尋的目的。

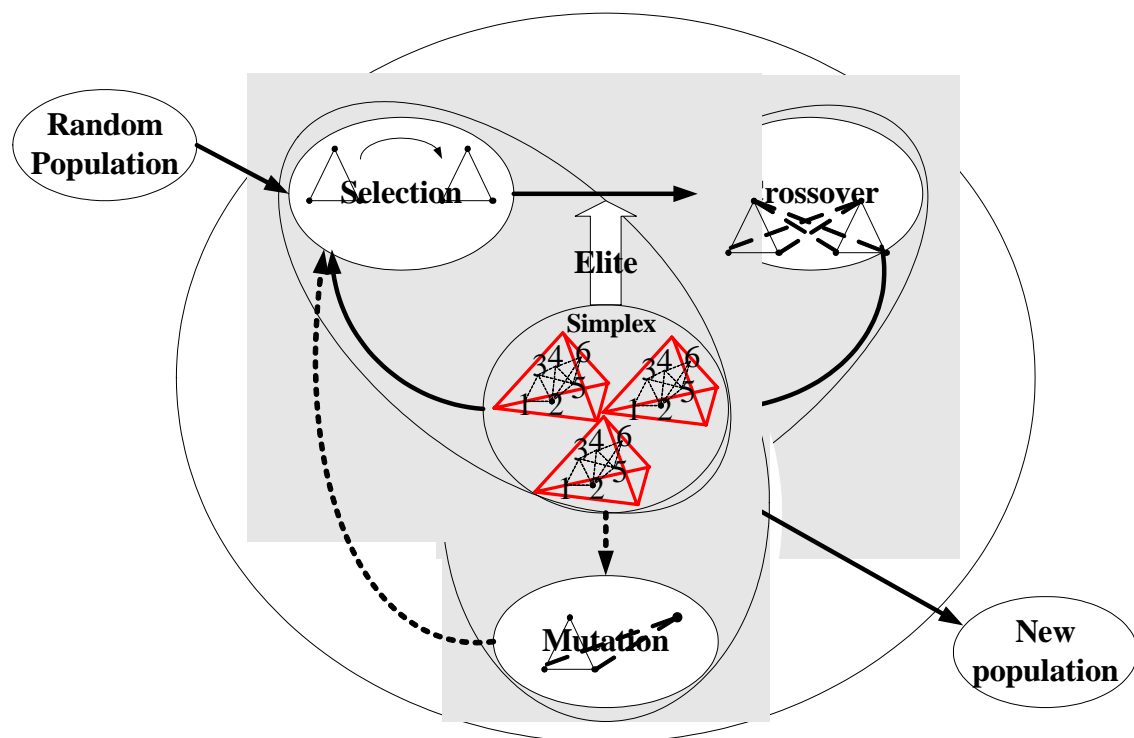


圖 1.2 合併概念圖

1.2.3 Y-L 混合型基因演算法[52]

由 Y-L 所發展的混合型基因演算法是採用機率型的單體法 (Probabilistic Simplex Method) 與實數型 (Real Value) 的基因演算法合併，並將母體做排序後較好的數點帶入單體法的演算中，因此，Y-L 所提出的方法乃是以菁英化為主的混合方法。

Y-L 所提出的單體法演算方式與傳統的單體法不同之處在於：假設有 N 個變數需最佳化，傳統的單體法是以 $N+1$ 個頂點開始運算，而 Y-L 的方法則是以 $N+$ 點為起始點，其中 α 大於 1。開始時就如同傳統的單體法一樣，最佳的 N 個點 X_1, X_2, \dots, X_N 被選擇，且計算它們的質心 \bar{x} 後，接著不是只有一個點會被反射而已而是會透過 \bar{x} 來反射數點 $X_{N+1}, X_{N+2}, \dots, X_{N+\alpha}$ 產生了 $X'_{N+1}, X'_{N+2}, \dots, X'_{N+\alpha}$ ，所有新的點如果必要的話再進行收縮 (Contraction) 的動作。經由排序、複製、反射、計算、收縮和淘汰的過程就如同序列式 (Sequential) 單體法一樣。

在 Y-L 的架構裡，其演算機制先假設現有 P 個母體，經由排序後先挑選出前 N 條染色體並複製到下一代，接著挑選出前 S 條染色體，將這 S 條染色體扣掉 N 條染色體後帶入 Y-L 所提出的機率型單體法中運算並產生出 $S-N$ 條染色體；另外，由 P 個母體中扣掉 S 個染色體後，執行傳統的基因演算法並產生出 $P-S$ 條染色體。

Y-L 所提出的演算法是以平行運算的結構來演化母代，也就是說，在每代演化時會按比例將不同數量的個體分別丟進基因演算法與單體法中，這個動作是互相獨立的，彼此並不存在著關連性，並在單代演化完成之後，在一起做比較與判斷。Y-L 的架構由此看來並未完全的將單體演化機制運用得宜。

1.3 研究方法與步驟

本研究將提出一套有效的合併方法，利用基因演算法 (GA) 與單體法 (Simplex method) 的特性，將此兩種方法完全的嵌在一起，以區別於以往演算法的合併方式。

本研究的流程主要可分為以下幾個步驟：

步驟一、確認研究方向與可行性

步驟二、基因演算法、單體法與混合型基因演算法等相關文獻的收集。

步驟三、架構混合型的基因演算法與單體法。

步驟四、進行實驗驗證與實驗設計。

步驟五、提出結論與未來可行的研究方向。

本研究的研究流程如圖 1.3 所示：

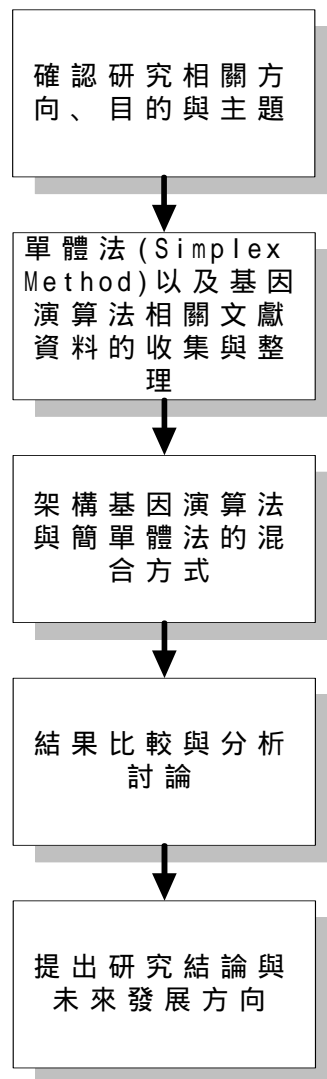


圖 1.3 研究流程圖

1.4 研究工具

本研究使用 Matlab 來作為運算工具，利用 Matlab 的特性，開發

出一套合理且可行的程式，並利用本研究中的範例求出最佳的解。Matlab 是 Matrix Laboratory 的簡寫，它是一個直譯式的語言程式，與其他的程式語言比較起來，因為語法較於單純，所以較容易學習與瞭解。Matlab 的主要特色在於數值分析、矩陣的運算與資料型態的轉換，它能夠讓使用者即使面對大量的資料以及冗長的運算問題時，也能有效的解決，因此，本研究選擇 Matlab 來作為演化最佳化問題的解決工具。

1.5 使用系統

本研究所使用的軟體平臺為 Windows 2000，並搭配 Matlab5.3 版；在硬體方面，本研究所使用的配備為 CPU：AMD 1.2G；RAM：256M。

1.6 論文架構

本研究共分為五個章節，各章節敘述摘要如下：

第一章：提出本研究的動機、背景、目的與方向及步驟等相關內容。

第二章：討論並整理與本研究相關的文獻。包含基因演算法的運算元、運作機制與運作流程等，並探討以二進位為主(Binary Coded)的基因演算法跟以實數為主(Real-valued Code)的基因演算法的差異與使用時機；另一方面，本研究也會介紹單體法(Simplex Method)的運算元及運作流程圖，並說明其優缺點以及使用時機。除此之外，對於各大電子資料庫，例如，IEEE、SDOS 或 EBSCO 等，所收集到的基因演算法與單體法合併的相關文章也將在此一併探討。

第三章：利用基因演算法與單體法的特性，提出一套有效的合併方法架構，並規劃程式的求解流程以及詳細說明合併的概念。

第四章：針對於提出來的的方法架構，進行實驗數據的驗證與分析，並與相關文獻中所提的方法比較其績效。

第五章：敘述研究的結果與心得，並探討未來可能的研究方向。

第二章 文獻探討

由於大部分的最佳化問題的一階微分都是不存在的，或者因為問題的複雜的原因，使的一階微分難以求得。針對此現象，有三種最常見的最佳化方法[37]就是單體最佳化(Simplex Optimization)、模擬退火法(Simulated Annealing)和基因演算法(Genetic Algorithms)。

另一方面，基因演算法和模擬退火法在求解的過程中都有可以跳脫區域解(Local Solution)的優點，不過，因為模擬退火法對於參數的選擇並不容易[42]，而且模擬退火法比其他兩種方法需要更多函數的計算，因此對於應用在現實環境中的例子裡，所需的成本便會較高。基因演算法的運用已將近二十年，對於基因演算法本身，在其運算元的作用上已有許多文獻予以討論，不過，由於傳統的基因演算法在求解的收斂速度上並無法滿足與些使用者的需求，於是部分使用者便將基因演算法和其他解法結合，例如基因演算法與模擬退火法(Simulated Annealing)、基因演算法與塔布搜尋法(Tabu Search)以及基因演算法與單體法(Simplex Method)的合併等。

文獻[18][4][43][7]將模擬退火法視為基因演算法之突變運算子，以替代原有的突變方式，它們之間的差異在於機率的設計方式不同。Murata [32]則是以模擬退火法所求出來的解當作基因演算法的初始解，以此來解答工作站排程問題。

Liaw[21]將基因演算法與塔步搜尋法(Tabu Search)合併來求解開放式工作站(Open Shop)的排程問題，該文獻將塔步搜尋法當作基因演算法的初始解後，再帶入基因演算法中並執行交配和突變，皆下來則勢將陷入區域解的個體以塔步搜尋法的機制來取代。

Shaffer[46]將單體法與基因演算法合併的方式是將交配池(Mating Pool)裡鄰近的點做競爭，得到的結論是好的初始點的產生可增快最佳化的收斂速度。Renders[39]將單體法的演算機制當作是基因演算法中的交配(Crossover)運算子(Operator)之一，搭配傳統式交配法則來運算，並以隨機方式來決定要使用哪種交配方法來進行交配。Yen[51]也將基因演算法和單體法合併使用，他們採用菁英化的方式將所有的母體 P 挑出 N 個好的母體先保留，再將次等的母體(S)以單

體法的方式演化出子代(S-N)，其餘的子代(P-S)再以基因演算法來產生，如此循環下去，直到達到收斂條件而停止。而 Lin[22]以及 Yen[52]則利用此方法來解答其他不同模式的問題。

綜觀上述的合併演算法，不是改變基因演算法的初始解就是將這些搜尋法則當成運算子的一部份，這樣看來似乎是為了合併而合併，並沒有考慮到群體之間的關連性以及在每代間應該記憶搜尋軌跡以提高搜尋效率。

2.1 基因演算法之理論與運算

2.1.1 基因演算法之理論

基因演算法(Genetic Algorithm)是由密西根大學教授 John Holland 在 1975 年於 *Adaption in Natural and Artifical System* 文中所提出的一般性最佳化演算法則。理論基礎可回到自 1985 年達爾文(Charles Darwin)的「物種演化」(*On the Origin of Species by Means of Nature Selection*)書中的「物競天擇，適者生存」的演化及淘汰觀念。在此原則下，透過一些人工的運算，例如，複製、交配以及突變等方式對可能的解進行演化，並根據適應函數來評估，以求得最佳解或近似最佳解。

基因演算法是近年來發展快速以及具有潛力的最佳化方法之一，它類似傳統搜尋方法之漫步法(Random Walk Method)是全域搜尋法的一種。由於它是同時以多點方式搜尋最佳解，而非點對點的搜尋，對於多峰谷之函數而言，基因演算法較傳統演算法更可以較快找出整體最佳解(Global Optimum)，同時也能避免陷入區域最佳解(Local Optimum)。此特性是基因演算法的最大優點。不過由於計算所需的時間過長、不易掌控，也是另一個缺陷。如何改善傳統基因演算法在最佳點附近收斂速度緩慢的缺點，並希望能快速求得全域最佳解，是目前 GA 最需加強的。Goldberg[15]說明了基因演算法不同於傳統的最佳化方法主要可歸納為下列幾項特性：

1. 基因演算法的運作是關於編碼後的參數，而不是參數的本身。因此可使適用的範圍更廣。

2. 基因演算法在搜尋的過程中是對整個母體做搜尋，也就是對多個點同時做搜尋，而不是單點而已。因此能避免落入區域最佳解(Local Optimum)。
3. 基因演算法使用的是目標函數(Objective Function)的資訊，不需要微分或其他輔助的資訊。因此能使用在各種型態的適應函數，例如多目標或非線性函數等。
4. 基因演算法利用機率性(Probabilistic)方式來求解，而不是使用明確性的(Deterministic)規則。這樣一來可使它更具彈性。

2.1.2 基因演算法之運算機制

一般基因演算法的運算流程可由圖 2.1 表示：

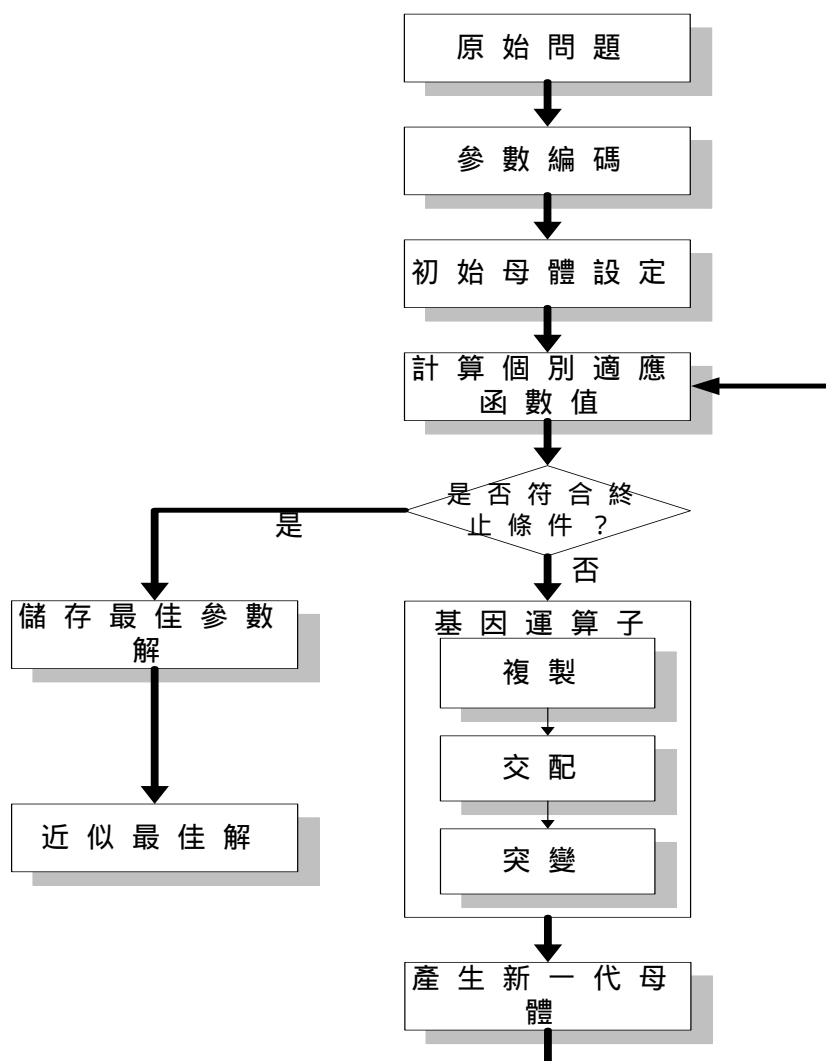


圖 2.1 基因演算法之運算流程圖

因此，若要以基因演算法來搜尋最佳解的問題時，首先需考慮以下要素：(1)決定參數編碼(Encode)方式；(2)產生初始族群，並決定族群大小(Population Size)；(3)根據問題的特性，設定適應函數(Fitness Function)；(4)執行基因運算子(Genetic Operator)，產生子代(Offsprings)；(5)設定控制參數，如交配率、突變率以及停止運算的條件等。(6)進行解碼(Decode)的動作。以下將對上述步驟加以說明。

(1)編碼及解碼

基因演算法所要運作的對象通常是未能表示可行解的字串，而非決策變數本身。所以必須先預估每個參數的搜尋範圍，再將每個參數予以編碼，在求解適應函數值後，再將字串解碼成實際的變數。一般參數的編碼方式有二進位編碼(Binary Encoding)、實數編碼(Real Encoding)。而實數編碼則是將參數集合完全不變的來做運算。

(A)二進位編碼(Binary Encoding)

由於基因演算法在初始發展時就是以此來運算，因此二進位編碼是最常用且最普遍的編碼方式。在二進位編碼中，每一條染色體都是由實數參數編碼而成，並由 0 與 1 的字串所組成。以下表 2.1 為二進位編碼的例子：

表 2.1 以二進位編碼為例的染色體

染色體 A	10110011100101011100101
染色體 B	11111110000011000001111

(B)實數編碼(Real Encoding)

對於一些比較複雜或變數很多的問題，採用實數編碼的方式來求解可以比二進位編碼更有效率，在(c)裡我們將有更詳細的探討。以下表 2.2 為實數編碼的例子：

表 2.2 以實數編碼為例的染色體

染色體 A	1.234 5.3243 0.4556 2.3293 2.4545
染色體 B	ABDJEIFDJYDSABXDSTRBVDMD
染色體 C	(back), (back), (right), (forward), (left)

(C)兩者的差異

兩者的差異在 Michalewicz[29]的書中提到：傳統的基因演算法所使用的表示法是以二進位(Binary)為主，但當遇到多維度(Multidimensional)或者需要高精確度(High-precision)的問題時就會有一些缺點存在。例如，對於一個擁有 100 個變數，範圍介於[-500,500]之間且需要精確到小數點第六位以上的問題時，以二進位的表示法來說就需要長度為 3000 的字串，對於這樣的問題，以二進位為主的基因演算法可能就無法有效的進行運算，且需浪費許多的計算時間。

在連續的區間裡要找出全域最佳解(Global Optimum)對基因演算法是一大挑戰。傳統上使用二進位表示法的基因演算法最後是將真實的設計空間給分離。雖然這樣以二進位為編碼方式的基因演算法已經很成功的運用在最佳化的問題上，但他仍然有些缺點：也就是當運用在真實世界裡的問題時，這些問題通常都會有很多的設計變數。另一個缺點是當運用在連續空間裡的參數最佳化問題時，二進位的表示法就會與真實問題的空間有所差異。例如，在真實空間裡很接近的兩個點可能在以二進位為表示法的空間裡就顯的離得很遠。

對於這樣的問題，有一種簡單的表示法就是浮點(Floating-point)表示法。在這樣以實數為編碼(Real-coded)的基因演算法裡，個體相對於設計變數是以實數來表示；因為浮點表示法接近真實的設計空間，所以這樣的基因演算法是結實的(Robust)、準確的(Accurate)和有效的(Efficient)，而且，字串的長度相對於二進位表示法減少了許多。不過，並非實數編碼的方式一定比二進位來的好，因為對於不同的問題必須常常發展一些新的交配(Crossover)或突變(Mutation)方式。

(2)初始母體的產生

在產生初始母體前必須先決定母體的大小，如果數目太大的話，

會耗費計算時間，而太小則有可能太早收斂。母體產生的方式是以隨機的方式產生或是以啟發式解產生。

(3)適應函數

應用基因演算法在求解最佳化問題之前，需將所遇到的問題轉換為適應函數，而適應函數代表著系統對外在環境的適應能力。適應函數值越高，表示該染色體具有較優的特質，將來被複製(Reproduction)或選取(Selection)的機率就越大；反之，則越容易被淘汰。適應函數的設定會影響最佳解的好壞，若設定的理想，結果就好；反之，若設定的不好，則結果就會有偏差。

(4)基因運算子(Operator)

(A)複製(Reproduction)或選取(Selection)

根據每個個體的適應程度，來決定個體被複製或選取的機率。因此，擁有較高適應值的染色體就有較高的機率被選出來進行複製。也因如此，適應函數值較低的個體，也會漸漸的消失被取代掉。目前常用的複製技術有下列兩種：(i)輪盤法(Roulette Wheel)：在輪盤上依據適應函數值的大小來劃分區域面積，也就是與適應函數值成正比。(ii)等機率法：每個個體被複製的機率皆相同。

(B)交配(Crossover)

(a)二進位表示法(Binary Representation)

此過程是隨機選取複製後的兩個母體，藉由彼此交換基因來產生新的兩個個體。交配過程發生的機率由交配機率所控制。基本上，常見的交配方式有三種形式[1]：

(i)單點交配：在所選取的兩個個體內，隨機選取一交配點後，將這兩個交配點後的基因全部交換。如圖 2.2 所示：

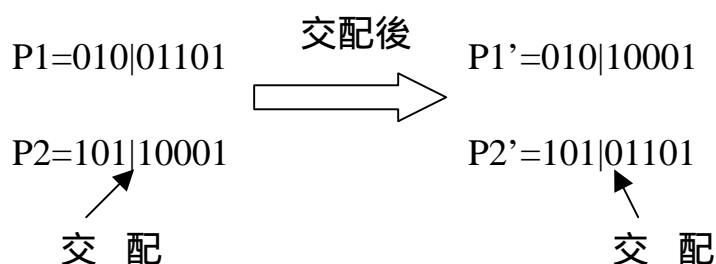


圖 2.2 單點交配

(ii) 雙點交配：在所選取的兩個個體內，隨機選取二個交配點後，將這兩個交配點內的基因全部交換。如圖 2.2 所示：

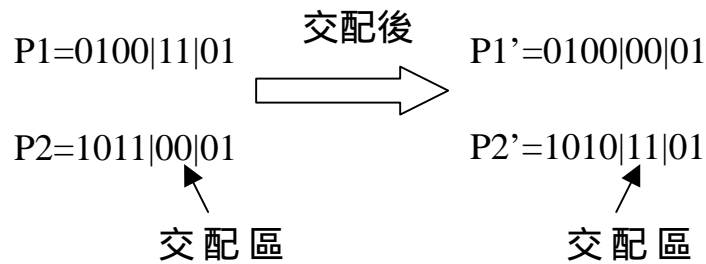


圖 2.3 雙點交配

(iii) 字罩交配：先產生一與染色體長度相同的字串，即稱為字罩。字罩由 0 與 1 隨機產生的字串組合，當字罩位元為 1 時，則兩個染色體於字罩為 1 的位置互相交換，以形成新的染色體。如圖 2.4 所示。

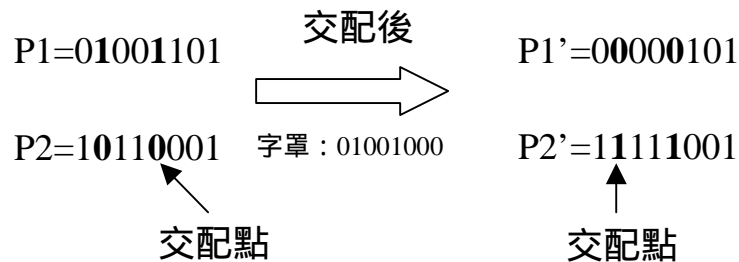


圖 2.4 字罩交配

(b) 實數表示法[29]

除了上述的交配方式外，也有很多的文獻在討論交配的方式，包括多點交配等等。由於二進位的編碼方式無法滿足現實生活中所需解決的問題，且無法保證子代一定會比母代好，因此以實數為編碼方式的基因演算法因而產生。以方向為主(Direction-based)[28]的運算子就是為了改善交配過程的演化，使子代能比母代更好。

除了上述的交配方式外，也有很多的文獻在討論交配的方式，包括多點交配等等。由於二進位的編碼方式無法滿足現實生活中所需解決的問題，且無法保證子代一定會比母代好，因此以實數為編碼方式

的基因演算法因而產生。以方向為主(Direction-based)[28]的運算子就是為了改善交配過程的演化，使子代能比母代更好。

以方向為主的交配運算子使用目標函數的值來決定搜尋的方向。假設有兩個母代 B_1 和 B_2 產生子代 B' ，則

$$B' = r \cdot (B_2 - B_1) + B_2 \quad (\text{式 2-1})$$

其中 r 是介於 1 和 0 的隨機變數，且假設 B_2 不比 B_1 差，也就是說對於最大化問題， $fitness(B_2) \geq fitness(B_1)$ 。

Murata[31]另提出 10 種不同的交配方式對於流程式排程問題進行電腦模擬測試。

(C)突變(Mutation)

(a)二進位表示法

在演算的過程中，程式會隨機產生一個突變的機率值，若此值比原先定義突變率低，則染色體會進行突變的程序。所謂的突變，是隨機選取染色體上的某一基因，將此基因做 0 或 1 的交換。舉例來說，假設預設的突變率為 0.01，若產生的亂數小於 0.01 則進行突變，否則繼續檢查下一字元。如圖 2.5 所示，僅第 3 個位元產生的亂數小於 0.01，因此將它變更。此目的是避免過早收斂，以跳脫區域解。

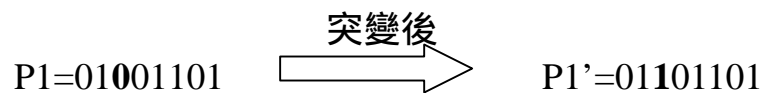


圖 2.5 基因突變

(b)實數表示法[29]

假設 a_i 和 b_i 是每個變數的上下界， \bar{X} 和 \bar{Y} 是其中的兩條染色體，且有 n 個變數，則

$$\bar{X} = (x_1, x_2, \dots, x_i, \dots, x_n), \quad \bar{Y} = (y_1, y_2, \dots, y_i, \dots, y_n)$$

(i)均勻突變(Uniform Mutation)：隨機選擇一變數 j ，使它屬於均一分配(Uniform Distribution)且介於 a_i 和 b_i 之間，也就是 $U(a_i, b_i)$ 。

$$x'_i = \begin{cases} U(a_i, b_i), & \text{if } i = j \\ x_i, & \text{otherwise} \end{cases} \quad (\text{式 2-2})$$

(ii)界線突變(Boundary Mutation)：隨機選擇一變數 j ，使它等於上界或下界。

$$x'_i = \begin{cases} a_i, & \text{if } i = j, r < 0.5 \\ b_i, & \text{if } i = j, r \geq 0.5 \\ x_i, & \text{otherwise} \end{cases} \quad (\text{式 2-3})$$

其中 $r=U(0,1)$ 。

(iii)非均勻突變(Non-uniform Mutation)：隨機設定一變數 j ，使它等於非均勻的隨機變數。

$$x'_i = \begin{cases} x_i + (b_i - x_i)f(G), & \text{if } r_1 < 0.5 \\ x_i - (x_i + a_i)f(G), & \text{if } r_1 \geq 0.5 \\ x_i, & \text{otherwise} \end{cases} \quad (\text{式 2-4})$$

其中

$$f(G) = \left(r_2 \left(1 - \frac{G}{G_{\max}}\right)\right)^b$$

$r_1, r_2 = a$ uniform random number between (0,1)

$G =$ the current generation

$G_{\max} =$ the maximum number of generations

$b =$ a shape parameter

(iv)多點非均勻突變(Multi-non-uniform Mutation)：將母代 \bar{x} 中所有的變數做非均勻突變。

Murata[31]另提出 4 種不同的突變方式來進行實驗。

(5)終止條件

因演算法在正常的運作下，如何使其停止演化，常用的方式有下列幾種方法：

(A)設定演化代數：我們可以在程式開始執行之前就先設定演化代數，當達到設定值後，則停止演化。至於應該設定幾代，則需視問題複雜度以及資料量而定。

(B)設定演化時間：如(1)所言，先給予演化的時間，時間到了則停止。

不過(1)和(2)並無法得知是否真的以達到最佳解。

(C)當最佳解經幾個世代的演化後，並無明顯的差異時，則可視為找到最佳解。

2.2 單體法(Simplex method)之理論與運算

2.2.1 單體法(Simplex Method)之搜尋理論

(1)定義

在一個為空間裡，由 $N+1$ 個點所形成的幾何圖形，稱為單體(Simplex)。例如，在二維空間裡，單體的形狀為三角形；在三維空間裡，其形狀為四面體，以此類推。

(2)原理

單體法(Simplex Method)是由 Spendley, Hext,和 Himsworth 所提出，接著由 Nelder 和 Mead[33]所發展。單體法能使用一群的點在反應面(Response Surface)上搜尋，即使是不連續的函數依然不會妨害搜尋績效。另外，單體法的好用之處在於對於最佳化的問題，它能找到好的初始點(Starting Point)[37]。

單體法(Simplex Method)的基本原理是比較在單體上的個個頂點的目標函數值，以及在演算過程中單體逐漸向最佳點移動的一種演算法。單體法在一開始時必須先定義單體的形式，接著比較個個頂點值，找出最大的目標函數值後，將此點與其餘的點所形成的質量中心連成一線，並反射到另一端，以產生新的點，形成新的單體，在重複上述的步驟後，若達到收斂條件則停止。單體在演化的過程中會自動收縮或擴大，當單體的體積夠小或各函數值非常接近時則達到收斂條件，迭代過程便終止。

2.2.2 單體法(Simplex Method)搜尋步驟

單體法搜尋的步驟主要由以下三個動作來完成，反射(Reflection)、擴張(Expansion)以及收縮(Contraction)。我們將此單體法之步驟敘述如下，並將整個分析過程繪製流程圖，如圖 2.6 所示。

(1)定義運算子參數 α, β, γ 以及收斂條件。

(2)假設有 $N+1$ 個點(初始值) x_1, x_2, \dots, x_{N+1} ，並計算其對應函數值。

$$f_1 = f(x_1), f_2 = f(x_2), \dots, f_{N+1} = f(x_{N+1})$$

(3)找出最大函數值 $f_h = \max_{i=1 \text{ to } N+1} f(x_i)$ 及最小函數值 $f_l = \min_{i=1 \text{ to } N+1} f(x_i)$ 。

(4)計算質量中心(除了函數值最大的點以外)，及其函數值 f_0 。

$$x_0 = \frac{1}{N} \sum_{\substack{i=1 \\ i \neq h}}^{N+1} x_i \quad (\text{式 2-5})$$

(5)在 x_h, x_0 兩點的射線上，藉由“反射”之動作可得到一個新點 x_r ，並計算其函數值 f_r 。其計算式為：

$$x_r = (1 + \alpha)x_0 - \alpha x_h \quad (\text{式 2-6})$$

其中 $\alpha = \frac{|x_r - x_0|}{|x_h - x_0|}$ ， α 是反射係數， $\alpha > 0$

(6)若 $f_r < f_l$ ，則藉由“擴張”動作，產生一個新點 x_e ，並計算其函數值 f_e 。其計算式為：

$$x_e = \gamma x_r + (1 - \gamma)x_0 \quad (\text{式 2-7})$$

其中 $\gamma = \frac{|x_e - x_0|}{|x_r - x_0|}$ ， γ 是擴張係數， $\gamma > 1$

接著，比較 f_e 與 f_l ；若 $f_e < f_l$ ，則以 x_e 將 x_h 取代，並判斷是否達到收斂條件，若否，則回到步驟(2)繼續演算。若 $f_e > f_l$ ，則以 x_r 將 x_h 取代，並判斷是否達到收斂條件，若否，則回到步驟(2)繼續演算。

(7)若 $f_r > f_h$ ，表示在 x_h, x_0 兩點的射線上移動得太遠，則藉由“收縮”動作，產生一個新點 x_c ，並計算其函數值 f_c 。其計算式為：

$$x_c = \beta x_h + (1 - \beta)x_0 \quad (\text{式 2-8})$$

其中 $\beta = \frac{|x_c - x_0|}{|x_h - x_0|}$ ， β 是收縮係數， $0 \leq \beta \leq 1$

接著，比較 f_c 與 f_h ；若 $f_c < f_h$ ，則以 x_c 將 x_h 取代，並判斷是否達到收斂條件，若否，則回到步驟(2)繼續演算。若 $f_c > f_h$ ，則將各點縮小與 x_l 的距離，以縮小單體的體積，並將所有的點以 $x_i = \frac{x_i + x_l}{2}$ 取代，並判斷是否達到收斂條件，若否，則回到步驟(2)繼續演算。

(8)此法所提的收斂條件方程式如下：

$$Q = \left\{ \sum_{i=1}^{N+1} \frac{[f(x_i) - f(x_0)]^2}{N+1} \right\}^{\frac{1}{2}} \quad (\text{式 2-9})$$

藉由上式所定義的 Q ，在進行最佳化之前先給予一容許值後，再進行最佳化的工作。

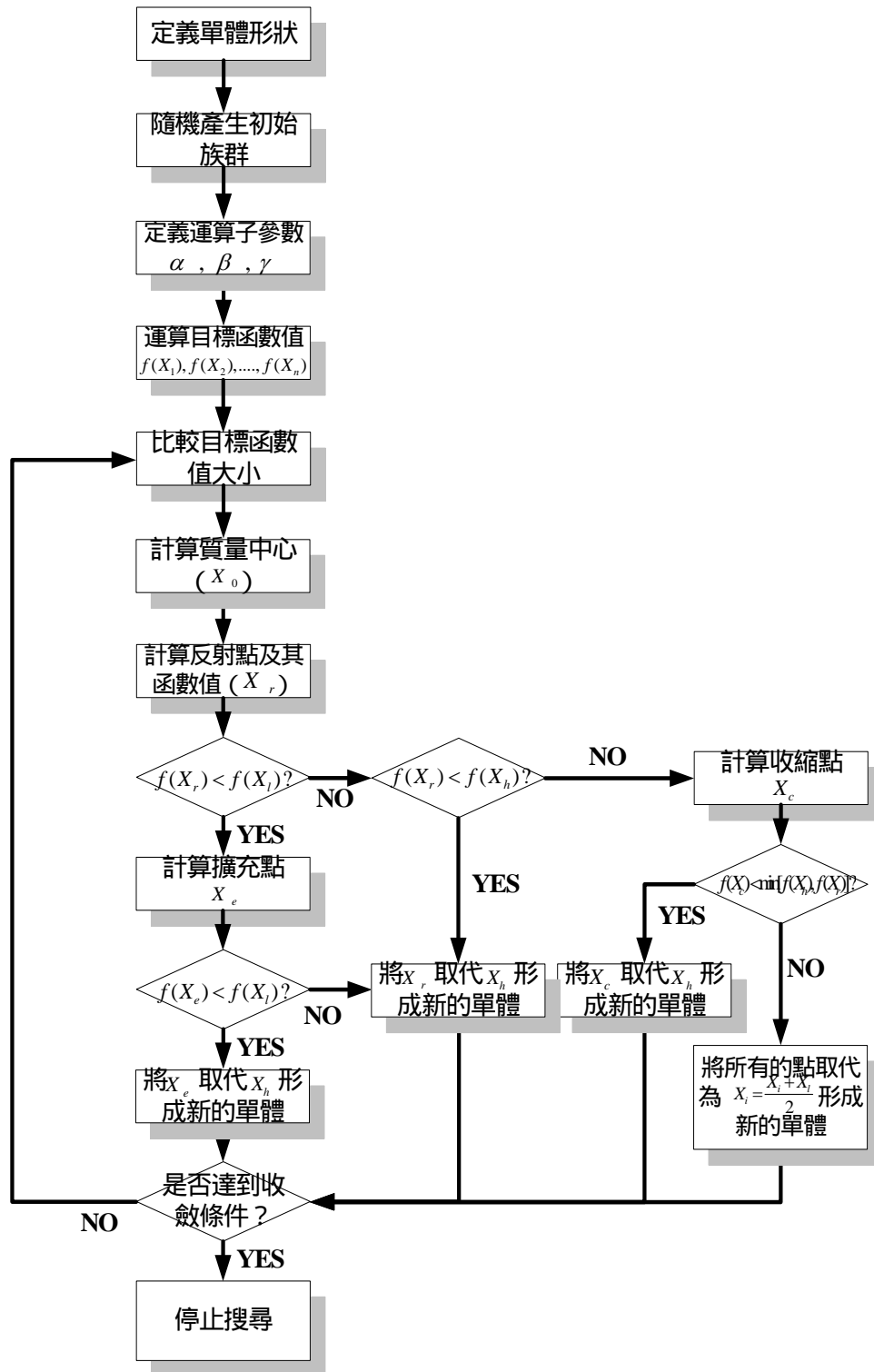


圖 2.6 單體法運算流程圖

第三章 混合式多單體 GA 協同搜尋法

本章節在說明本研究所提出的混合式多單體 GA 協同搜尋法 (Hybrid Multi-Simplexes GA Cooperative Search, HMSGGA-CS)的架構與流程，並詳細解說如何運用基因演算法與單體法(Simplex Method)來實現協同搜尋的機制，使搜尋效率更佳。

3.1 混合式多單體 GA 協同搜尋法(HMSGGA-CS)

3.1.1 混合式多單體 GA 協同搜尋法(HMSGGA-CS)概念

本節將說明我們所提出的混合式多單體 GA 協同搜尋法 (HMSGGA-CS)的概念。很多文獻在探討演算法的合併時，並非真正的將它們合併起來，有的可能是透過基因演算法的演算後找出較好的解之後，再由另一種演算法繼續演化，而有的文獻則是先讓其它的演算法先運算，待有好的初始解後，再帶入基因演算法中[25]。另外，也有的文獻是將母體共用，同時並行演算[44]；有的文獻則是採用將演算法分開來搜尋不同維度的方法[41]。總而言之，這類的合併演算法不是將它們分開，就是想辦法改變一些運算子(Operator)[16]-[31]。因此，本研究覺得所提出的合併演算法不能應該只有如此的想法，應必須將協同(Cooperative)的概念加入其中，並以適當的演算法來實現它，且強調各種演算法的完整性，因此基因演算法與單體法(Simplex Method)的合併則為本研究選擇的工具之一。

本研究主張單體(Simplex)演算時應保有其單體的結構與完整，不因為代數的更換之後而被打散掉。在基因演算法的運算子中本研究皆將單體運作在其中，藉由單體與基因演算子的組合進行單體間或單體內部資訊的溝通與交換，使單體被激化(Stimulate)並適時的做變化，並在每代演算之後，由單體與單體之間來做比較與演化，而不是點對點的方式比較，因為本研究認為如果考慮到單體結構之完整性，這些單體便能延續原有的搜尋軌跡與路線，而不需要在對整個解空間再做一次無效率的搜尋動作。單體在經過演化數代的溝通與持續交換後，必能快速包圍最佳點並挖掘(Exploitation)出最佳解。除此之外，本研究也加入了菁英化的概念，適應能力高的就完整的保留到下一代的交

配過程中。

3.1.2 混合式多單體 GA 協同搜尋法(HMSGGA-CS)架構

本研究所發展的混合式多單體 GA 協同搜尋法(HMSGGA-CS)是將協同機制建立在 Nelder-Mead 所提出的單體法(Simplex Method)與傳統的基因演算法上，並結合此兩種方法的優點，使基因演算法在運算時能夠利用到單體法(Simplex Method)搜尋區域解的特性，並將單體法(Simplex Method)分為多單體(Multi-simplexes)結構同時運用在基因演算法的運算子(Operator)中。本研究的方法在運算時主張的是除了基因演算法原有的機制外，還必須在每代的演化過程中必須保有單體的形狀，也就是說即使在複製(Selection)階段時都能將整個單體複製到下一代，以保留單體演化及記憶資訊的特性。在交配階段(Crossover)時，藉由單體對單體(不再是點對點)內外資訊的溝通與交換，將最佳解的區域包圍起來，以達到更快速的收斂。在突變(Mutation)階段時，為了避免單體因為連續的演化而落入區域解(Local Optimal)，因此本研究以突變機制來將單體上的某些點跳脫此區域解，以尋求更好的解。另外，本研究也採用菁英化(Elite)的方式將每代演化後所產生的單體保留起來，以強化搜尋效率。

圖 3. 1 為本研究所發展的混合式多單體 GA 協同搜尋法(HMSGGA-CS)架構圖，其中 N 為母體總數， P 為單體菁英化(Elite)數。藉由多單體複製(Multi-simplexes Selection)及點複製(Point Selection)的機制將較優良的母代保留到下一代，再將複製後的單體(Simplex)或個體(Individual)進行多單體交配(Multi-simplexes Crossover)或點交配(Point Crossover)，接下來則由交配完的群體分為多單體(Multi simplexes)，並將這些單體同時進行單體法(Simplex Method)演化機制，演化完成的單體則依適應值大小排序，優良的單體則被保留到下一代交配動作中。圖中的虛線表示在演化的過程中並不一定會使用到，如果陷入區域解(Local Optimum)的話，則進行突變，如此反覆進行，直到找出最適解。

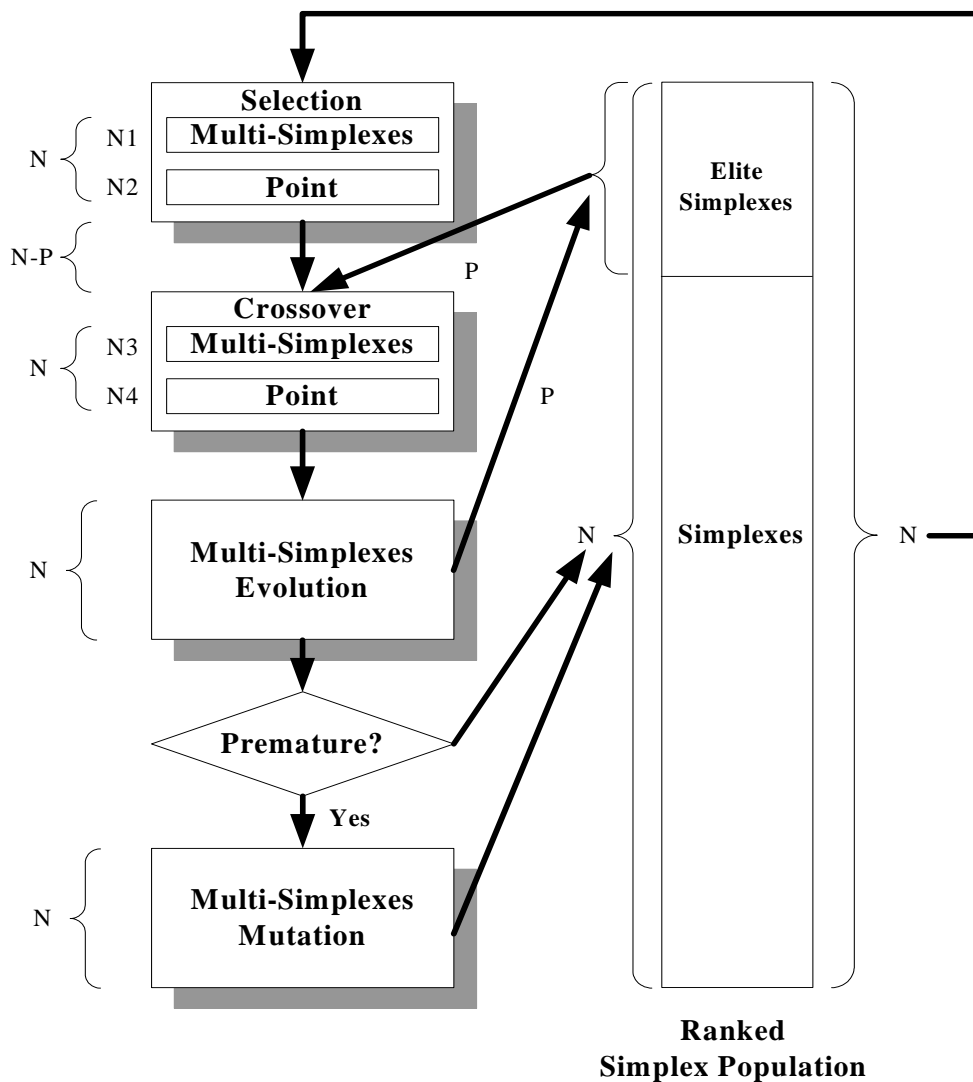


圖 3.1 混合式多單體 GA 協同搜尋法(HMSGGA-CS)架構圖

3.1.3 混合式多單體 GA 協同搜尋法(HMSGGA-CS)運算流程

在本節將介紹混合式多單體 GA 協同搜尋法(HMSGGA-CS)的運算流程，並說明如何使此兩種演算法來實現協同式搜尋機制。圖 3.2 為演算流程圖。

- 步驟一、隨機產生初始母體 P 。
- 步驟二、採用輪盤法來複製母代到下一代。
- 步驟三、由機率來決定交配母體。
- 步驟四、將多點 P 分為多單體 $P/(N+1)$ 並同時執行單體法運算機制 K 代，其中 N 為變數個數， K 為設定參數。
- 步驟五、將步驟四執行完的多個單體，保留數個單體到下一代的

步驟二中。

步驟六、判斷是否達收斂條件。如果是，則程式停止；如果沒有，則執行步驟七。

步驟七、判斷是否若入區域解(Local Optimal)。如果是，則執行步驟八；如果沒有，則執行步驟二。

步驟八、執行突變動作。

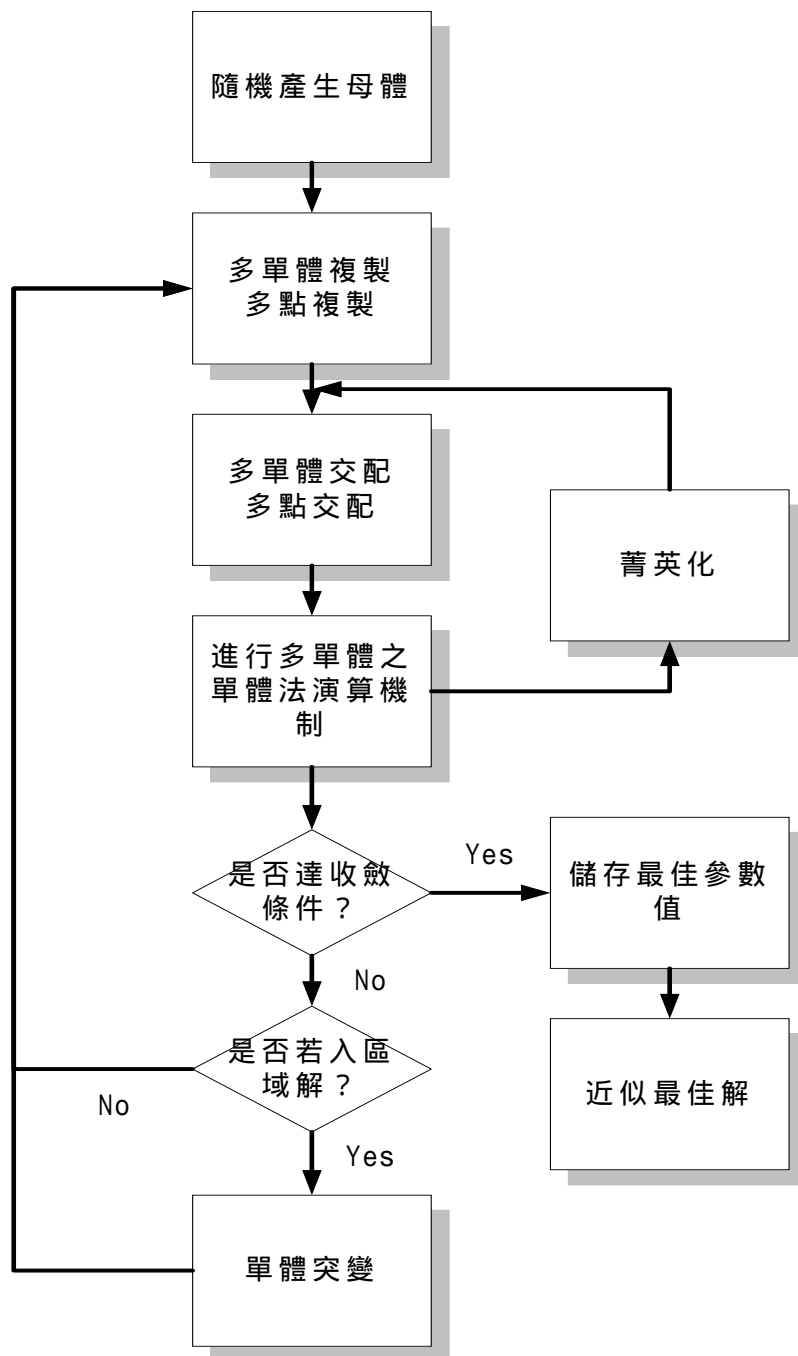


圖 3.2 演算流程圖

接下來我們將為流程中的個個步驟作更詳細的說明，並說明如何協同搜尋法則如何運作：

(1)多單體(Multi-simplexes)複製，點(Point)複製：

在第一代的演化過程中與傳統的基因演算法並無太大差異，也就是說，適應函數值大的較容易被複製到下一代去。但到了第二代以後，本研究所主張的不再只是點跟點的複製而已，而是藉由數個單體本身演化數代後，整個單體的複製，一旦某些單體演化不佳，使的它所對應的適應函數值也跟著不佳，這樣一來就會使的這些單體因為複製的機制而被淘汰掉，並讓其它適應值較優良的單體整個的被複製並繼續上一代的演化，因此在這裡本研究所強調的是維持單體的完整性。本研究不希望因為代數的替換而將原本所擁有的資訊給忽略掉，使的原本應該可以按相同搜尋軌跡而快速找到最佳值的機會，因為結構被切割或打散，而造成搜尋的效率不佳。

圖 3. 3 及圖 3. 4 為本研究中多單體(Multi-simplexes)複製的概念圖。假設該圖為最小化問題，將圖中單體視為是經過一代或數代演化後的形狀，圖中由螺旋所構成的區域即代表該處有區域解(Local Optimal)。在圖 3. 3 中，多單體經過數代單體法(Simplex Method)演化後形狀為 A、B 及 C，假設單體 B 及單體 C 因為擁有較佳的適應函數值的關係，因此在下一代時將被複製保留，而單體 A 則被取代。因此在圖 3. 4 中，單體 B 及 C 則因被複製到的關係，可以繼續維持單體的完整性，而繼續演化到 B' 及 C'，進而可以更接近最佳值甚至包圍最佳值。而其餘沒被以單體型態複製到的個體則有可能以傳統的點(Point)複製的機制被複製到下一代。

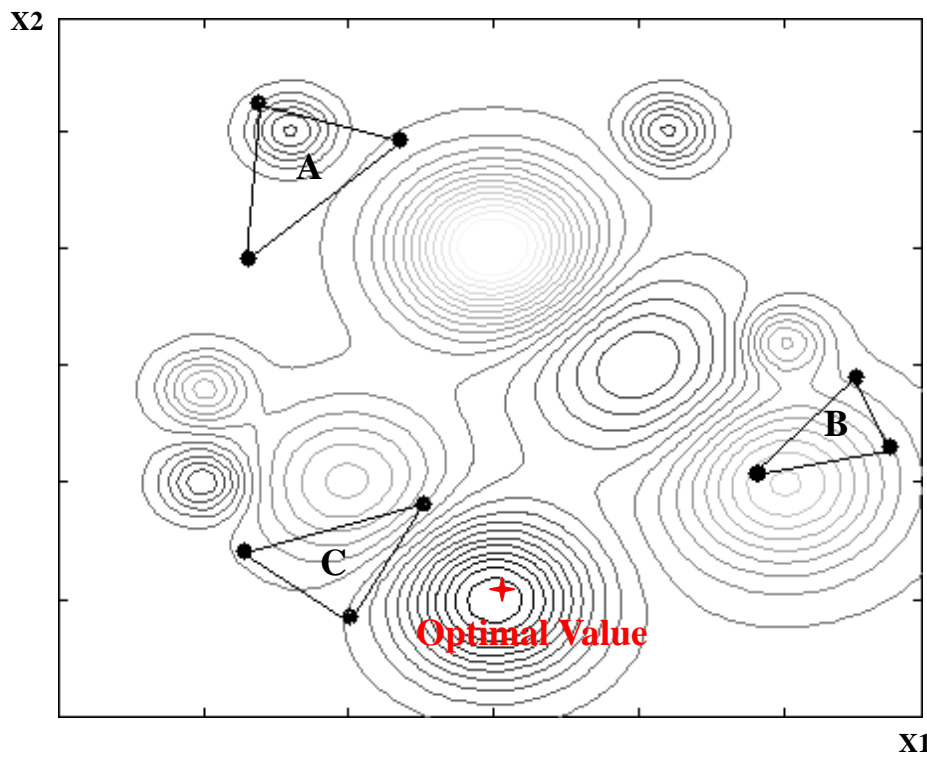


圖 3.3 多單體：複製前

單體 A、B 及 C 為自我演化(self-evolution)後的結構

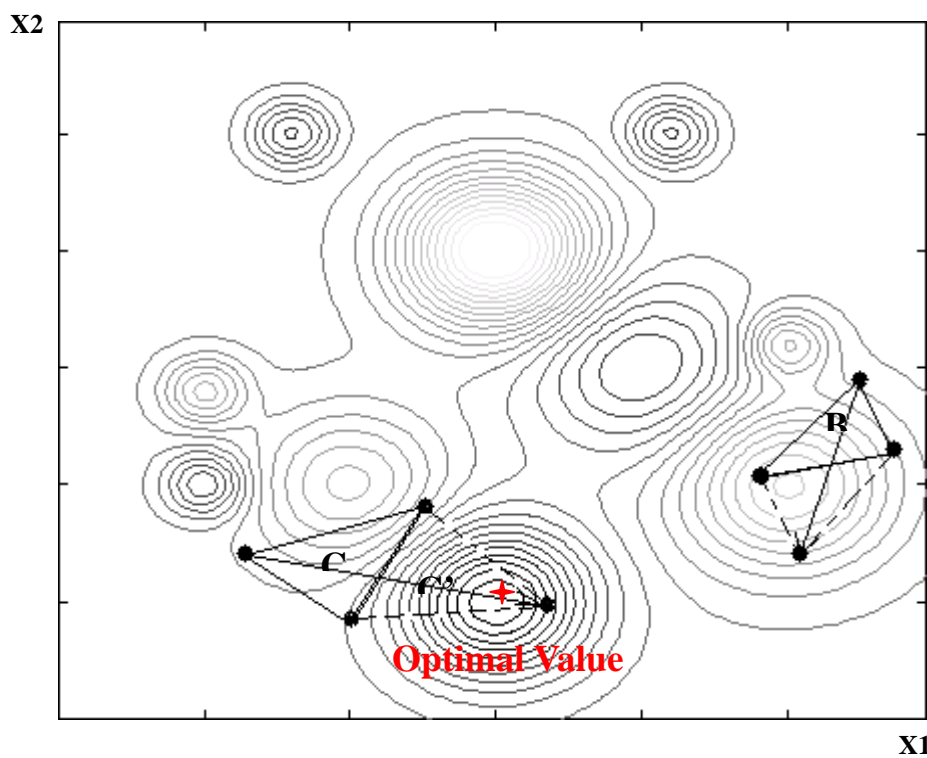


圖 3.4 多單體：複製後

單體 B 及 C 為自我演化(self-evolution)後的結構，因複製機制得以繼續演化為單體 B' 及單體 C'

(2)多單體(Multi-simplexes)交配，點(Point)交配

在第一代時，演化機制如同傳統的基因演算法一樣，並無太大差異。但到了第二代後，對二進位式(Binary-based)的基因演算法，我們所強調的不再是基因對基因的交換而已，而是不同單體上不同點的基因對基因的交換，這樣看起來似乎與傳統的基因演算法並無太大差異，不過並非如此。由於傳統的基因演算法的交配機制是為了使搜尋範圍更廣，但這樣的過程只會造成搜尋的效率不佳，時間增長而已，如果再加上單體運算的機制，就有可能讓最佳解的區域被單體包圍著，然後再利用單體有方向的搜尋，最佳解的位置就很容易被搜尋到，如此便可大大的提升搜尋的效率。

圖 3.5 為基因式的交配概念圖。假設圖 3.5 為一最小化問題，最後形成單體 A、B 及 E。單體 A 中的 a1 點因為與 B 裡的 b1 點進行基因交配，而使得單體的形狀略微改變，單體 A 的 a1 點變為單體 A' 的 a1'，單體 B 的 b1 點變為單體 B' 的 b1'，但並沒有將所有的單體的形狀給打散掉，而單體 E 因為已經接近最佳值，所以它並沒有進行交配的動作，而是繼續下一代的演化。

圖 3.6 為多單體(Multi-simplexes)交配概念圖。同樣假設此問題為最小化問題，單體在經過多代的演化後，最後剩下由 a1a2a3 以及 b1b2b3 所組成的單體 A 及 B。在圖 3.6 中，假設因為單體間資訊的溝通與傳遞而使得單體重新分合後，造成單體 A 之 a1 點與單體 B 之 b2 點彼此交換，形成由 a1b1b2 和 a2a3b3 的兩個新的單體 A' 及 B'。由 a2a3b3 所構成的新單體靠近了最佳值，甚至將最佳值所在的位置包圍起來，這樣一來藉由單體法的運算機制便可以達到快速收斂。由單體 a1b1b2 所形成的新單體 B' 雖然看似離最佳解較遠，但反而跳脫原本的區域解。而其餘未被以單體形式進行交配的個體，則有可能進行傳統的點(Point)交配法則。

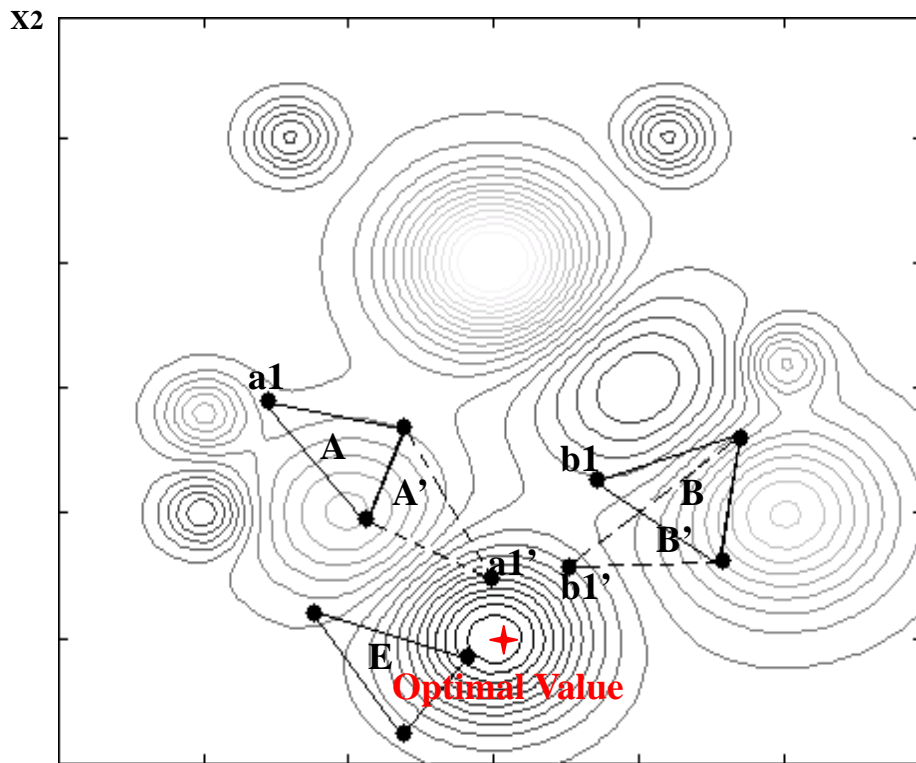


圖 3.5 交配概念圖：基因(Gen)方式

X1

a1 與 b1 點因基因交配方式使單體 A 變為 A'，單體 B 變為 B'。

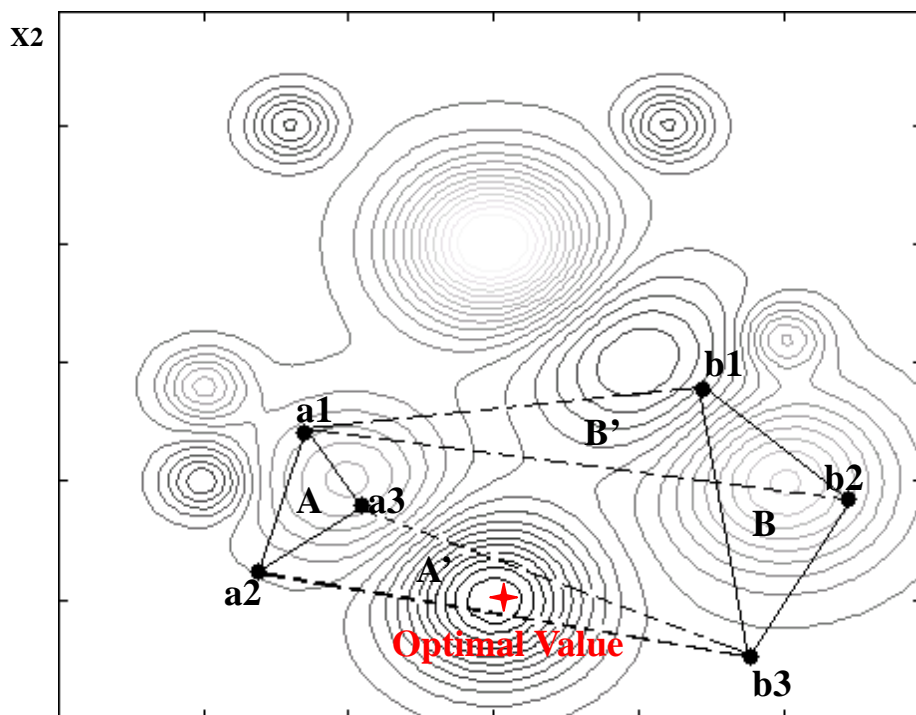


圖 3.6 交配概念圖：單體(Simplex)方式

X1

單體 A 及 B 因單體交配方式使其變為單體 A' 及 B'，並包圍最佳解。

(3)進行多單體之單體法(Simplex Method)運算機制

此步驟主要是將交配後產生的點或單體進行單體法(Simplex Method)的運算,在此所用的單體法運算子(Operator)是由 Nelder-Mead 所提出,包含反射(Reflection)、擴張(Expansion)或收縮(Contraction)。在第一代時,單體的形成是採用隨機的方式,由步驟三所產生的點來隨機構成單體。在此有一點必須注意,因為單體的形狀是跟變數有關,以 Nelder-Mead 所提出的方法來說,假設有 N 個變數,則單體就由 $N+1$ 邊行所構成,因此本研究的母體族群大小必須是 $N+1$ 的倍數才行。如果演化已經進入第二代,則單體的形成則是由步驟三所演化的單體而來。本研究將這些單體藉由單體法的演化數代後,帶入適應函數做比較,如果所求出的適應函數值越好,則本研究將部分的單體完全的保留,並將它們留到下一代,以進行步驟三。

(4)菁英化政策

本研究將多單體(Multi-simplexes)演算單體法的演算法則後,保留部分適應值較佳的單體(Simplex)到下一代。藉由此機制,可使單體不會因為每代的轉換而使得單體形狀被破壞,因為有可能因為單體本身演算代數不夠多使得探勘(Exploitation)的能力不足,一旦破壞單體的結構後,先前所搜尋過的軌跡便有可能不被紀錄,而需要再重新來過,這樣只會使得搜尋效率變差而已。

(5)判定是否達收斂條件

在本研究中將收斂條件定為(i)如果可以收斂,當達到允許的誤差時,並且需連續在 20 代之內都達到允許的誤差,則視為已收斂到最佳值;(ii)如果無法收斂,則我們將訂定最大代數,當達到最大代數時,我們便記錄最好的值。

(6)判定是否若入區域解

此步驟的目的是當步驟六未達收斂條件時,且連續幾代中最好的適應值標準差很小的話,我們即判定單體已經若入區域解(Local Optimal),因此為了避免此現象,我們則採用了步驟七的突變的方式,讓單體能夠跳脫此區域。如果也沒落入區域解,則我們將步驟四的單體全部帶入步驟二繼續演化。

(7)多單體突變(Simplex Mutation)

在單體已經落入區域解時，便需要進行突變的動作，使單體能夠跳開此區域繼續進行演算避免提早收斂，並繼續步驟二的演化。圖 3.7 為單體突變概念圖。圖中的單體經過多代的演化後形成 $a_1a_2a_3$ 及 $b_1b_2b_3$ 的單體。假設 $a_1a_2a_3$ 及 $b_1b_2b_3$ 都已經落入區域解，如果依照單體法的演算機制，是不太可能跳出該區域而找到最佳解的，因為它只會卡在那兩個位置，所以本研究必須利用突變的技術並以單體的形式來進行突變，使的單體上的某個點能夠跳離該區。所以突變後的單體變為 $a_1'a_2a_3$ 和 $b_1b_2'b_3$ 。由圖可知，突變後的點有可能會接近最佳點，這樣一來再藉由單體法的演算機制再加上本研究所設計的演化機制，就有可能朝最佳點的位置繼續搜尋。單體的點因突變所需移動的方向與步距請參考第 2.1.2 節(4)。

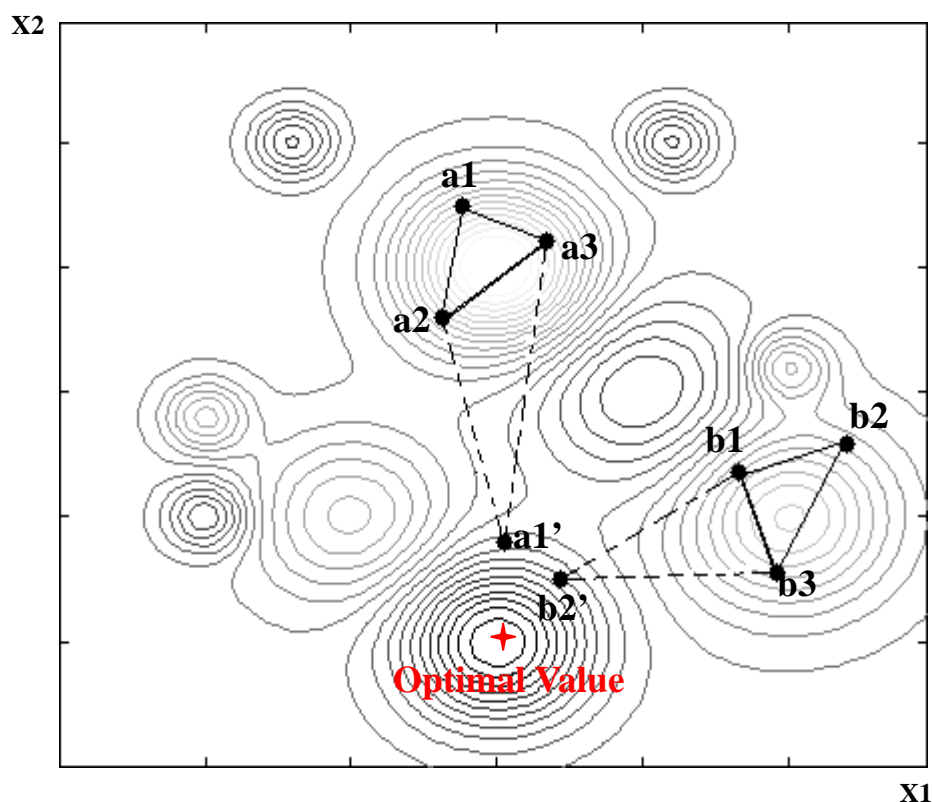


圖 3.7 單體突變概念圖

3.2 比較

雖然本研究和 Y-L[52]所提出的方法皆是將基因演算法與單體法(Simplex Method)合併，但在架構上和概念上確有很大的差異：在架構方面，Y-L 所提出的演算法強調的是以菁英化為主，但 Y-L 合併法並非完整的將單體法(Simplex Method)與基因演算法合併，該方法只是在挑選母體的過程中選出較好的母代再交由單體法來運算。因此，這樣的演算法其實就像先找出好的初始解然後再丟入單體法運算一樣，在結構上看似平行運算，雖然藉由這種方式可以搜尋更廣的空間，不過因為無法真正的運用單體法的優點，而使得搜尋的效率不佳。本研究的演算法則是在基因演算法的每個運算子(Operator)之中都保有單體的結構，且藉由多單體(Multi-simplexes)的同時運算，能搜尋更廣的空間且能達到快速收斂的優點。在概念方面，Y-L 在每一代演算完之後，子代的單體結構會被打散，並與基因演算法演化出來的子代一併做比較，Y-L 強調的是在於點對點的交換方式，而本研究的作法則強群體對群體的觀念。因為本研究認為假如在演化的過程中，群體跟群體間能存在著溝通機制的話，便能避免搜尋時間的浪費，且藉由維持原有的結構，繼續鑽入(Exploitation)區域解的深處必能比傳統的搜尋法更快找出最佳解。表 3.1 說明了兩種方法主要的差異。

表 3.1 混合式多單體 GA 協同搜尋法與混合型基因演算法的差異

比較點	HMSGGA-CS	Y-L 的演算法
架構	以協同為主並建立在單體法機制上 (Cooperative-based)	以菁英化為主 (Elite-based)
是否維持單體結構	是	否
單體法的形式	Nelder-Mead	機率型
單體法是否嵌入基因演算法中	是	否

每代單體個數	多單體	一個單體
菁英化的方式	以單體為主	以點為主
母體產生方式	隨機	排序
每代反射的點數	$\frac{P}{N+1} \times p_s$	S-N

註：P - 母體數目

N - 變數個數

P_s - 菁英化比例

S - 在母體排序後最好的 S 條染色體

第四章 實驗驗證

本章節將依第三章所提出的研究架構，分別用二進位與十進位的編碼方式做一探討，並以 15 個多變數函數範例[29]來測試實驗結果，實驗結果也將與由 Y-L 所提出的架構及基因演算法一併做比較。我們將隨機挑選兩個範例來針對部分重要的參數作實驗設計，此參數包括：資料型態、菁英化單體數目以及單體演化次數等，用以探討這些參數對我們求解效率的影響程度。

4.1 多變數函數

算例(一)與算例(二)為 Y-L 在文獻[52]中所測試的範例。在該文獻中除了以此二例來實驗驗證 Y-L 所提出的架構外，也與傳統的基因演算法(GA)、適合的模擬退火法(Adaptive Simulated Annealing,ASA)以及 G 位元改善法(G-bit Improvement)做一比較。

本研究將根據程式執行的結果，以下列幾項指標做為我們的比較點：

- (1)收斂百分比(Convergence Percentage)：此指標的計算方式為：
能夠達到收斂條件的次數/設定的測試次數)*100%。
- (2)達收斂時的平均代數：在 30 次的測試中，將達收斂條件的所
- (3)有次數的代數取平均值。
- (4)達收斂時的平均代數之標準差：在(2)中計算標準差。
- (5)達收斂時之平均計算時間：在 30 次的測試中，將達收斂條件的次數計算其演算時間，並取平均值。
- (6)達收斂時之平均計算時間之標準差：在(4)中計算標準差。
- (7)平均值：將每次測試後的最好的值取平均值。
- (8)標準差：將(6)計算標準差。
- (9)正確率(Accuracy)：介於理想值(Ideal Value)與(6)的平均絕對誤差值(Root Mean Squared Error)。
- (10)顯著水準 = 0.05。

4.1.1 算例(一)：De Jong function F5

(1)目標函數

目標函數如(4-1)所示，此函數目的在求此函數的最小值，圖 4. 1 為此目標函數之圖形。

$$\text{Min. } F(X) = \frac{1}{0.002 + \sum_{j=0}^{24} \frac{1}{j + \sum_{i=0}^1 (X_i - a_{ij})}} \quad (4-1)$$

where $X_i \in [-65.536, 65.536]$

$$a_{0,j} = -32, -16, 0, 16, 32 \quad \text{for } j=0,1,2,3,4$$

$$a_{0,j} = a_{0,j \bmod 5} \quad \text{for } j=5,6,\dots,24$$

$$a_{1,j} = -32, -16, 0, 16, 32 \quad \text{for } j=0,5,10,15,20$$

$$a_{1,j} = a_{1,j+k} \quad \text{for } k=1,2,3,4$$

此函數之理想(Ideal)最佳值為 0.998004。此函數有 25 個區域最佳解。

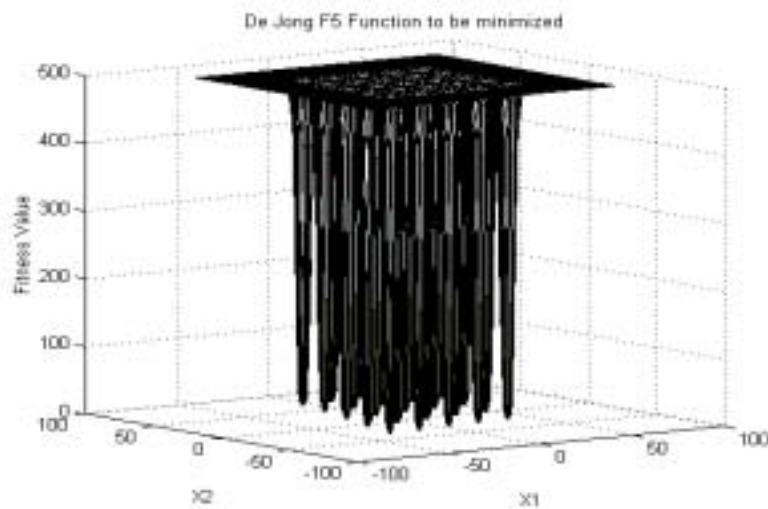


圖 4. 1 De Jong F5 之圖形

(2)測試環境與變數設定

此研究之相關變數如下：

- a. 族群大小：在我們架構的實驗中，我們的母體必須為 $N+1$ 的倍數，其中 N 為變數個數，所以我們取接近文獻[52]中所設的母體數目為 42，而在文獻中所設定的母體大小為 44。
- b. 單一變數之基因數：在文獻中由於並未詳細記載，因此我們以精

確度為 10^5 來計算，因此，以此例來說，單一變數之染色體長度為 24。

- c. 複製法則：輪盤法(Roulette)。
- d. 交配法則：以二進位制為編碼方式的交配方式為雙點交配。交配機率 $P_c=0.25$ ，與文獻[52]相同設定值；以十進位為編碼方式的交配方式為算術交配法(Arithmetic Crossover)。
- e. 突變法則：以二進位制為編碼方式的突變為參考 2.1 節。突變機率 $P_m=0.389$ ，與文獻[52]相同設定值；以十進位為編碼方式的突變方式為均勻突變法(Uniform Mutation)。
- f. 演化代數：在本範例中，我們取最大代數為 3000。
- g. 收斂條件：在本範例中，當與最佳值的誤差小於 10^{-6} 時且連續 20 代皆如此，則視為收斂。
- h. 測試次數：30 次。
- i. 參數設定：在本範例中，菁英化單體數目為兩個；單體演化次數為 5 次、10 次和 15 次。

(3)實驗結果與比較

圖 4. 2 至圖 4. 8 以及表 4. 1 為本研究實驗的結果。根據變異數分析(ANOVA)的結果(請參考附錄一表 A1. 1 及表 A1. 3), 方法的不同對收斂代數有顯著的影響且方法的不同對收斂時間有顯著的影響。

在表 4. 1 中，本研究所提出的方法 HMSGGA-CS 在『收斂百分比』的欄位中皆達 100%的比例，顯示 HMSGGA-CS 是穩定的方法。

在『平均收斂代數』方面，所有的組合也皆比 Y-L 及 GA 的好且穩定不少，值得注意的是在 Real Value 的資料型態上，單體演化次數的不同並不會對『平均收斂代數』有顯著的影響($p\text{-value}=0.0813>$ ，請參考附錄一表 A1. 2)。

在『平均收斂時間』方面，所有的組合也皆比 Y-L 及 GA 的方法快了許多，且 HMSGGA-CS 在不同的組合上皆對『平均收斂時間』沒有顯著影響($p\text{-value}=0.3585>$ ，請參考附錄一表 A1. 4)。

表 4. 2 為文獻[52]中所測試的實驗數據。以十進位(Real Value)為主的基因演算法在 Y-L 的實驗中其『平均收斂代數』遠大於其他三種

演算法，而其所提出的演算法在此範例中也並沒有比模擬退火法 (ASA) 來的好。根據 Y-L 所說，他們認為模擬退火法並不適合在此一問題的求解上，只因為它的區域解不夠多而已。對於本研究將此例重新演算後的結果雖然也比 Y-L 的好，不過因為 Y-L 的『收斂代數的標準差』過大且在 Y-L 的實驗中只測試 10 次，所以可將其結果視為相近。

表 4.1 測試結果表(算例一)

Method		Convergence Percentage	Average* Generation	Standard Deviation (Generation)	Mean* Time(sec.) (Convergence)	Mean Value	Accuracy	
HMSGA -CS	Binary	Nds=5	100%	26.967	7.009	1.909	0.9980039	3.650E-08
		Nds=10	100%	23.533	2.029	1.304	0.9980038	8.815E-08
		Nds=15	100%	22.167	0.791	1.085	0.9980038	9.657E-08
	Real Value	Nds=5	100%	35.367	26.677	1.923	0.9980038	1.010E-07
		Nds=10	100%	30.621	26.320	2.389	0.9980038	1.053E-07
		Nds=15	100%	22.833	1.683	0.905	0.9980038	1.125E-07
Y-L (Binary)		56.67%	168.2353	120.3477	39.3653	1.064972	4.74E-02	
Y-L (Real Value)		100%	563	714.98	46.3057	0.998004	4.24E-07	
GA (Binary)		0%	1000	0	110.6535	1.0120	9.9E-03	
GA (Real Value)		0%	1000	0	48.8918	1.0591	4.32E-02	

*代表只計算有收斂的數值。當『收斂百分比』為 0 時，『平均收斂時間』取演算 1000 代的平均時間。

表 4.2 文獻中之測試結果表(算例一)

Method	Average Generation
GA	12,103
Y-L's Hybrid(Real Value)	1,695
G-bit Improvement	6,961
ASA	1,065

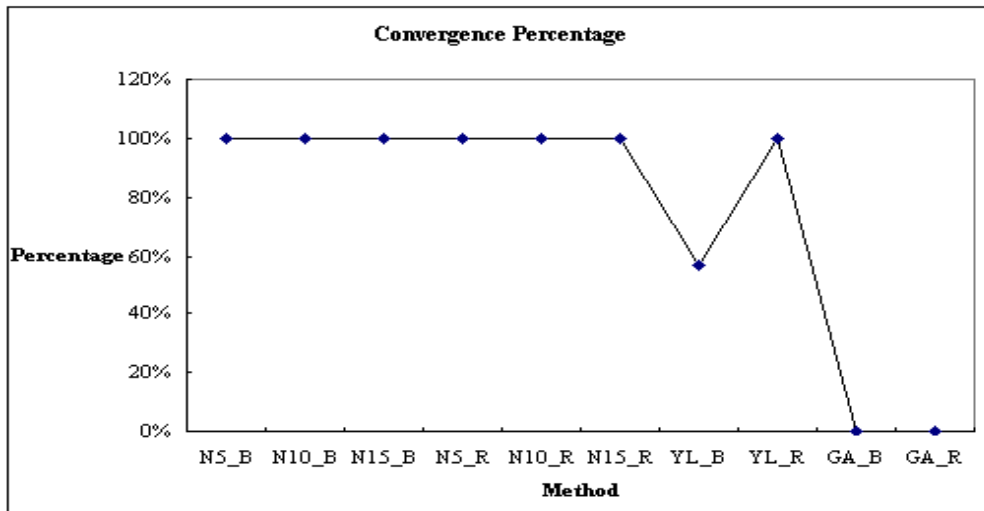


圖 4.2 收斂率比較圖(算例一)

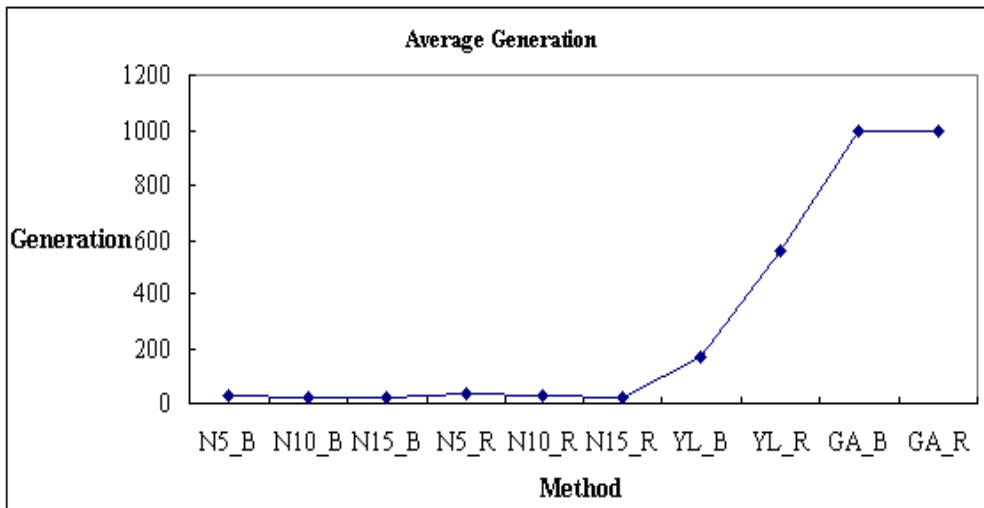


圖 4.3 平均收斂代數比較圖(算例一)

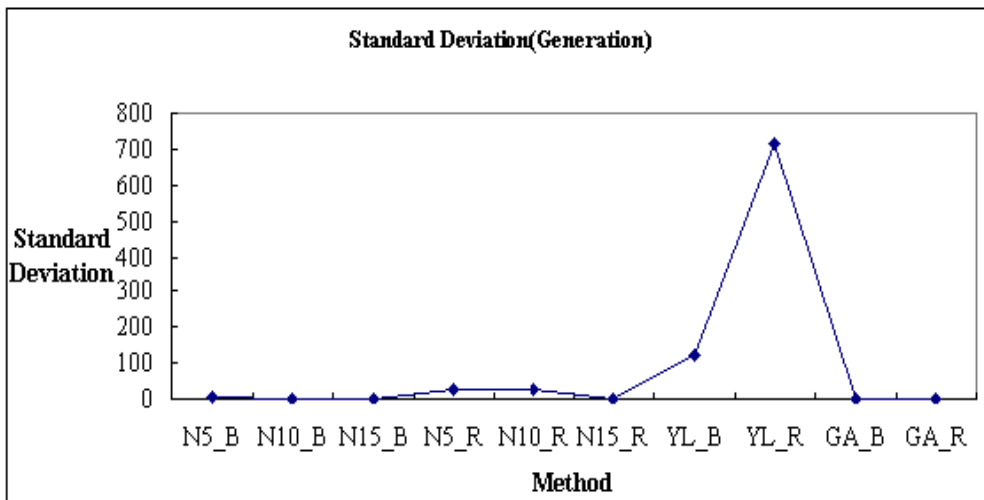


圖 4.4 平均收斂代數之標準差比較圖(算例一)

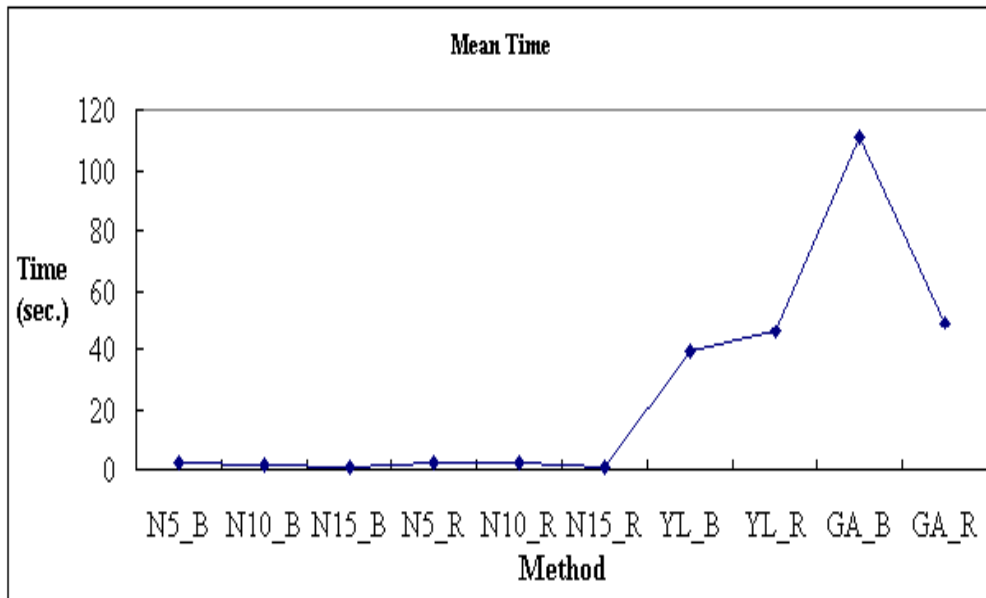


圖 4.5 達收斂時之平均運算時間比較圖(算例一)

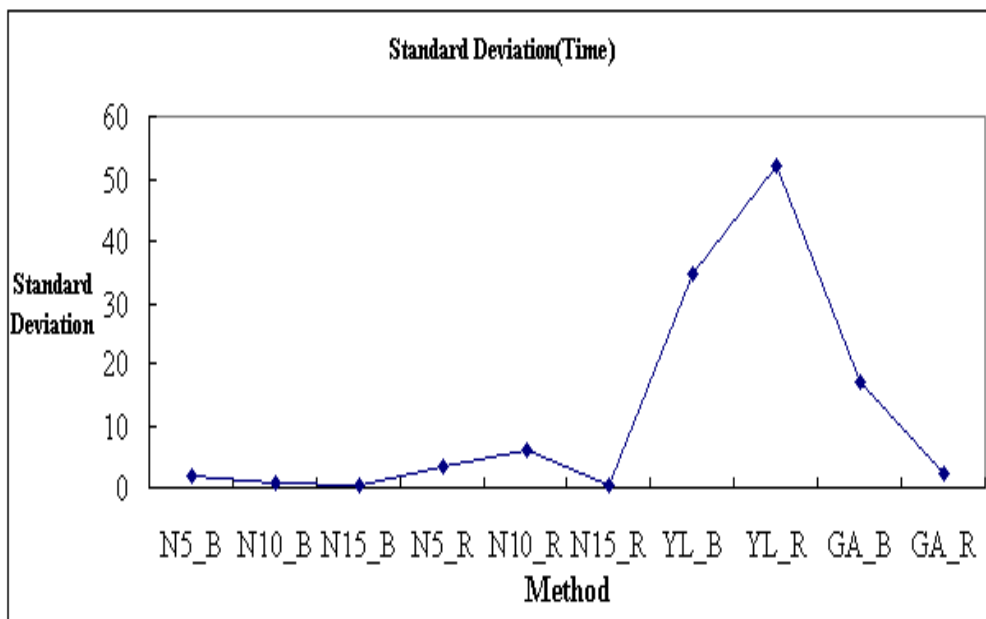


圖 4.6 平均運算時間之標準差比較圖(算例一)

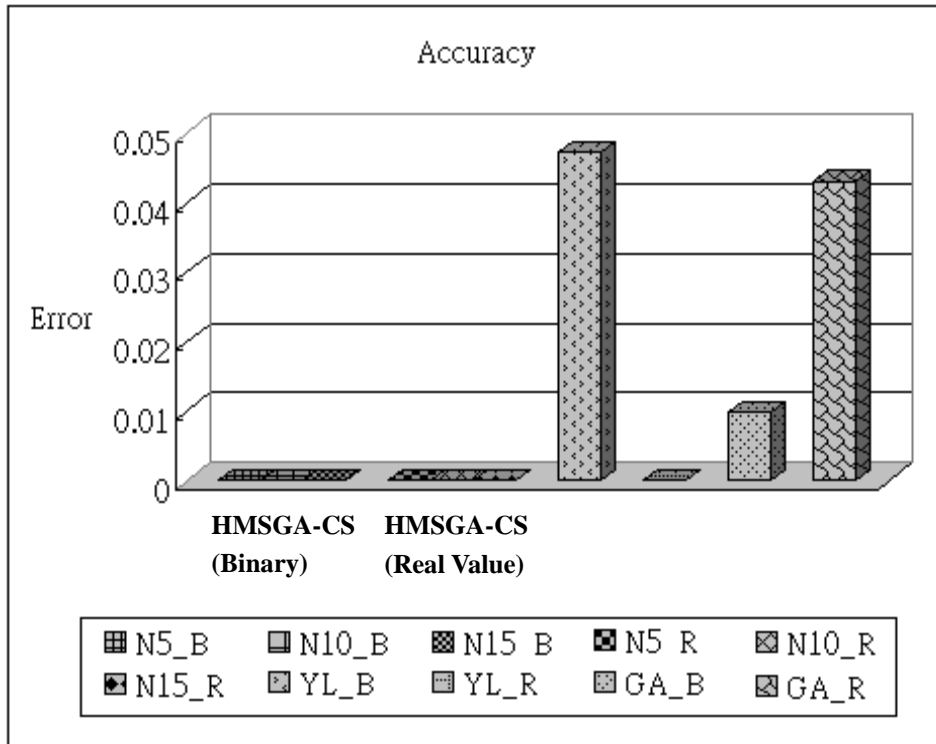


圖 4.7 與理想值之誤差比較圖(算例一)

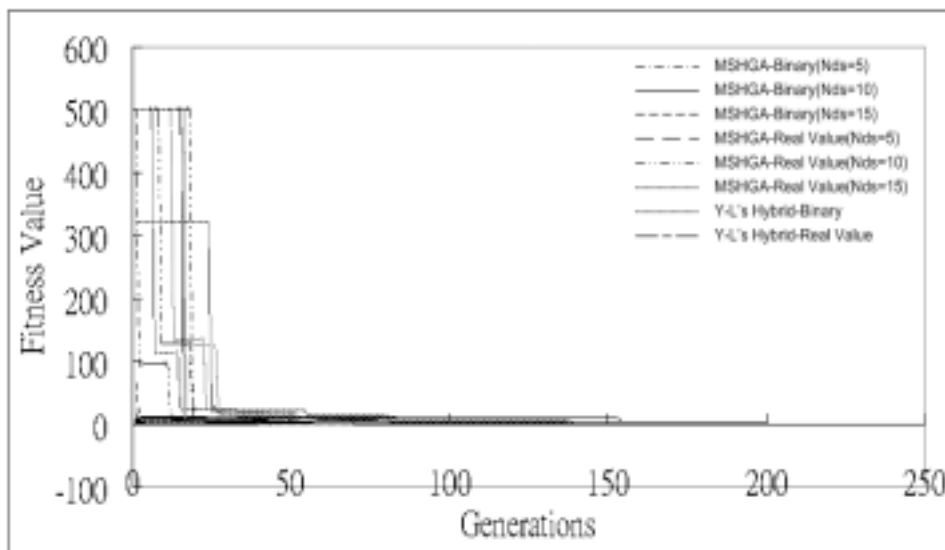


圖 4.8 代數收斂表示圖(算例一)

4.1.2 算例(二) : Sin function

(1) 目標函數

目標函數如(4-2)所示，此函數目的在求此函數的最大值，圖 4.9 為此目標函數之二維圖形。

$$\text{Max. } F(X) = \sum_{i=1}^N \sin(x_i) \sin^{2m} \left(\frac{i \times x_i^2}{\pi} \right) \quad (4-2)$$

where $x_i \in [0, \pi]$, $m=10$ or 100 , $N=10$

此函數之理想(Ideal)最佳值在 $m=10$ 時為 9.66 ; $m=100$ 時為 9.655。此函數有 $N!$ 個區域解。

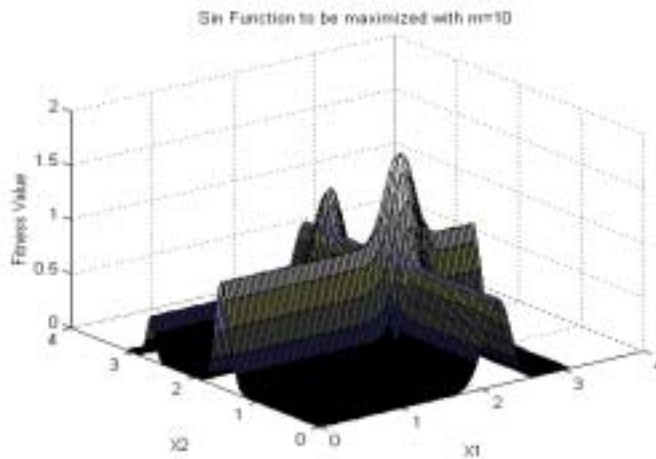


圖 4.9 Sin 函數之圖形

(2)測試環境與變數設定

此研究之相關變數如下：

- a. 族群大小：與文獻中所設定的母體大小一致為 44。
- b. 單一變數之基因數：在文獻中由於並未詳細記載，因此我們以精確度為 10^5 來計算，因此，以此例來說，單一變數之染色體長度為 19。
- c. 複製法則：輪盤法(Roulette)。
- d. 交配法則：以二進位制為編碼方式的交配方式為雙點交配。交配機率 $P_c = 0.4$ ；以十進位為編碼方式的交配方式為算術交配法(Arithmetic Crossover)。
- e. 突變法則：以二進位制為編碼方式的突變為參考 2.1 節。突變機率 $P_m = 0.03$ ；以十進位為編碼方式的突變方式為均勻突變法(Uniform Mutation)。
- f. 演化代數：在本範例中，我們取最大代數為 40000。

- g. 收斂條件：在本範例中，當與最佳值的誤差小於 0.01 時且連續 20 代皆如此，則視為收斂。
- h. 測試次數：10 次。
- i. 參數設定：在本範例中，菁英化單體數目為兩個；單體演化次數為 5 次、10 次和 15 次。

(3)實驗結果與比較

圖 4. 10 至圖 4. 15 以及表 4. 3 為本研究實驗的結果。根據變異數分析(ANOVA)的結果(請參考附錄一表 A1. 5 及表 A1. 6), 方法的不同對收斂代數有顯著的影響且方法的不同對收斂時間有顯著的影響。

在表 4. 3 中，所有方法的『收斂百分比』皆不高，在 HMSGGA-CS 的 Real Value 與 Nds = 5 的組合中反而有較高的比例。從此表中可看出實數型(Real Value)的方法不論在『平均演算時間』或『平均收斂代數』上均比 Binary 的型態來的好。

在『平均收斂代數』方面，HMSGGA-CS 比其它兩種方法來的好許多。且經變異數分析得知，在 HMSGGA-CS 的 Real Value 資料型態上，單體演化次數的不同對『平均收斂代數』不會有顯著影響(p-value=0.3884>)。同樣的，Binary 的資料型態對『平均收斂代數』也不會有顯著的影響(p-value=0.2718>)。

在『平均演算時間』方面，HMSGGA-CS 也比其它兩種方法來的快許多。經變異數分析得知，在 HMSGGA-CS 的 Real Value 資料型態上，單體演化次數的不同對『平均演算時間』不會有顯著影響(p-value=0.1744>)。Binary 的資料型態對『平均演算時間』也不會有顯著影響(p-value=0.1317>)。

表 4.3 測試結果表(算例二)

Method		Convergence Percentage	Average* Generation	Standard Deviation (Generation)	Mean* Time(sec.) (Convergence)	Mean Value	Accuracy	
HMSGA -CS	Binary	Nds=5	40%	16033	2671.4	1761.7	9.6498	7.193E-03
		Nds=10	20%	21645	2122.7	2140.9	9.6466	9.462E-03
		Nds=15	20%	23007	2790.9	2477.9	9.6507	6.549E-03
	Real Value	Nds=5	70%	10046	1635.4	812.6	9.6581	1.369E-03
		Nds=10	50%	12585	1910.9	1139.3	9.6566	2.437E-03
		Nds=15	50%	14252	1747.6	1287.1	9.6565	2.502E-03
Y-L (Binary)		0%	40000	0	5803.8	9.5978	4.400E-02	
Y-L (Real Value)		40%	32980	2660.3	2550.9	9.6571	2.079E-03	
GA (Binary)		0%	40000	0	20944.1	8.8734	5.562E-01	
GA (Real Value)		0%	40000	0	14368.7	9.0947	3.997E-01	

*代表只計算有收斂的數值

表 4.4 文獻中之測試結果表(算例二)

Approach	m=10	
	Average trials to reach the optimum	Runs that found the optimum
GA	315,501	10
Y-L 57% Simplex-GA Hybrid	37,307	10
G-bit Improvement	97,174	10
ASA	184,426	8

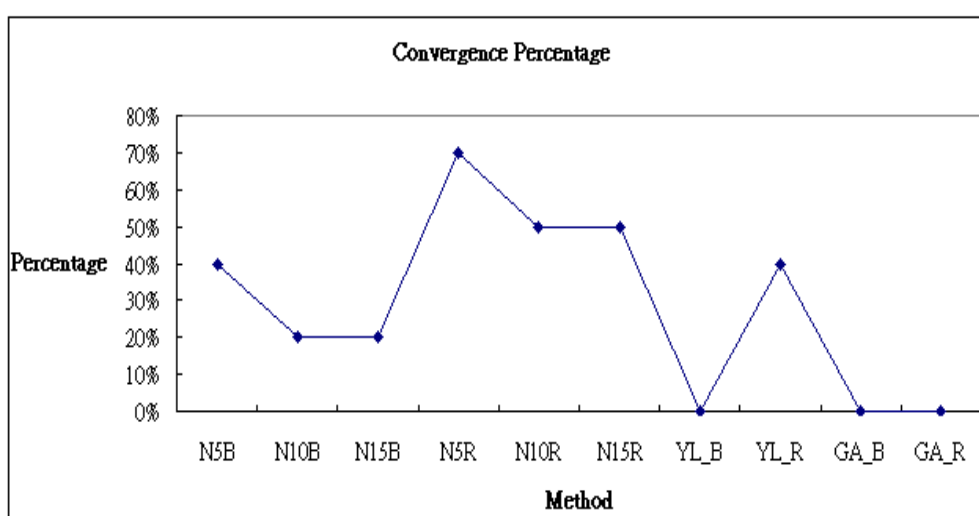


圖 4.10 收斂率比較圖(算例二)

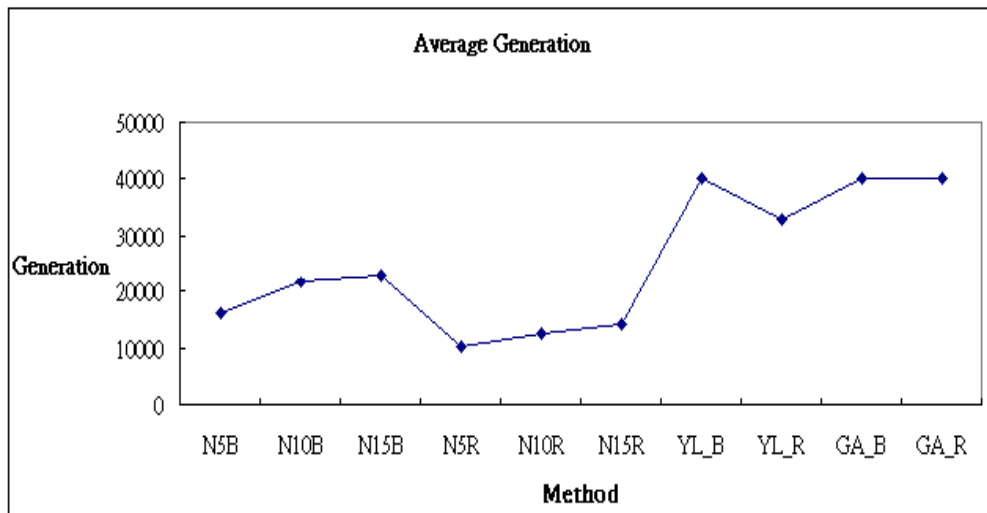


圖 4.11 平均收斂代數比較圖(算例二)

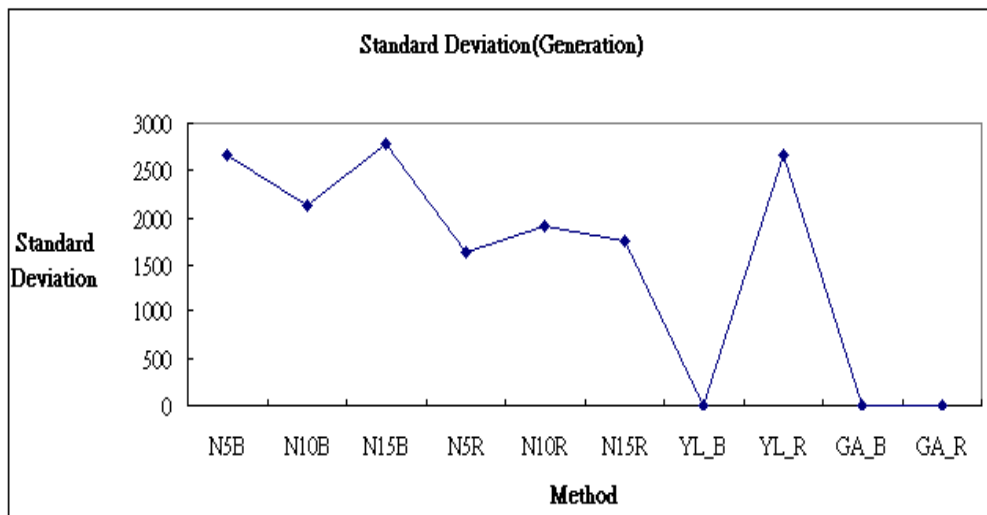


圖 4.12 平均收斂代數之標準差比較圖(算例二)

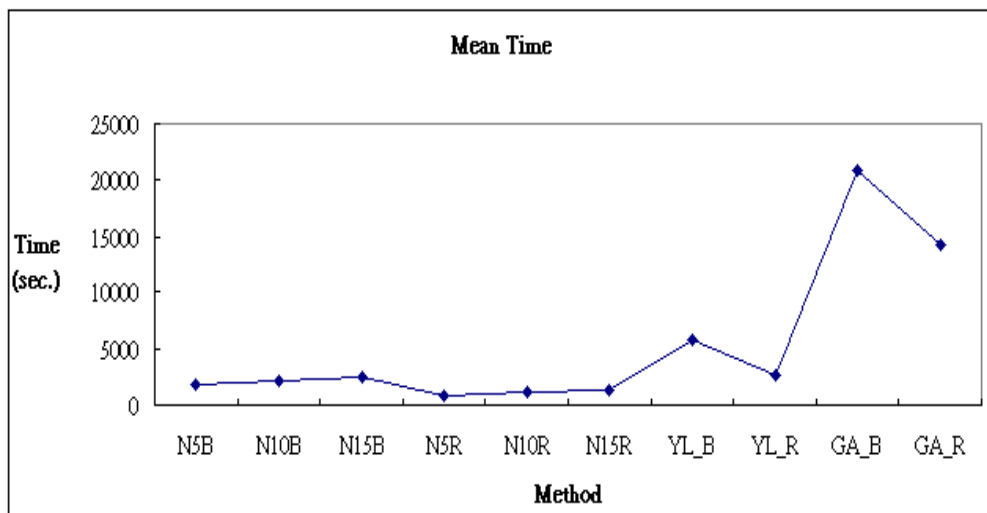


圖 4.13 達收斂時之平均運算時間比較圖(算例二)

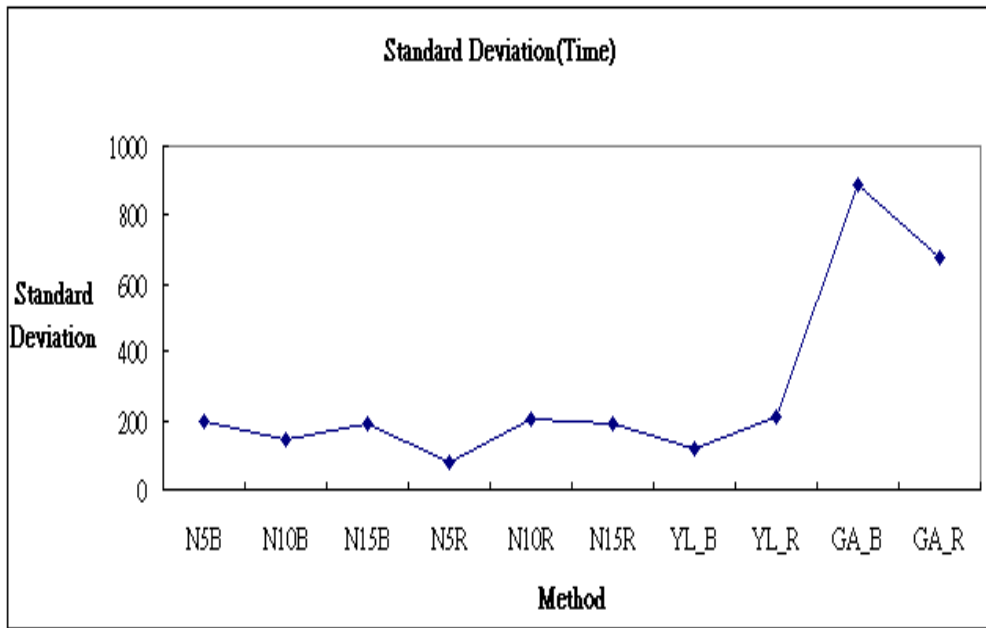


圖 4.14 平均運算時間之標準差比較圖(算例二)

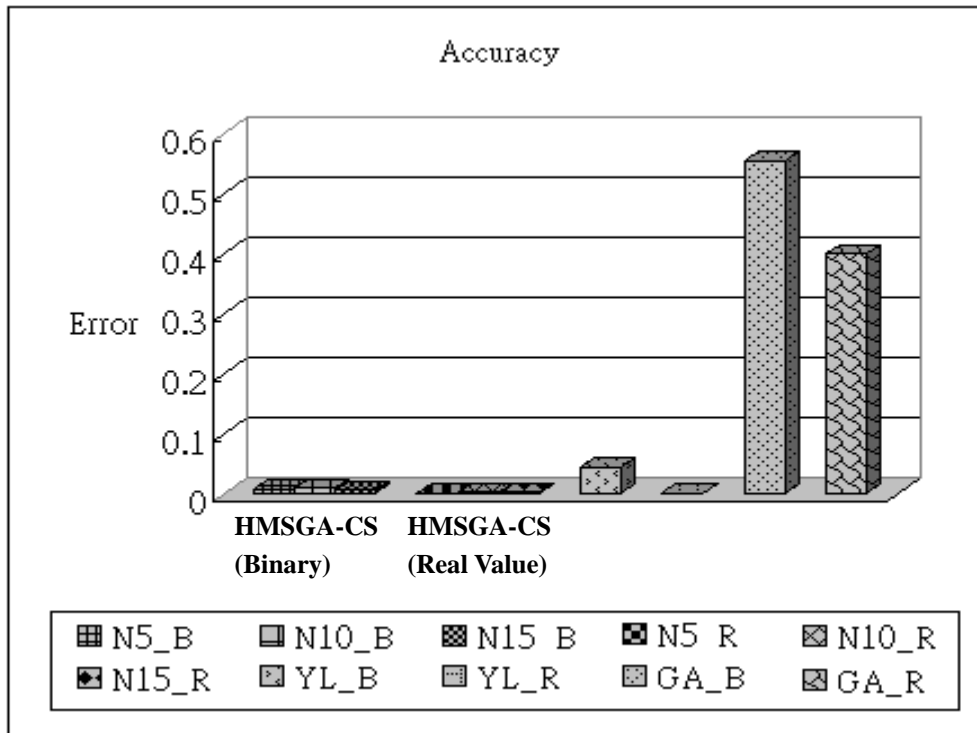


圖 4.15 與理想值之誤差比較圖(算例二)

4.1.3 算例(三) : De Jong function F1

目標函數如(4-3)所示，此函數目的在求此函數的最小值，圖 4.16

為此目標函數之二維圖形。

$$\text{Min. } F(X) = \sum_{i=1}^3 x_i^2 \quad (4-3)$$

where $-5.12 \leq x_i \leq 5.12$

此函數之理想(Ideal)最佳值為 0。

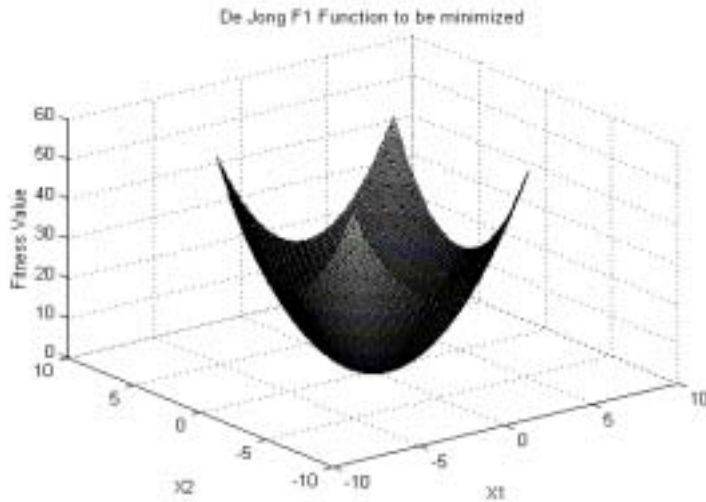


圖 4. 16 De Jong F1 之圖形

(2)測試環境與變數設定

此研究之相關變數如下：

- a. 族群大小：本範例母體大小為 44。
- b. 單一變數之基因數：單一變數之染色體長度為 20，精確度為 10^5 。
- c. 複製法則：輪盤法(Roulette)。
- d. 交配法則：以二進位制為編碼方式的交配方式為雙點交配。交配機率 $P_c = 0.4$ ；以十進位為編碼方式的交配方式為算術交配法 (Arithmetic Crossover)。
- e. 突變法則：以二進位制為編碼方式的突變為參考 2.1 節。突變機率 $P_m = 0.03$ ；以十進位為編碼方式的突變方式為均勻突變法 (Uniform Mutation)。
- f. 演化代數：在本範例中，我們取最大代數為 1000。
- g. 收斂條件：在本範例中，當與最佳值的誤差小於 0.001 時且連續 20 代皆如此，則視為收斂。

h. 測試次數：30 次。

i. 參數設定：在本範例中，菁英化單體數目取 0%、50%和 100% 來比較；單體演化次數為 5 次、10 次和 15 次。

(3) 實驗結果

圖 4. 17 至圖 4. 23 以及表 4. 5 為本研究取菁英化數目為兩個單體的實驗結果。根據變異數分析(ANOVA)的結果(請參考附錄一表 A1. 7 及表 A1. 8)，方法的不同對收斂代數有顯著的影響且方法的不同對收斂時間有顯著的影響。

在表 4. 5 中，雖然所有的組合在 30 的實驗中皆滿足收斂條件，但 HMSGGA-CS 的方法在『平均收斂代數』方面卻比 Y-L 及 GA 好很多，且經變異數分析得知在二進位制(Binary)的 HMSGGA-CS 中，單體演化的次數不同對結果並無顯著的差異($p\text{-value}=0.083>$)；在『平均收斂時間』方面，HMSGGA-CS 在各組合上所需的求解時間皆比 Y-L 及 GA 的時間少。另外，值得注意的是在 Binary 的資料型態上，單體演化次數的不同並不會對『平均收斂時間』有顯著的差異($p\text{-value}=0.274>$)，請參考附錄一表 A1. 9)。

在本範例中，本研究將以實驗設計來驗證參數對收斂代數及計算時間的影響。表 4. 6 本研究的測試組合表與數據。

表 4. 5 測試結果表(算例三)

Method			Convergence Percentage	Average* Generation	Standard Deviation (Generation)	Mean* Time(sec.) (Convergence)	Mean Value	Accuracy
HMSGGA-CS	Binary	Nds=5	100%	23.6000	1.4541	0.8064	0.0005	4.964E-03
		Nds=10	100%	22.9333	1.6242	0.7136	0.0003	9.151E-04
		Nds=15	100%	22.4000	1.1832	0.6630	0.0002	5.539E-04
	Real Value	Nds=5	100%	23.6923	0.9473	0.1634	0.0003	3.811E-04
		Nds=10	100%	23.2000	1.5213	0.2309	0.0002	4.708E-04
		Nds=15	100%	22.5333	0.8338	0.2543	0.0002	5.539E-04
Y-L (Binary)			100%	125.8	22.740	16.469	0.00058	4.482E-04
Y-L (Real Value)			100%	229.8	229.4736	8.7953	0.00046	3.644E-04
GA (Binary)			100%	40	6.4219	1.5412	0.00066	4.68E-04
GA (Real Value)			90%	457.89	247.448	13.481	0.00079	5.6E-04

*代表只計算有收斂的數值

表 4.6 參數實驗設計組合表(算例三)

變數 組別	資料型態	菁英化單體數目	單體演化次數	收斂代數	時間(秒)
第 1 組	Binary	0%	Nds=5	27	2.083
第 2 組				30	1.743
第 3 組				40	4.326
第 4 組				38	2.393
第 5 組				29	1.793
第 6 組				33	2.383
第 7 組				26	1.543
第 8 組				35	4.656
第 9 組				33	0.831
第 10 組				37	1.232
第 11 組			Nds=10	49	5.788
第 12 組				32	2.333
第 13 組				62	8.262
第 14 組				30	1.933
第 15 組				25	0.971
第 16 組				23	0.631
第 17 組				26	1.192
第 18 組				38	3.445
第 19 組				28	1.602
第 20 組				23	0.571
第 21 組			Nds=15	60	9.303
第 22 組				62	9.944
第 23 組				44	5.618
第 24 組				29	1.953
第 25 組				24	0.932
第 26 組				145	29.282
第 27 組				28	1.802
第 28 組				27	1.643
第 29 組				53	7.56
第 30 組				24	0.912
第 31 組	50%	Nds=5	35	3.485	
第 32 組			27	1.633	
第 33 組			25	1.141	
第 34 組			36	3.726	
第 35 組			25	1.141	
第 36 組			25	1.152	
第 37 組			29	2.243	
第 38 組			28	2.063	

第 39 組				25	1.202
第 40 組				25	1.181
第 41 組			Nds=10	25	1.342
第 42 組				24	1.041
第 43 組				22	0.601
第 44 組				24	1.532
第 45 組				22	0.511
第 46 組				22	0.801
第 47 組				25	1.662
第 48 組				33	3.656
第 49 組				24	1.121
第 50 組				23	0.751
第 51 組			Nds=15	22	0.651
第 52 組				22	1.192
第 53 組				22	0.571
第 54 組				23	0.921
第 55 組				23	0.841
第 56 組				39	6.55
第 57 組				23	1.111
第 58 組				26	2.073
第 59 組				22	0.761
第 60 組				23	0.952
第 61 組		ALL	Nds=5	702	186.479
第 62 組				503	100.24
第 63 組				201	48.177
第 64 組				410	95.107
第 65 組				304	87.24
第 66 組				57	18.942
第 67 組				29	6.359
第 68 組				403	93.775
第 69 組				84	32.124
第 70 組				962	257.911
第 71 組			Nds=10	415	111.751
第 72 組				37	4.246
第 73 組				85	21.471
第 74 組				152	33.458
第 75 組				45	1.252
第 76 組				64	10.936
第 77 組				112	23.093
第 78 組				50	6.145
第 79 組				72	12.435

第 80 組				52	3.724
第 81 組			Nds=15	22	0.561
第 82 組				30	2.421
第 83 組				23	0.831
第 84 組				35	4.126
第 85 組				40	4.824
第 86 組				485	144.047
第 87 組				24	1.122
第 88 組				33	3.625
第 89 組				28	2.137
第 90 組				180	52.085
第 91 組	Real Value	None		Nds=5	25
第 92 組			22		0.11
第 93 組			22		0.11
第 94 組			23		0.14
第 95 組			23		0.15
第 96 組			23		0.141
第 97 組			24		0.11
第 98 組			22		0.07
第 99 組			25		0.18
第 100 組			22		0.11
第 101 組			Nds=10	23	0.3
第 102 組				23	0.261
第 103 組				22	0.2
第 104 組				23	0.26
第 105 組				22	0.201
第 106 組				24	0.35
第 107 組				22	0.19
第 108 組				23	0.261
第 109 組				25	0.44
第 110 組				22	0.181
第 111 組			Nds=15	22	0.25
第 112 組				24	0.431
第 113 組				23	0.361
第 114 組				21	0.1
第 115 組				24	0.45
第 116 組				23	0.301
第 117 組				23	0.551
第 118 組				22	0.19
第 119 組				22	0.21
第 120 組				23	0.381

第 121 組				46	1.001
第 122 組				23	0.11
第 123 組				23	0.121
第 124 組				23	0.11
第 125 組			Nds=5	25	0.18
第 126 組				24	0.16
第 127 組				25	0.191
第 128 組				24	0.15
第 129 組				25	0.19
第 130 組				23	0.11
第 131 組				22	0.15
第 132 組				23	0.19
第 133 組				22	0.14
第 134 組				22	0.14
第 135 組			Nds=10	24	0.261
第 136 組		50%		24	0.26
第 137 組				23	0.19
第 138 組				23	0.201
第 139 組				23	0.19
第 140 組				22	0.13
第 141 組				25	0.501
第 142 組				22	0.18
第 143 組				56	3.345
第 144 組				22	0.19
第 145 組			Nds=15	21	0.09
第 146 組				22	0.191
第 147 組				22	0.19
第 148 組				22	0.19
第 149 組				23	0.271
第 150 組				21	0.1
第 151 組		100%		24	0.2
第 152 組				28	0.33
第 153 組				31	0.441
第 154 組				23	0.12
第 155 組			Nds=5	24	0.17
第 156 組				24	0.161
第 157 組				25	0.21
第 158 組				23	0.13
第 159 組				24	0.16
第 160 組				22	0.1
第 161 組			Nds=10	22	0.16

第 162 組				24	0.35
第 163 組				24	0.27
第 164 組				24	0.281
第 165 組				23	0.22
第 166 組				23	0.21
第 167 組				21	0.061
第 168 組				21	0.08
第 169 組				21	0.07
第 170 組				24	0.27
第 171 組			Nds=15	22	0.21
第 172 組				22	0.201
第 173 組				23	0.28
第 174 組				30	0.941
第 175 組				22	0.191
第 176 組				22	0.19
第 177 組				24	0.391
第 178 組				22	0.19
第 179 組				21	0.09
第 180 組				21	0.1

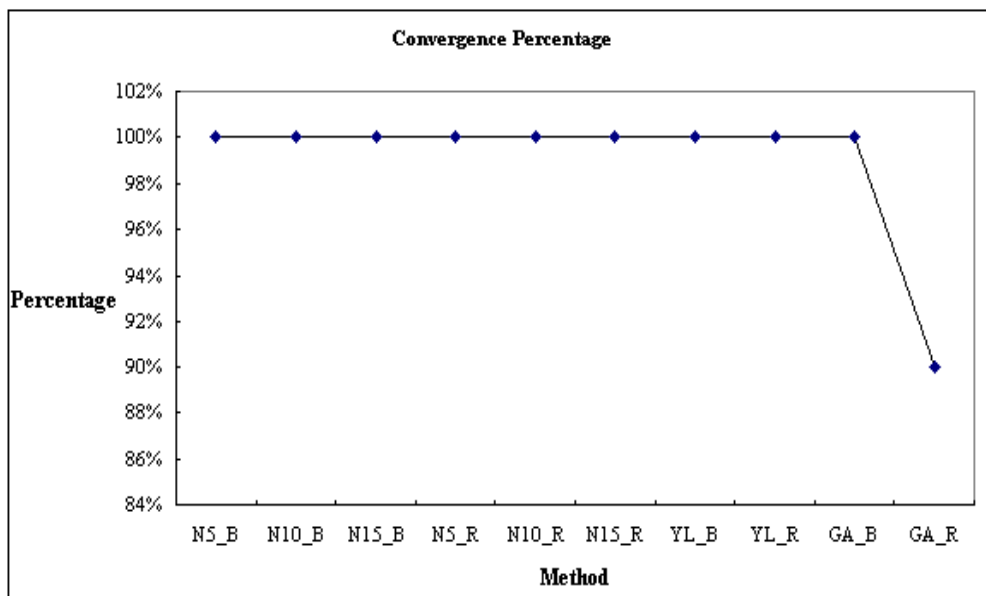


圖 4.17 收斂率比較圖(算例三)

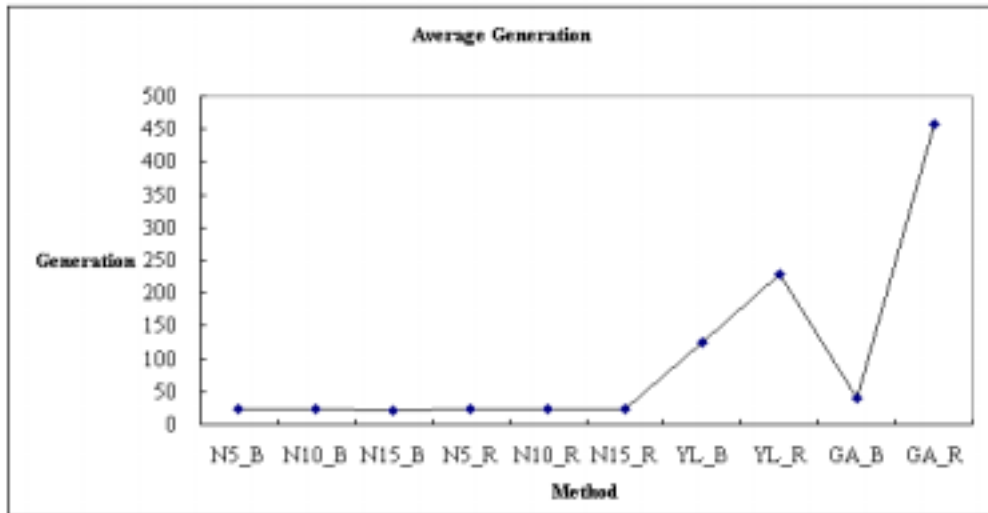


圖 4.18 平均收斂代數比較圖(算例三)

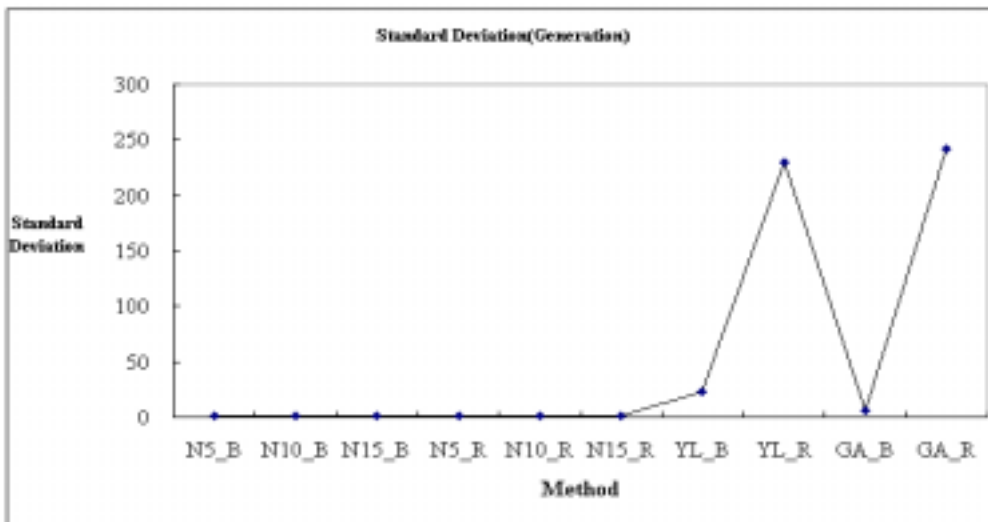


圖 4.19 平均收斂代數之標準差比較圖(算例三)

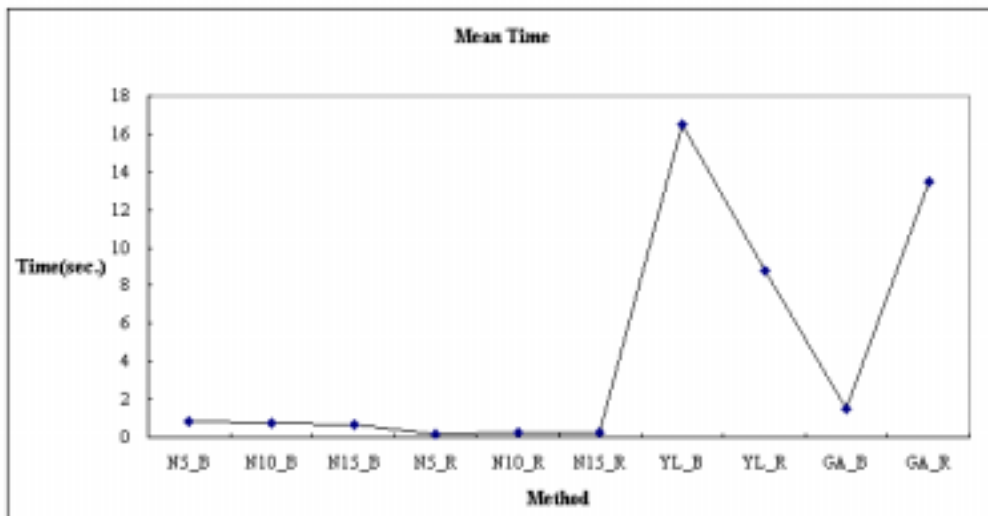


圖 4.20 達收斂時之平均運算時間比較圖(算例三)

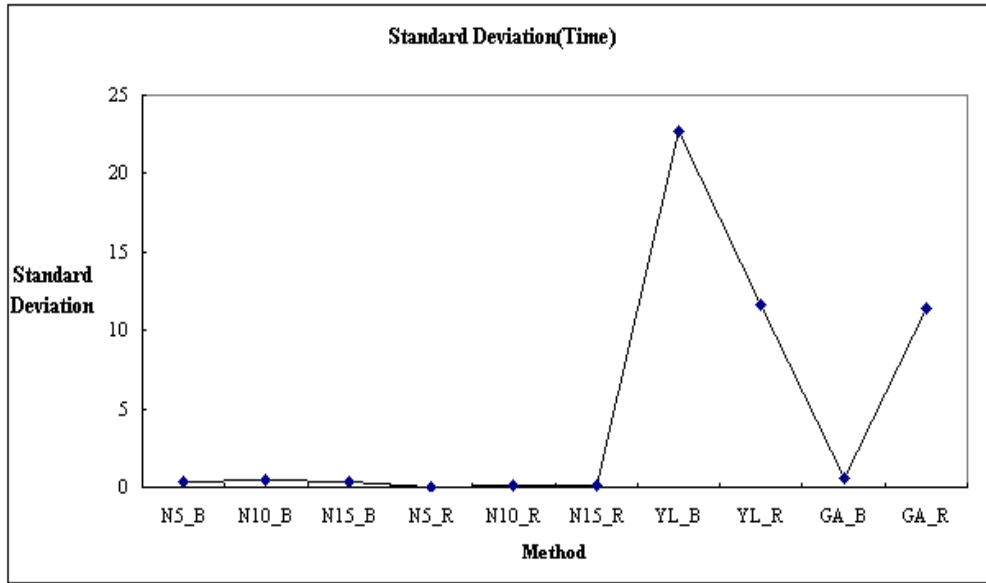


圖 4.21 平均運算時間之標準差比較圖(算例三)

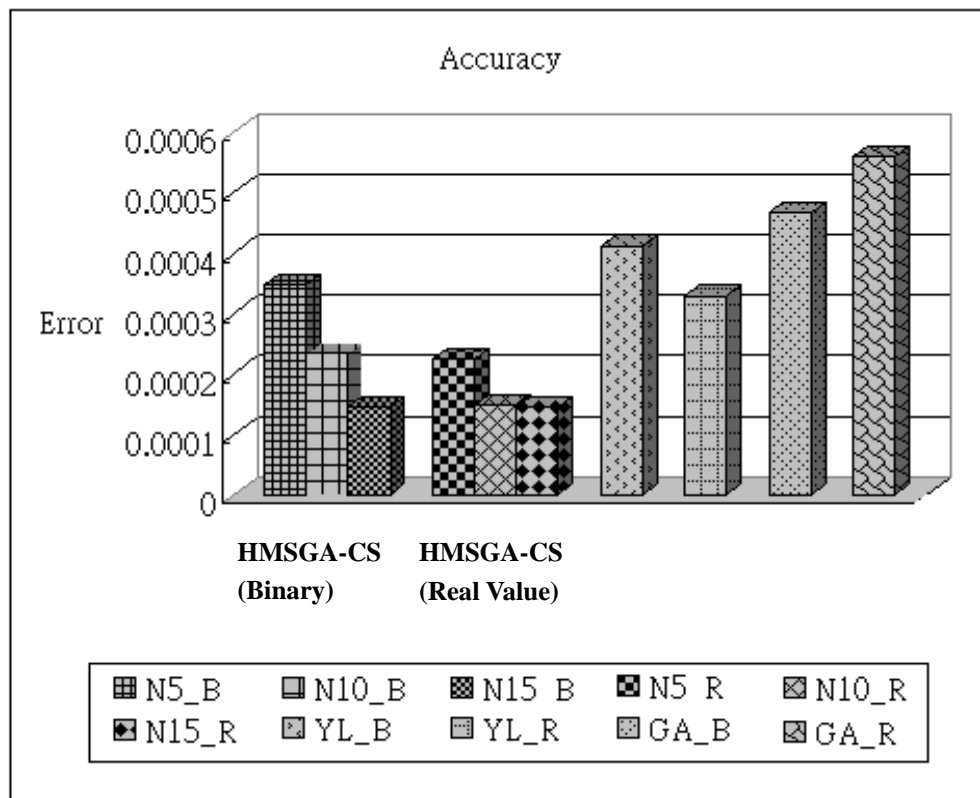


圖 4.22 與理想值之誤差比較圖(算例三)

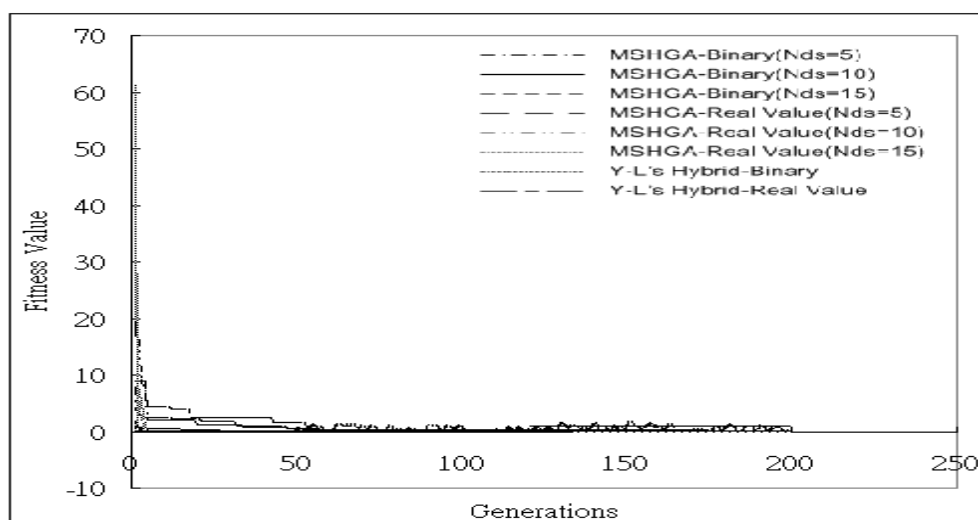


圖 4.23 代數收斂表示圖(算例三)

(4)實驗數據分析

1.收斂代數分析及探討

表 4.7 為針對各參數對收斂代數的數據分析結果。在分析方面，我們取顯著水準 $\alpha=0.05$ 來檢定是否有顯著關係。

表 4.7 收斂代數之變異數分析表(算例三)

變異來源	自由度	均方	誤差均方	F 值	P-value
資料型態	1	46980.36	12933.96	3.632326	0.058439
菁英化數目	2	98251.32	12933.96	7.596384	0.000702
單體演化次數	2	91995.05	12933.96	7.112675	0.001094
資料型態 vs 菁英化數目	2	214521.5	12933.96	16.58591	2.8E-07
資料型態 vs 單體演化次數	2	8161.839	12933.96	0.63104	0.533341
菁英化數目 vs 單體演化次數	4	29093.64	12933.96	2.249399	0.066021
資料型態 vs 菁英化數目 vs 單體演化次數	4	75045.87	12933.96	5.802235	0.000217
誤差值	162				

首先，先就單因子來探討對收斂代數的影響：

a. 資料型態的不同對收斂代數沒有顯著的影響。

在表 4.8 中，雖然對於不同資料型態的平均代數有些許的差異，但在統計的信心水準內則可視為是相近的。

b. 菁英化數目的不同對收斂代數有顯著的影響。

在表 4.8 中，菁英化的數目對於最後的收斂結果雖然有顯著的差異，不過並非菁英化的比例越高就代表有越好的效率，反而是 50% 的比例下得到的結果較好。原因可能在於部分單體雖然沒被挑到，但它們卻是在最佳解的附近，而那些被挑到的則可能是在區域最佳解的位置，因而造成這種結果。

c. 單體演化次數的不同對收斂代數有顯著的影響。

在表 4.8 中，單體演化的代數在 10 代和 15 代的收斂代數比 5 代來的少，原因可能在於本範例的函數適應值差異很大，因此一旦單體演化代數增加便會使的單體鑽的越深，越容易找出最佳解，另一方面，一旦單體演化代數到一定程度之後，如 15 代時，其效率便沒有太大差異的提升。

表 4.8 單因子平均代數比較表(算例三)

參數	資料型態		菁英化比例			單體演化代數		
	Binary	Real Value	0%	50%	100%	5	10	15
平均代數	84.089	51.78	72.783	25.233	105.71	112.97	48.916	41.9167

接下來檢視各因子之間是否存在交互作用：

a. 資料型態與菁英化數目之間的交互關係顯著。

在表 4.9 中，我們可得知實數型的資料型態與 100% 菁英化的組合所測試出來的結果是最好的，不過，並非菁英化的數量越多其效率就越好，因為二進位制資料型態與 50% 的組合結果就比 100% 來的好。

b. 資料型態與單體演化次數之間的交互關係較不顯著。

- c. 菁英化數目與單體演化次數之間的交互關係較不顯著。
 d. 資料型態、菁英化數目與單體演化次數之間的交互關係顯著。

在表 4. 11 中，平均收斂代數隨著菁英化的數目以及單體演化次數的增加而減少，除了在組合 3 有些許的異常外。

表 4.9 雙因子平均代數比較表(算例三)

平均代數 \ 因子	資料型態 vs 菁英化數目		資料型態 vs 單體演化次數	
	組合 1	B vs 0%	38.67	B vs 5
組合 2	B vs 50%	25.63	B vs 10	55.47
組合 3	B vs 100%	187.97	B vs 15	54.7
組合 4	R vs 0%	107.03	R vs 5	83.83
組合 5	R vs 50%	24.83	R vs 10	42.37
組合 6	R vs 100%	23.47	R vs 15	29.13

表 4.10 雙因子平均代數比較表(續)

平均代數 \ 因子	菁英化數目 vs 單體演化次數	
	組合 1	0% vs 5
組合 2	0% vs 10	57.6
組合 3	0% vs 15	44.25
組合 4	50% vs 5	27.05
組合 5	50% vs 10	23.6
組合 6	50% vs 15	25.05
組合 7	100% vs 5	195.15
組合 8	100% vs 0	65.55
組合 9	100% vs 15	56.45

表 4. 11 三因子平均代數比較表(算例三)

平均代數 \ 因子	資料型態 vs 菁英化數目 vs 單體演化次數	
組合 1	Binary vs 0% vs Nds=5	32.8
組合 2	Binary vs 0% vs Nds=10	33.6
組合 3	Binary vs 0% vs Nds=15	49.6
組合 4	Binary vs 50% vs Nds=5	28
組合 5	Binary vs 50% vs Nds=10	24.4
組合 6	Binary vs 50% vs Nds=15	24.5
組合 7	Binary vs All vs Nds=5	365.5
組合 8	Binary vs All vs Nds=10	108.4
組合 9	Binary vs All vs Nds=15	90
組合 10	Real-Val vs 0% vs Nds=5	200.6
組合 11	Real-Val vs 0% vs Nds=10	81.6
組合 12	Real-Val vs 0% vs Nds=15	38.9
組合 13	Real-Val vs 50% vs Nds=5	26.1
組合 14	Real-Val vs 50% vs Nds=10	22.8
組合 15	Real-Val vs 50% vs Nds=15	25.6
組合 16	Real-Val vs All vs Nds=5	24.8
組合 17	Real-Val vs All vs Nds=10	22.7
組合 18	Real-Val vs All vs Nds=15	22.9

2.收斂時間分析及探討

表 4. 12 為針對各參數對收斂時間的數據分析結果。在分析方面，我們取顯著水準 $\alpha=0.05$ 來檢定是否有顯著關係。

表 4. 12 收斂時間之變異數分析表(算例三)

變異來源	自由 度	均方	誤差均方	F 值	P-value
資料型態	1	3413.379	996.479	3.42544	0.06602
菁英化數目	2	7370.566	996.479	7.39661	0.000843
單體演化次數	2	6361.471	996.479	6.383949	0.002144
資料型態 vs 菁英化數目	2	16991.4	996.479	17.05144	1.9E-07
資料型態 vs 單體演化次數	2	593.7584	996.479	0.595856	0.552294
菁英化數目 vs 單體演化次數	4	2069.367	996.479	2.076679	0.086222
資料型態 vs 菁英化數目 vs 單體演化次數	4	5408.572	996.479	5.427683	0.000398
誤差值	162				

首先，先就單因子來探討對收斂時間的影響：

a. 資料型態的不同對收斂時間沒有顯著的影響。

在表 4. 13 中，不同的資料型態對收斂時間雖然有少許差異，不過在統計的信賴區間內可視為是相近的。

b. 菁英化數目的不同對收斂時間有顯著的影響。

在表 4. 13 中，菁英化的數目對於最後的收斂結果雖然有顯著的差異，不過並非菁英化的比例越高就代表有越好的效率，反而是 50% 的比例下得到的結果較好。原因可能在於部分單體雖然沒被挑到，但它們卻是在最佳解的附近，而那些被挑到的則可能是在區域最佳解的位置，因而造成這種結果。

c. 單體演化次數的不同對收斂時間有顯著的影響。

在表 4. 13 中，單體演化的代數在 10 代和 15 代的收斂時間比 5 體演化代數增加便會使的單體鑽的越深，越容易找出最佳解，進而縮短搜尋時間。另一方面，一旦單體演化代數到一定程度之後，如 15 代時，其效率便沒有太大差異的提升。

表 4. 13 單因子平均收斂時間比較表(算例三)

參數	資料型態		菁英化比例			單體演化代數		
	Binary	Real Value	0%	50%	100%	5	10	15
平均收斂時間	17.077	8.367	14.258	0.95	22.957	24.58	7.49	6.08

接下來檢視各因子之間是否存在交互作用：

a. 資料型態與菁英化數目之間的交互關係顯著。

在表 4. 14 中，值得注意的是在二進位制的資料型態中菁英化的比例越高不代表收斂時間越短，反而在 50% 的條件下其結果遠甚於其他兩組。

b. 資料型態與單體演化次數之間的交互關係較不顯著。

c. 菁英化數目與單體演化次數之間的交互關係較不顯著。

d. 資料型態、菁英化數目與單體演化次數之間的交互關係顯著。

在表 4. 16 中，值得注意的是菁英化數目增加或單體演化次數增加，並不會造成收斂時間的減少。組合 14 為最佳的結果。

表 4. 14 雙因子平均時間比較表(算例三)

平均時間 \ 因子	資料型態 vs 菁英化數目		資料型態 vs 單體演化次數	
	組合 1	B vs 0%	3.95	B vs 5
組合 2	B vs 50%	1.58	B vs 10	8.94
組合 3	B vs 100%	45.68	B vs 15	10.01
組合 4	R vs 0%	24.56	R vs 5	16.89
組合 5	R vs 50%	0.31	R vs 10	6.05
組合 6	R vs 100%	0.22	R vs 15	2.16

表 4.15 雙因子平均時間比較表(續)

平均時間	因子	菁英化數目 vs 單體演化次數	
		組合 1	0% vs 5
組合 2	0% vs 10	10.22	
組合 3	0% vs 15	6.28	
組合 4	50% vs 5	1.06	
組合 5	50% vs 10	0.74	
組合 6	50% vs 15	1.04	
組合 7	100% vs 5	46.42	
組合 8	100% vs 0	11.52	
組合 9	100% vs 15	10.93	

表 4.16 三因子平均時間比較表(算例三)

平均時間	因子	資料型態 vs 菁英化數目 vs 單體演化次數	
		組合 1	Binary vs 0% vs Nds=5
組合 2	Binary vs 0% vs Nds=10	2.6728	
組合 3	Binary vs 0% vs Nds=15	6.8949	
組合 4	Binary vs 50% vs Nds=5	1.8967	
組合 5	Binary vs 50% vs Nds=10	1.3018	
組合 6	Binary vs 50% vs Nds=15	1.5623	
組合 7	Binary vs All vs Nds=5	92.6354	
組合 8	Binary vs All vs Nds=10	22.8511	
組合 9	Binary vs All vs Nds=15	21.5779	
組合 10	Real-Val vs 0% vs Nds=5	50.2387	
組合 11	Real-Val vs 0% vs Nds=10	17.7716	
組合 12	Real-Val vs 0% vs Nds=15	5.6766	
組合 13	Real-Val vs 50% vs Nds=5	0.2323	
組合 14	Real-Val vs 50% vs Nds=10	0.1852	
組合 15	Real-Val vs 50% vs Nds=15	0.5248	
組合 16	Real-Val vs All vs Nds=5	0.2022	
組合 17	Real-Val vs All vs Nds=10	0.1972	

組合 18	Real-Val vs All vs Nds=15	0.2784
-------	---------------------------	--------

4.1.4 算例(四) : De Jong function F2

(1)目標函數

目標函數如(4-4)所示，此函數目的在求此函數的最小值，圖 4. 24 為此目標函數之圖形。

$$\text{Min. } F(X) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2 \quad (4-4)$$

$$\text{where } X_i \in [-2.048, 2.048]$$

此函數之理想(Ideal)最佳值為 0。

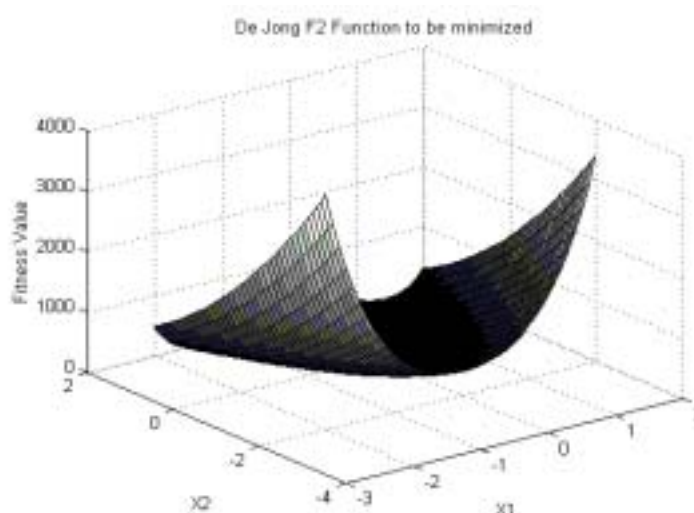


圖 4. 24 De Jong F2 之圖形

(2)測試環境與變數設定

此研究之相關變數如下：

- a. 族群大小：本範例母體大小為 42。
- b. 單一變數之基因數：單一變數之染色體長度為 19，精確度為 10^5 。
- c. 複製法則：輪盤法(Roulette)。
- d. 交配法則：以二進位制為編碼方式的交配方式為雙點交配。交配機率 $P_c=0.4$ ；以十進位為編碼方式的交配方式為算術交配法 (Arithmetic Crossover)。
- e. 突變法則：以二進位制為編碼方式的突變為參考 2.1 節。突變機

率 $P_m = 0.03$; 以十進位為編碼方式的突變方式為均勻突變法 (Uniform Mutation)。

- f. 演化代數：在本範例中，我們取最大代數為 1000。
- g. 收斂條件：在本範例中，當與最佳值的誤差小於 0.001 時且連續 20 代皆如此，則視為收斂。
- h. 測試次數：30 次。
- i. 參數設定：在本範例中，菁英化單體數目為兩個；單體演化次數為 5 次、10 次和 15 次。

(3)實驗結果與比較

圖 4. 25 至圖 4. 31 以及表 4. 17 為本研究實驗的結果。根據變異數分析(ANOVA)的結果(請參考附錄一表 A1. 10 及表 A1. 11), 方法的不同對收斂代數有顯著的影響且方法的不同對收斂時間有顯著的影響。

在表 4. 17 中，在單體演化次數為 5 次時的實數型 HMSGGA-CS 其『收斂百分比』並不如 Y-L 的來的高外，其餘的組合皆比 GA 或 Y-L 的好。在『平均收斂代數』方面，除了在 Binary 且單體演化次數為 5(Nds=5)的 HMSGGA-CS 中比 Y-L 的差外，其餘的組合皆比 Y-L 及 GA 的方法好，『平均收斂時間』也比較短。

表 4. 17 測試結果表(算例四)

Method			Convergence Percentage	Average* Generation	Standard Deviation (Generation)	Mean* Time(sec.) (Convergence)	Mean Value	Accuracy
HMSGGA-CS	Binary	Nds=5	96.67%	295.5385	303.1154	34.7382	0.0005545	3.9209E-04
		Nds=10	93.33%	44.4286	39.7878	6.0362	0.0004656	3.2922E-04
		Nds=15	100%	25.8333	5.6513	1.58593	0.0003016	2.1326E-04
	Real Value	Nds=5	40%	134.3333	157.9996	9.0575	0.0583	4.1224E-02
		Nds=10	83.33%	53.64	91.5336	2.6086	0.0237	1.6758E-02
		Nds=15	83.33%	23.52	2.0025	0.3922	0.0255	1.8031E-02
Y-L (Binary)			90%	225.8889	229.5406	26.0029	0.0011087	7.8396E-04
Y-L (Real Value)			56.67%	188.0588	178.4042	7.2492	0.0154738	1.0941E-02
GA (Binary)			46.67%	318	281.88	27.53	0.07964	5.632E-02
GA (Real Value)			6.67%	487.5	174.66	18.006	0.0479	3.387E-02

*代表只計算有收斂的數值

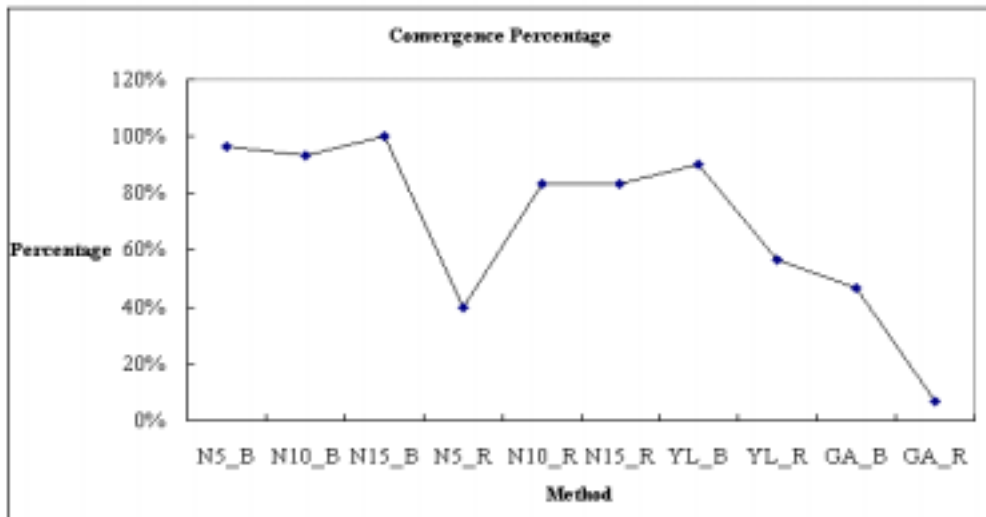


圖 4.25 收斂率比較圖(算例四)

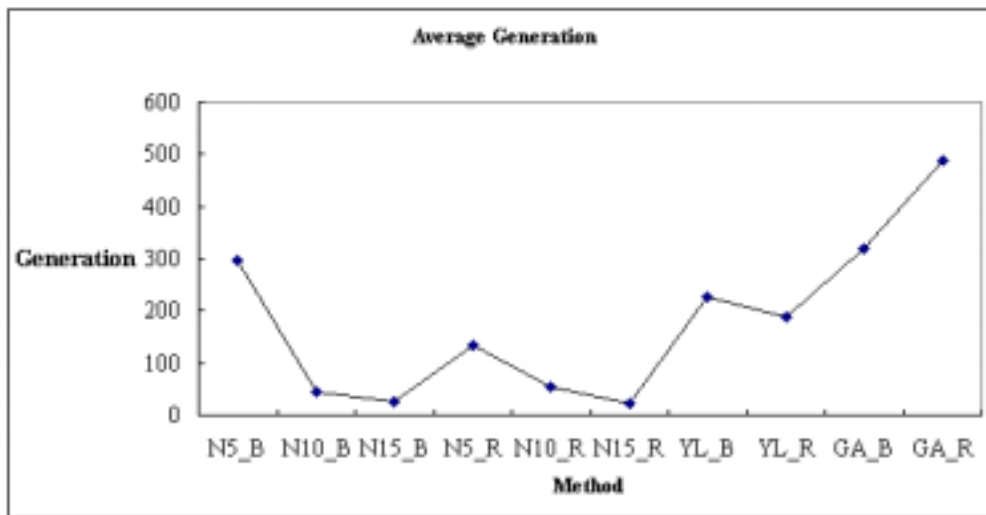


圖 4.26 平均收斂代數比較圖(算例四)

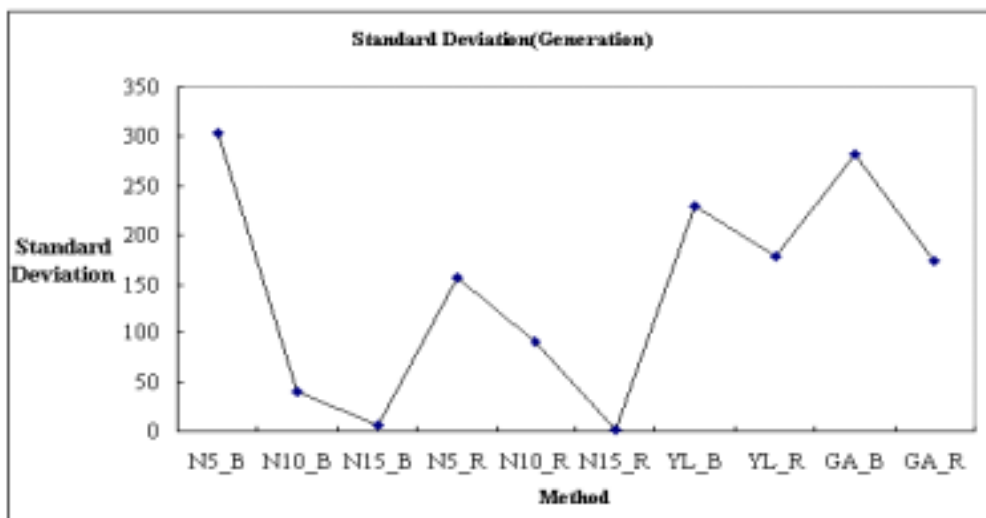


圖 4.27 平均收斂代數之標準差比較圖(算例四)

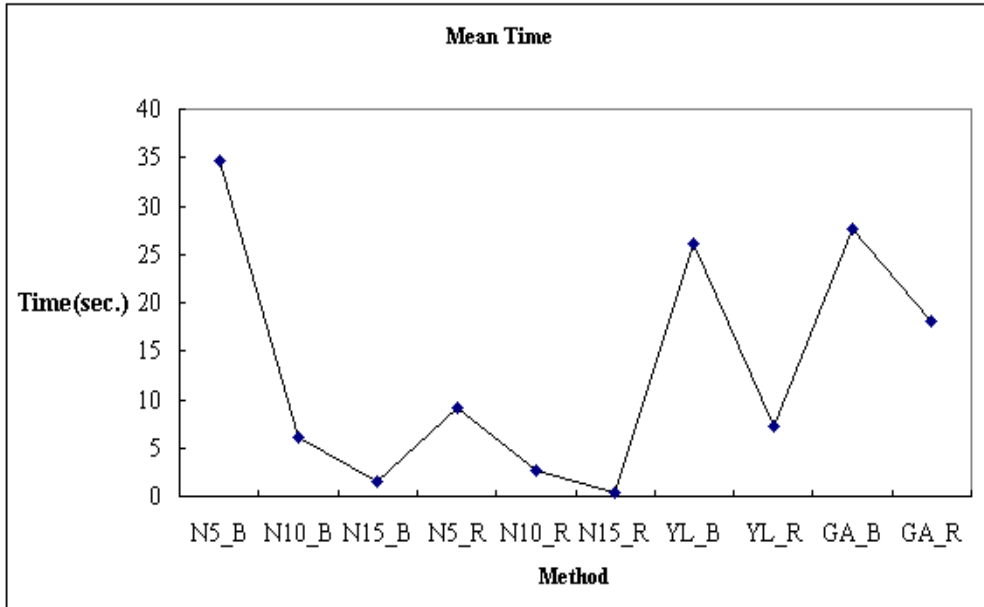


圖 4.28 達收斂時之平均運算時間比較圖(算例四)

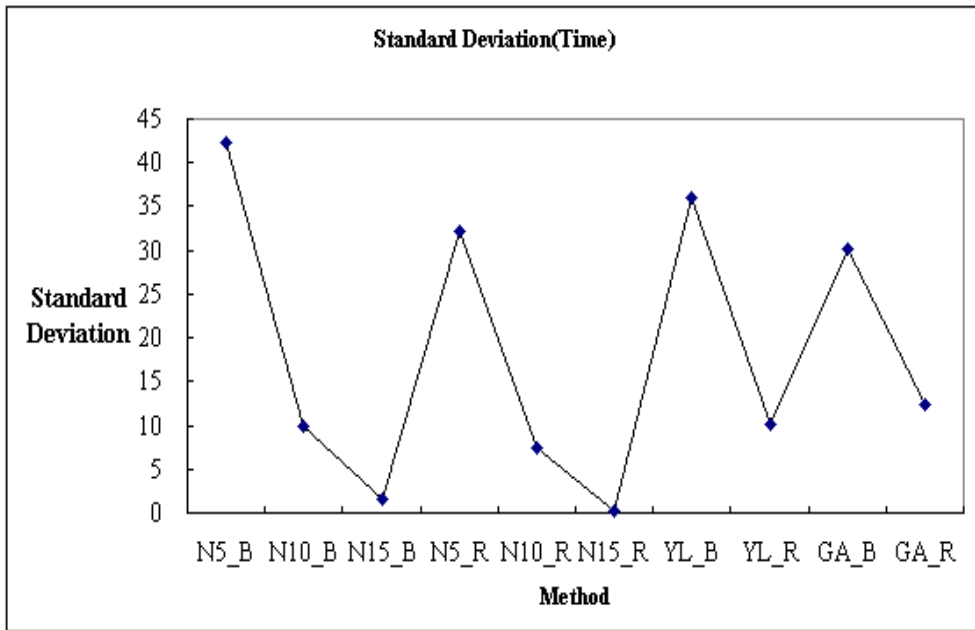


圖 4.29 平均運算時間之標準差比較圖(算例四)

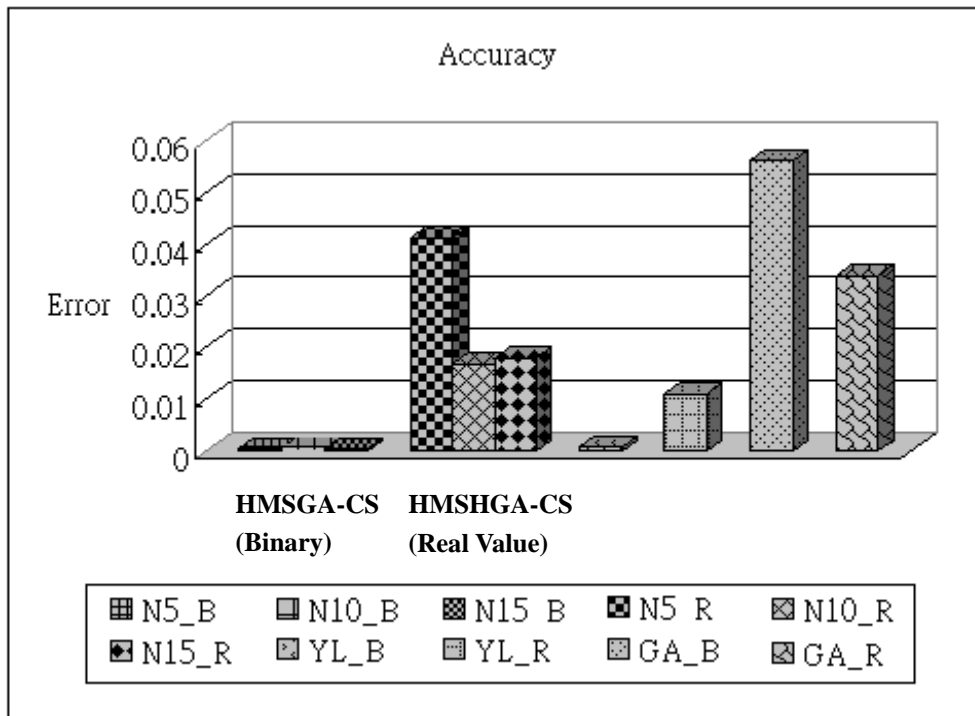


圖 4.30 與理想值之誤差比較圖(算例四)

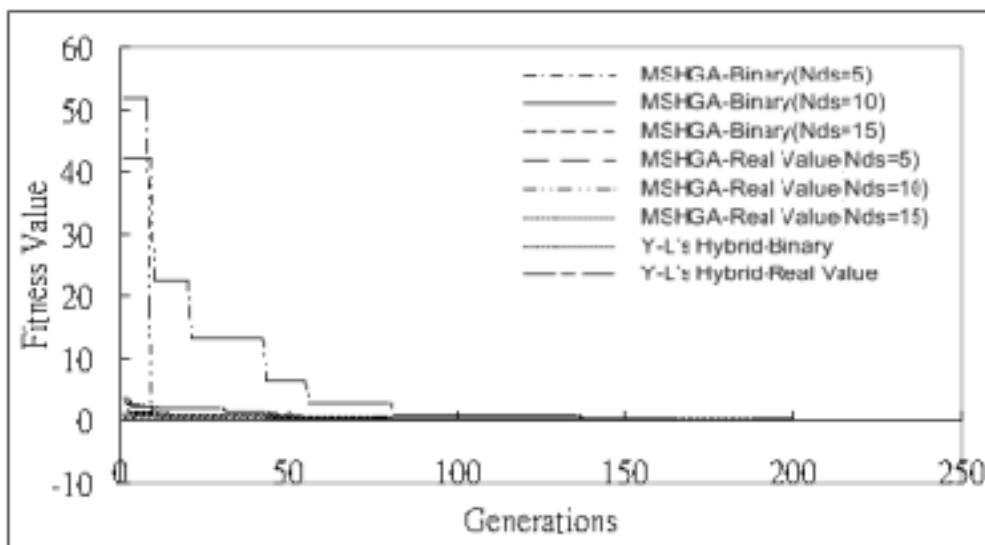


圖 4.31 代數收斂表示圖(算例四)

4.1.5 算例(五) : Schaffer F6 function

(1)目標函數

目標函數如(4-5)所示，此函數目的在求此函數的最小值，圖 4.32 為此目標函數之圖形。

$$\text{Min. } F(X) = 0.5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1.0 + 0.001(x_1^2 + x_2^2)]^2} \quad (4-5)$$

where $X_i \in [-100, 100]$

此函數之理想(Ideal)最佳值為 0。

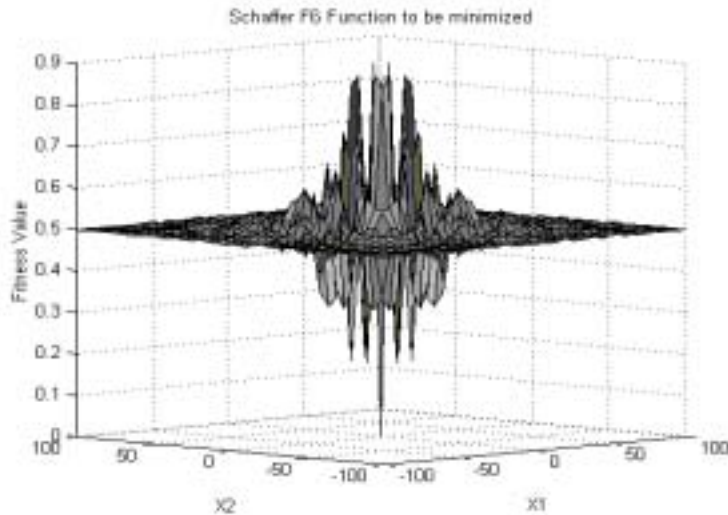


圖 4. 32 Schaffer F6 之圖形

(2)測試環境與變數設定

此研究之相關變數如下：

- a. 族群大小：本範例母體大小為 42。
- b. 單一變數之基因數：單一變數之染色體長度為 25，精確度為 10^5 。
- c. 複製法則：輪盤法(Roulette)。
- d. 交配法則：以二進位制為編碼方式的交配方式為雙點交配。交配機率 $P_c = 0.4$ ；以十進位為編碼方式的交配方式為算術交配法 (Arithmetic Crossover)。
- e. 突變法則：以二進位制為編碼方式的突變為參考 2.1 節。突變機率 $P_m = 0.03$ ；以十進位為編碼方式的突變方式為均勻突變法 (Uniform Mutation)。
- f. 演化代數：在本範例中，我們取最大代數為 1000。
- g. 收斂條件：在本範例中，當與最佳值的誤差小於 0.00001 時且連續 20 代皆如此，則視為收斂。
- h. 測試次數：30 次。

- i. 參數設定：在本範例中，菁英化單體數目為兩個；單體演化次數為 5 次、10 次和 15 次。

(3)實驗結果與比較

圖 4. 33 至圖 4. 39 以及表 4. 18 為我們本研究的結果。根據變異數分析(ANOVA)的結果(請參考附錄一表 A1. 12 及表 A1. 14), 方法的不同對收斂代數有顯著的影響且方法的不同對收斂時間有顯著的影響。

另外，HMSGGA-CS 在不同的組合上經變異數分析(ANOVA)的結果證實，對『平均收斂代數』不會有顯著的影響(p-value=0.48，請參考附錄一表 A1. 13)。

在 Y-L(Real Value)的實驗中的『收斂百分比』為 0，也就是在我們 30 次的測試中都未收斂過，所以我們便將最大代數的設限取消，經實驗 5000 代後仍然無法達到設定的誤差值內，其值為 0.009715，時間為 883.831 秒。

表 4. 18 測試結果表(算例五)

Method			Convergence Percentage	Average* Generation	Standard Deviation (Generation)	Mean* Time(sec.) (Convergence)	Mean Value	Accuracy
HMSGGA-CS	Binary	Nds=5	56.67%	33.5294	13.4959	2.9456	0.0038	2.694E-03
		Nds=10	50.00%	27.5333	4.7036	1.8640	0.0049	3.436E-03
		Nds=15	60.00%	25.6111	2.9533	1.688	0.0036	2.580E-03
	Real Value	Nds=5	56.67%	32.6429	13.4084	0.4743	0.0070	4.930E-03
		Nds=10	50.00%	33.7273	24.1416	1.0808	0.0065	4.583E-03
		Nds=15	56.67%	26.2353	6.8422	0.7152	0.0043	3.071E-03
Y-L (Binary)			20%	117.5	95.459	38.6955	0.0020	1.477E-03
Y-L (Real Value)			0%	1000	0	261.085	0.0152	1.080E-02
GA (Binary)			30%	101.56	40.908	5.73	0.0068	4.81E-03
GA (Real Value)			0%	1000	0	69.591	0.02248	1.589E-02

*代表只計算有收斂的數值。當『收斂百分比』為 0 時，『平均收斂時間』取演算 1000 代的平均時間。

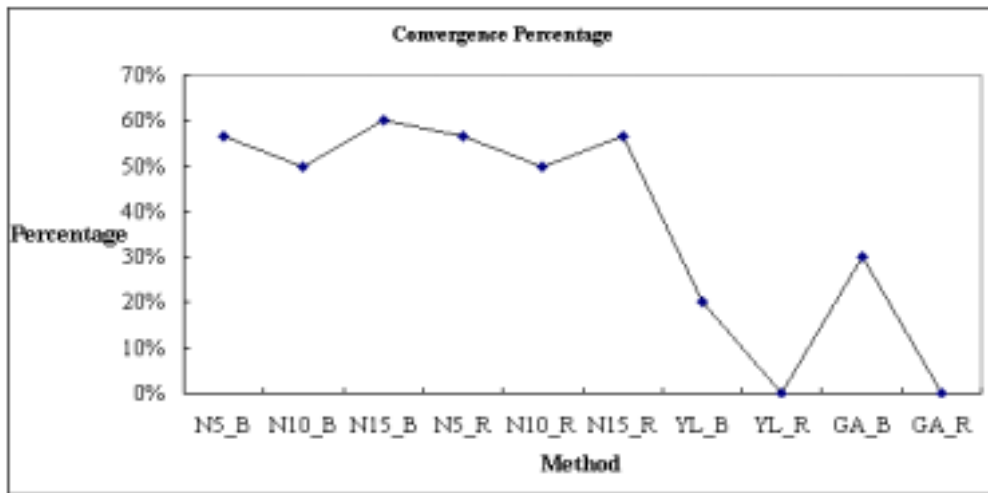


圖 4.33 收斂率比較圖(算例五)

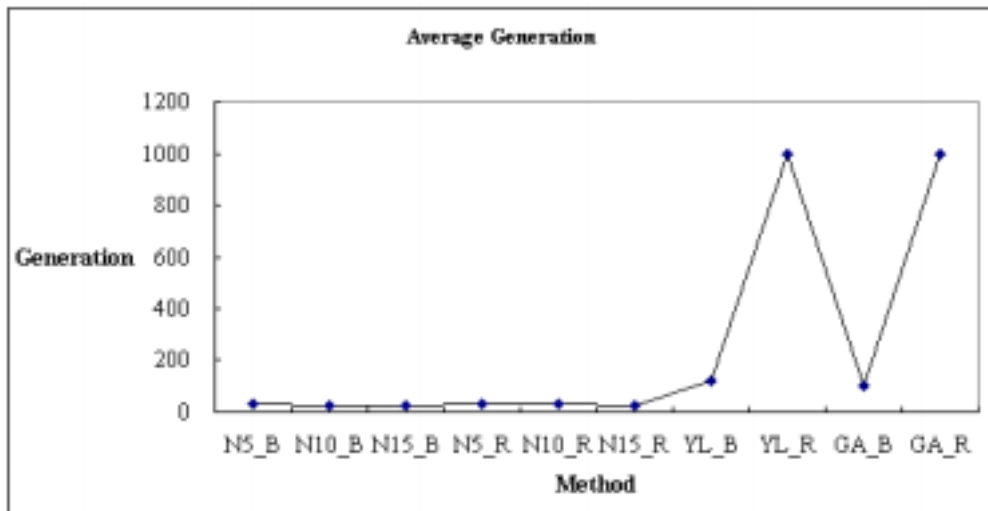


圖 4.34 平均收斂代數比較圖(算例五)

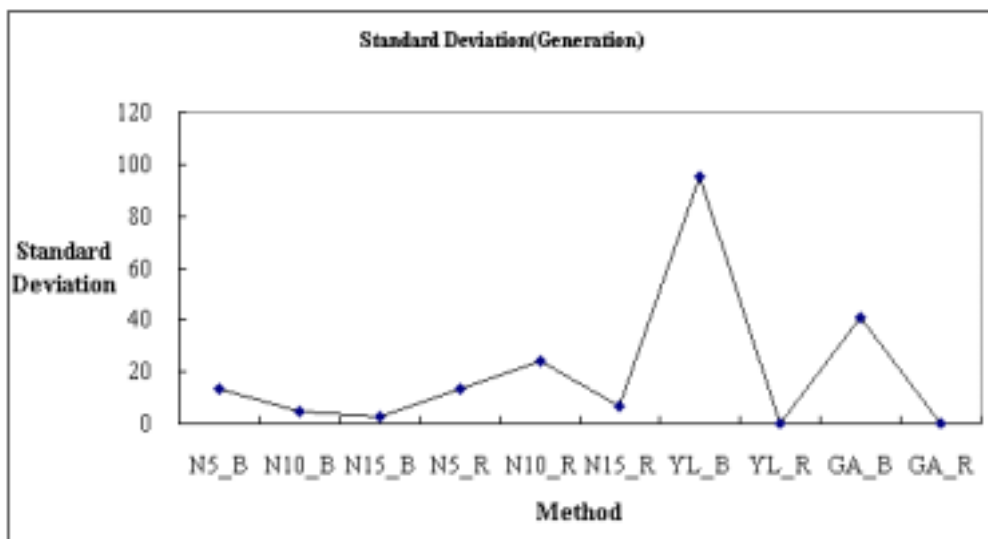


圖 4.35 平均收斂代數之標準差比較圖(算例五)

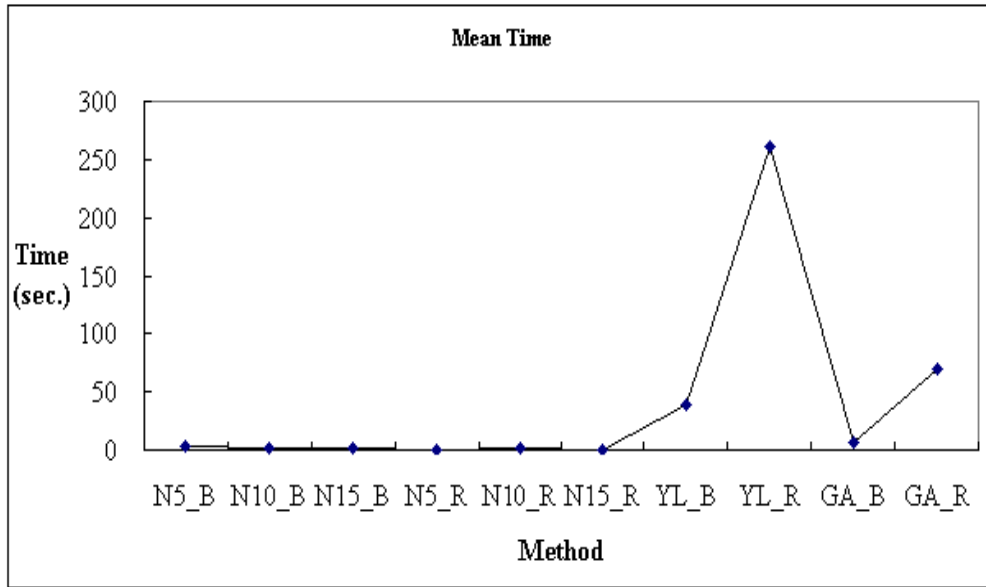


圖 4.36 達收斂時之平均運算時間比較圖(算例五)

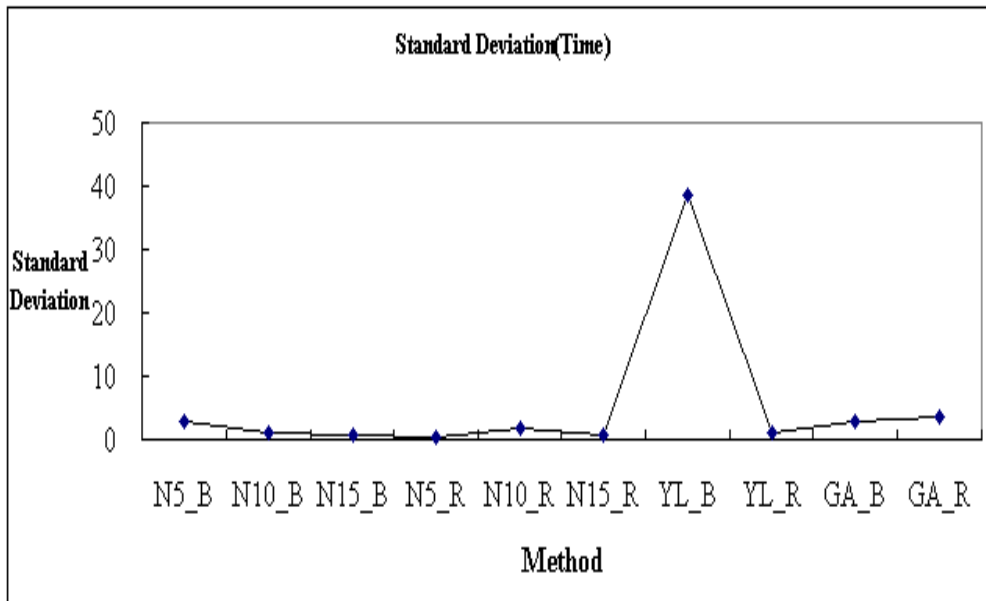


圖 4.37 平均運算時間之標準差比較圖(算例五)

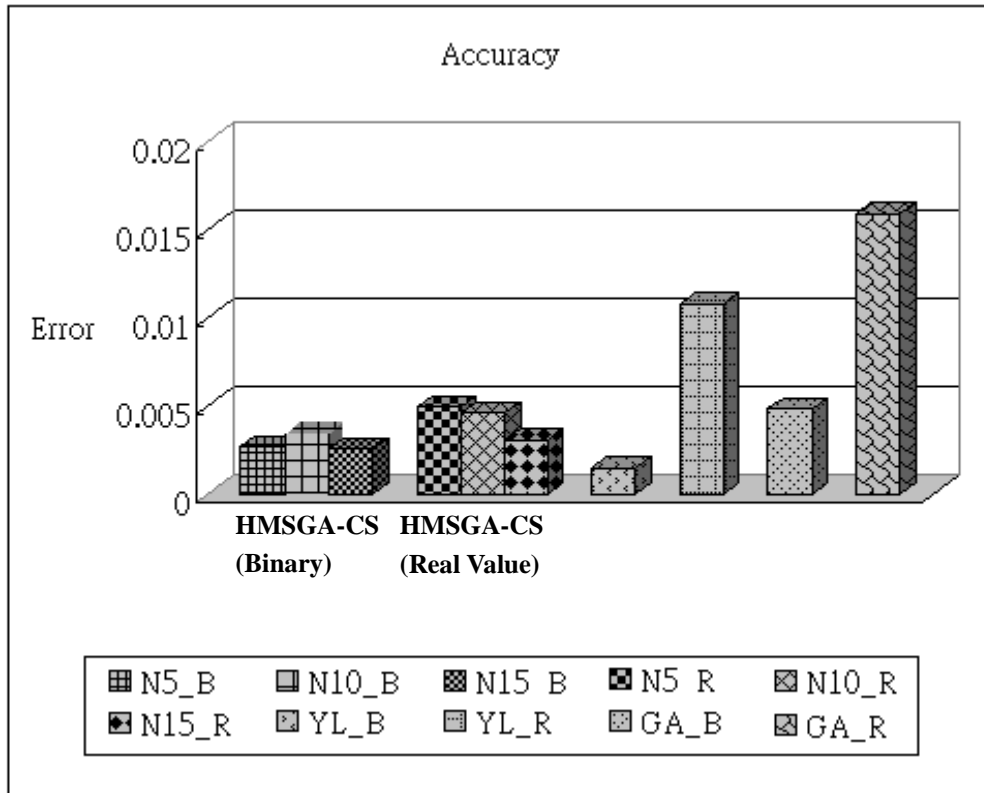


圖 4. 38 與理想值之誤差比較圖(算例五)

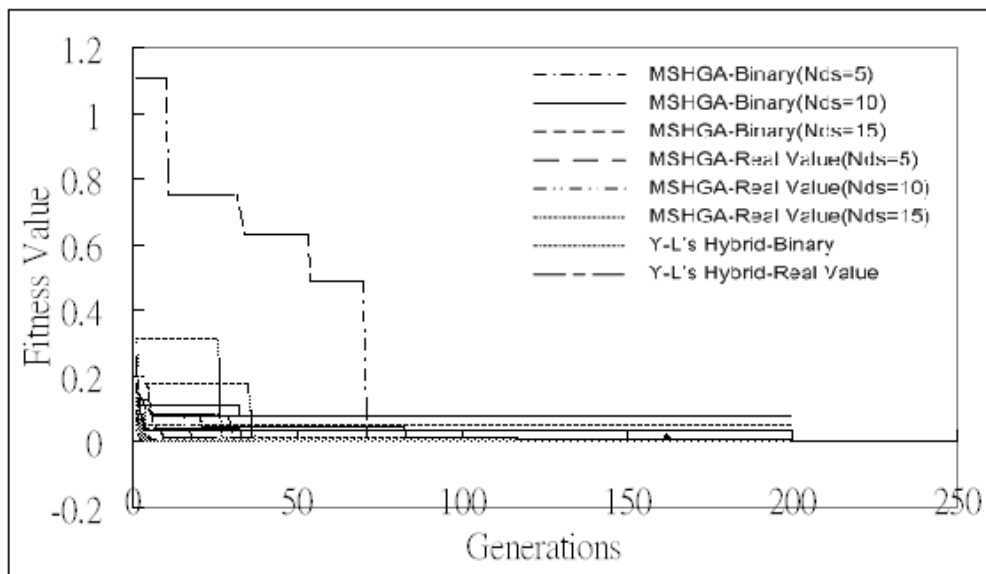


圖 4. 39 代數收斂圖(算例五)

4.1.6 算例(六) : Schaffer F7 function

(1)目標函數

目標函數如(4-6)所示，此函數目的在求此函數的最小值。圖 4. 40

為此目標函數之圖形。

$$\text{Min. } F(X) = (x_1^2 + x_2^2)^{0.25} [\sin^2(50(x_1^2 + x_2^2)^{0.1}) + 1.0] \quad (4-6)$$

where $X_i \in [-100, 100]$

此函數之理想(Ideal)最佳值為 0。

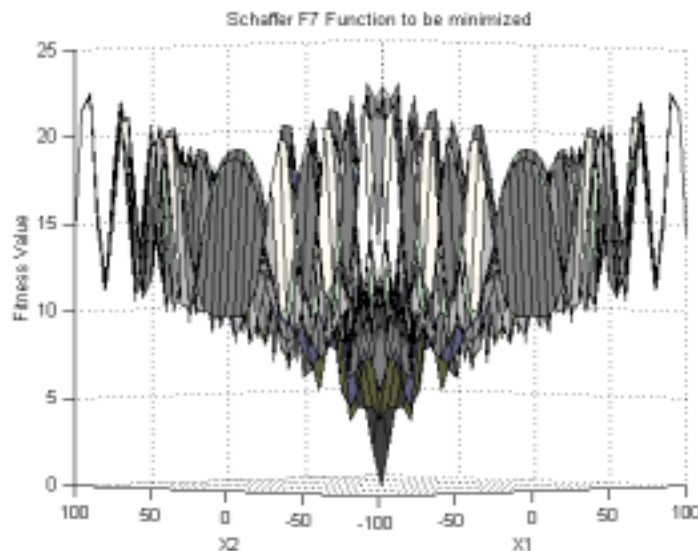


圖 4. 40 Schaffer F7 之圖形

(2)測試環境與變數設定

此研究之相關變數如下：

- 族群大小：本範例母體大小為 42。
- 單一變數之基因數：單一變數之染色體長度為 25，精確度為 10^5 。
- 複製法則：輪盤法(Roulette)。
- 交配法則：以二進位制為編碼方式的交配方式為雙點交配。交配機率 $P_c=0.4$ ；以十進位為編碼方式的交配方式為算術交配法 (Arithmetic Crossover)。
- 突變法則：以二進位制為編碼方式的突變為參考 2.1 節。突變機率 $P_m=0.03$ ；以十進位為編碼方式的突變方式為均勻突變法 (Uniform Mutation)。
- 演化代數：在本範例中，我們取最大代數為 1000。
- 收斂條件：在本範例中，當與最佳值的誤差小於 0.001 時且連續

20 代皆如此，則視為收斂。

h. 測試次數：30 次。

i. 參數設定：在本範例中，菁英化單體數目為兩個；單體演化次數為 5 次、10 次和 15 次。

(3) 實驗結果與比較

圖 4. 88 至圖 4. 94 以及表 4. 19 為本研究實驗的結果。根據變異數分析(ANOVA)的結果(請參考附錄一表 A1. 15 及表 A1. 17), 方法的不同對收斂代數有顯著的影響且方法的不同對收斂時間有顯著的影響。

在 Y-L(Real Value)的實驗中的『收斂百分比』為 0，也就是在我們 30 次的測試中都未收斂過，所以我們便將最大代數的設限取消，經實驗 5000 代後仍然無法達到設定的誤差值內，其值為 0.2009316，時間為 597.489 秒，而由於其未收斂，所以其收斂代數之標準差便為 0，此一數據不得與其他組合探討。

另一方面，經變異數分析(ANOVA)後，得知 HMSGGA-CS 在不同的資料型態上即使搭配不同的單體演化代數的組合，對『平均收斂代數』(p-value=0.2589>，請參考附錄一表 A1. 16)和『平均收斂時間』(p-value=0.2619>，請參考附錄一表 A1. 18)並無顯著的影響。

表 4. 19 測試結果表(算例六)

Method		Convergence Percentage	Average* Generation	Standard Deviation (Generation)	Mean* Time(sec.) (Convergence)	Mean Value	Accuracy	
HMSGGA-CS	Binary	Nds=5	100.00%	35	3.605551	3.27267	8.02E-04	5.676E-04
		Nds=10	100.00%	27.3333	2.663688	1.7652	8.0E-04	5.691E-04
		Nds=15	100.00%	25.5333	1.28388	1.619	5.28E-04	3.735E-04
	Real Value	Nds=5	93.33%	32.3571	2.273835	0.51078	1.67E-02	1.182E-02
		Nds=10	100.00%	27.6	1.80475	0.57946	6.8E-04	4.817E-04
		Nds=15	100.00%	26.3333	1.588650	0.675	5.79E-04	4.091E-04
Y-L (Binary)		66.67%	179.8	101.856	16.035	2.99E-03	2.120E-03	
Y-L (Real Value)		0.00%	1000	0	49.96853	5.401E-01	3.819E-01	
GA (Binary)		100.00%	285.167	98.966	19.108	8.19E-04	5.79E-04	
GA (Real Value)		0%	1000	0	40.161	8.56E-01	6.058E-01	

*代表只計算有收斂的數值。當『收斂百分比』為 0 時，『平均收斂時間』取演算 1000 代的平均時間。

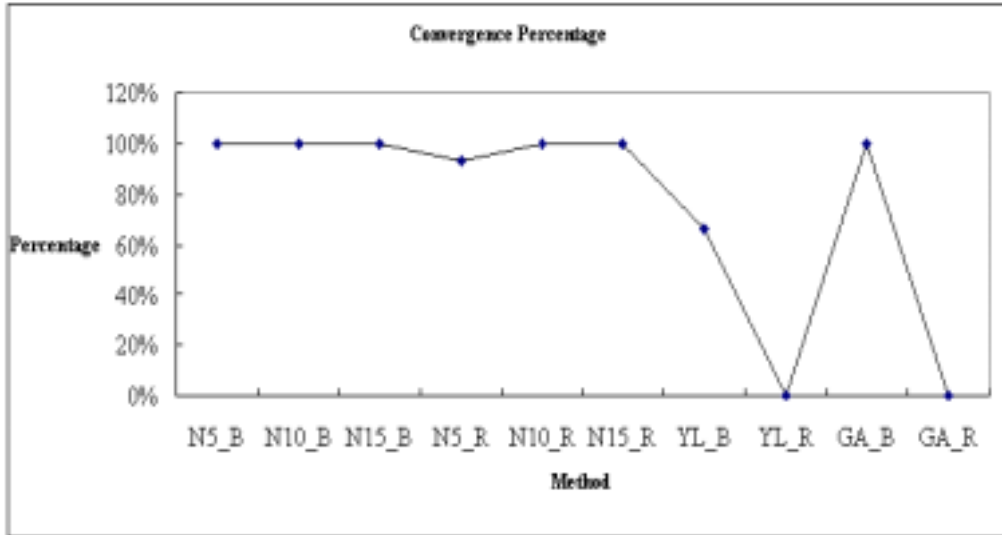


圖 4. 41 收斂率比較圖(算例六)

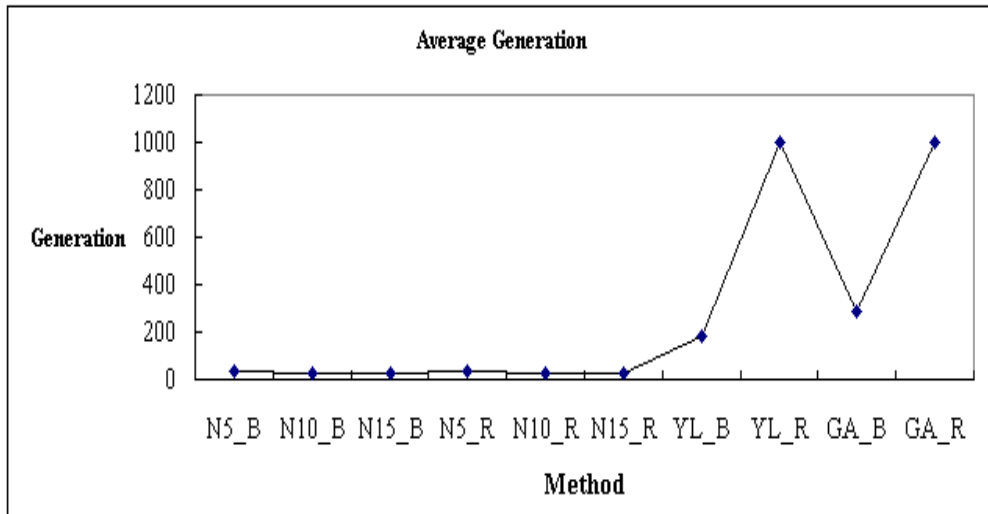


圖 4. 42 平均收斂代數比較圖(算例六)

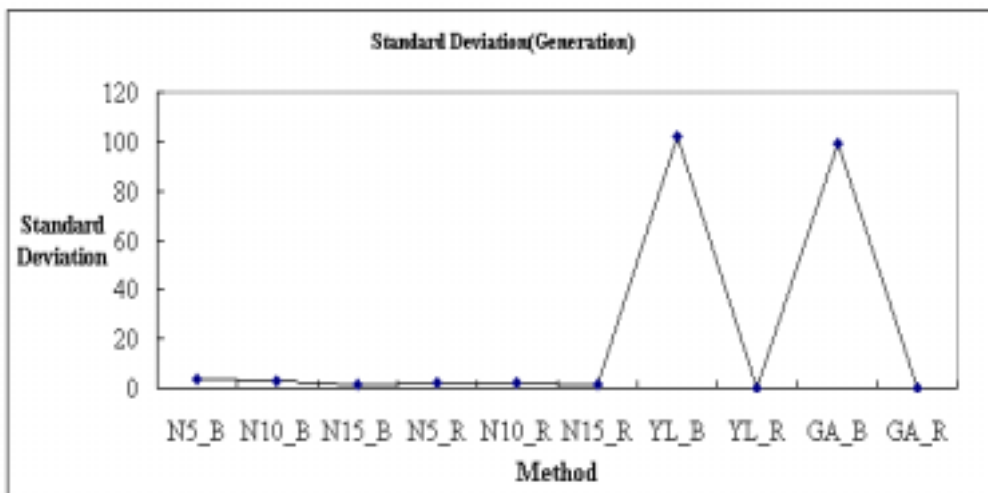


圖 4. 43 平均收斂代數之標準差比較圖(算例六)

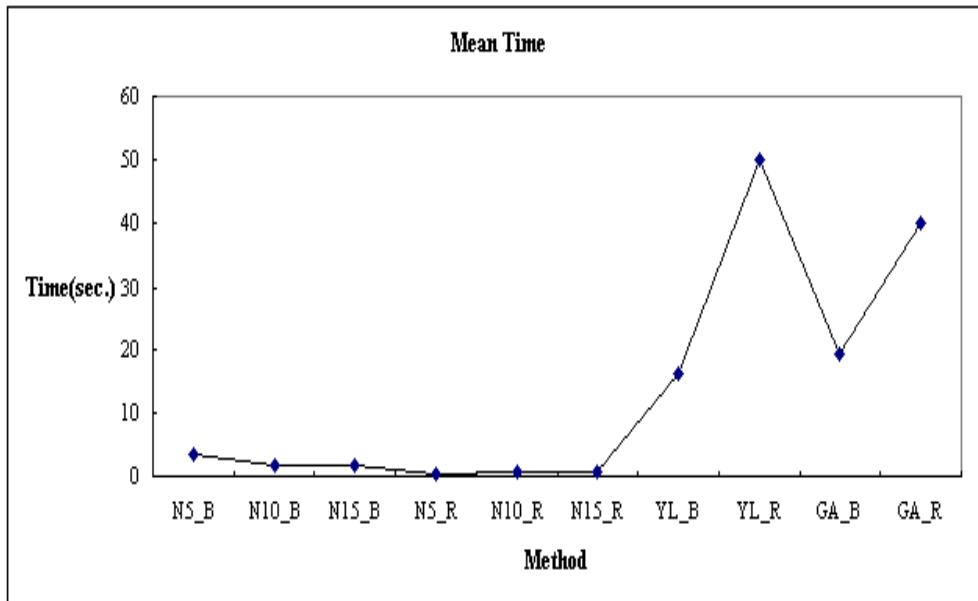


圖 4.44 達收斂時之平均運算時間比較圖(算例六)

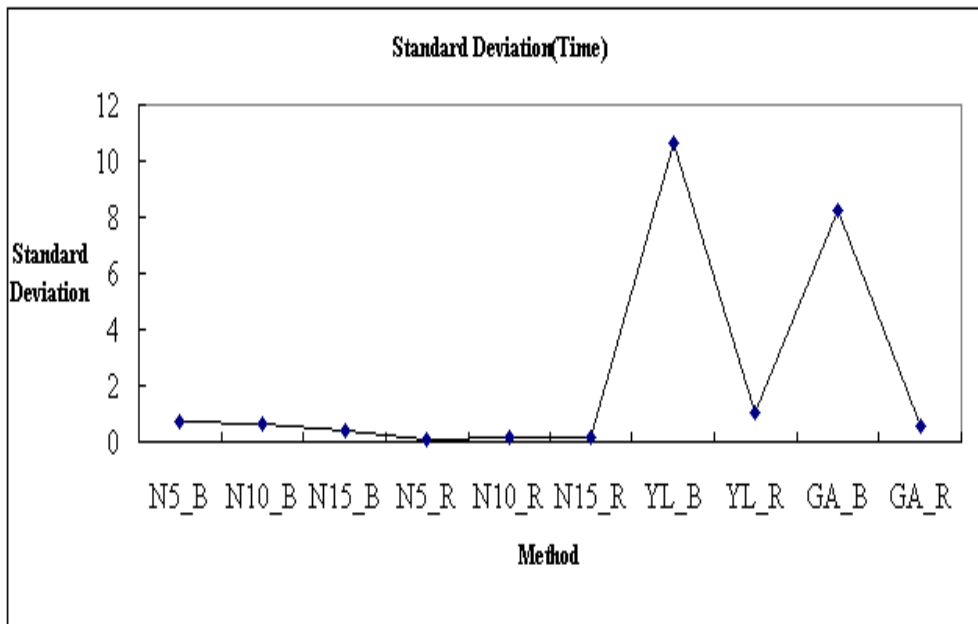


圖 4.45 平均運算時間之標準差比較圖(算例六)

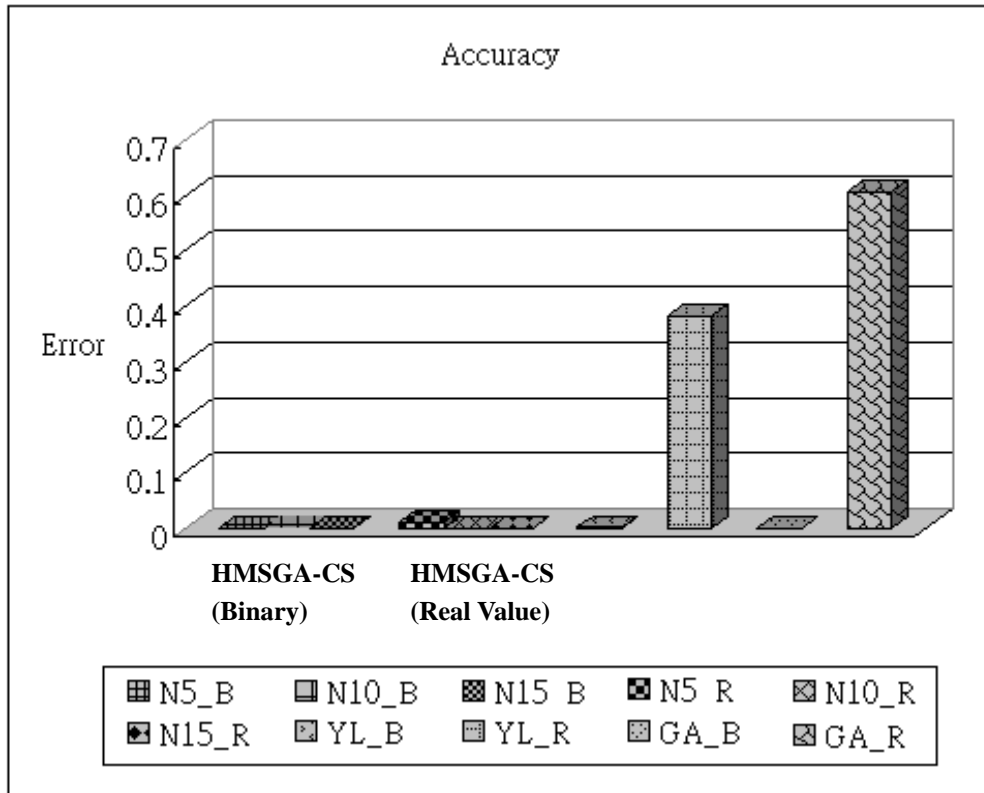


圖 4.46 與理想值之誤差比較圖(算例六)

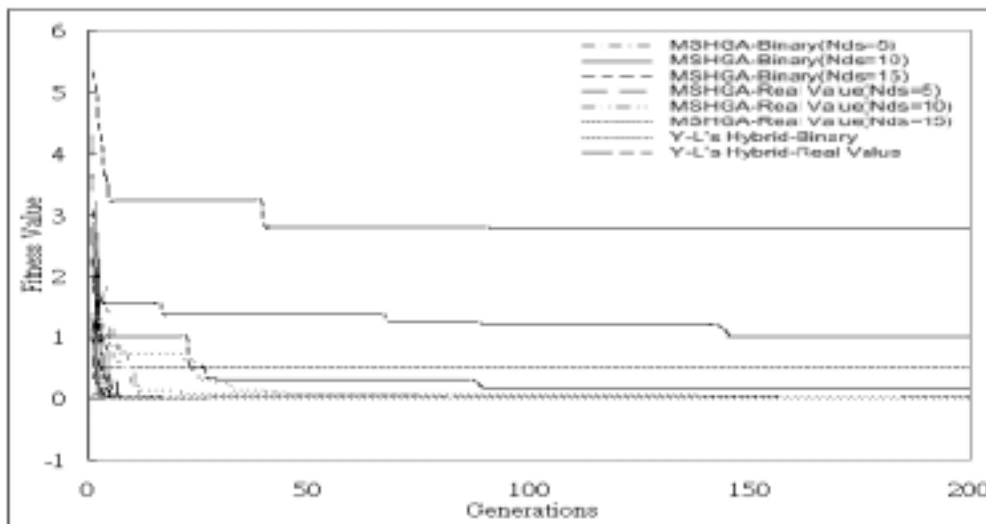


圖 4.47 代數收斂表示圖(算例六)

4.1.7 算例(七) : Shubert function

(1) 目標函數

目標函數如(4-7)所示，此函數目的在求此函數的最小值。圖 4.48

為此目標函數之圖形。

$$\text{Min. } F(X) = \sum_{i=1}^5 i \cos[(i+1)x_1 + i] \cdot \sum_{i=1}^5 i \cos[(i+1)x_2 + i] \quad (4-7)$$

where $X_i \in [-10,10]$

此函數之理想(Ideal)最佳值為-186.727。此函數有 760 個區域最佳解，其中有 18 個為全域最佳解。

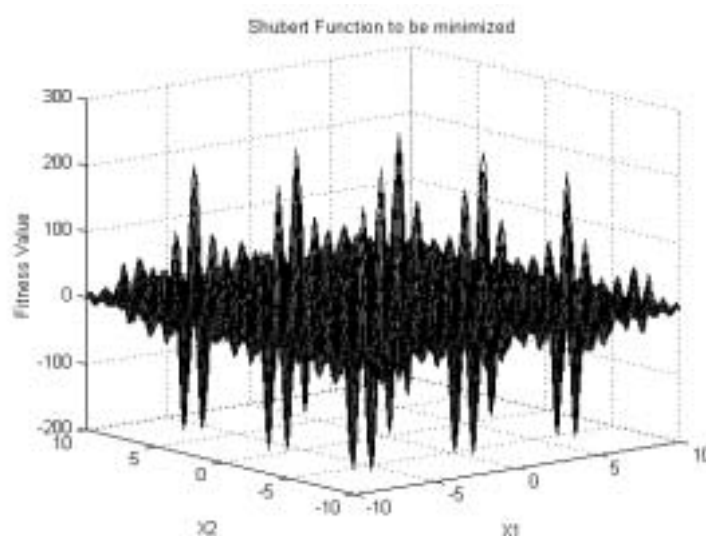


圖 4.48 Shubert 之圖形

(2)測試環境與變數設定

此研究之相關變數如下：

- 族群大小：本範例母體大小為 42。
- 單一變數之基因數：單一變數之染色體長度為 21，精確度為 10^5 。
- 複製法則：輪盤法(Roulette)。
- 交配法則：以二進位制為編碼方式的交配方式為雙點交配。交配機率 $P_c=0.4$ ；以十進位為編碼方式的交配方式為算術交配法 (Arithmetic Crossover)。
- 突變法則：以二進位制為編碼方式的突變為參考 2.1 節。突變機率 $P_m=0.03$ ；以十進位為編碼方式的突變方式為均勻突變法 (Uniform Mutation)。
- 演化代數：在本範例中，我們取最大代數為 1000。

- g. 收斂條件：在本範例中，當與最佳值的誤差小於 0.001 時且連續 20 代皆如此，則視為收斂。
- h. 測試次數：30 次。
- i. 參數設定：在本範例中，菁英化單體數目為兩個；單體演化次數為 5 次、10 次和 15 次。

(3) 實驗結果與比較

圖 4. 49 至圖 4. 55 以及表 4. 20 為本研究實驗的結果。根據變異數分析(ANOVA)的結果(請參考附錄一表 A1. 19 及表 A1. 21), 方法的不同對收斂代數有顯著的影響且方法的不同對收斂時間有顯著的影響。

在表 4. 20 中，以『收斂百分比』來說，HMSGGA-CS 與 Y-L 混合法皆在二進位制(Binary)的資料型態裡，有較佳的比例。對於『平均收斂代數』來說，若將二進位(Binary)資料型態的 Nds=10 和 Nds=15 與 Y-L(Binary)的三種組合進行變異數分析後，得到的結果並無顯著的差異($p\text{-value}=0.121 > \alpha$ ，請參考附錄一表 A1. 20)。

表 4. 20 測試結果表(算例七)

Method			Convergence Percentage	Average* Generation	Standard Deviation (Generation)	Mean* Time(sec.) (Convergence)	Mean Value	Accuracy
HMSGGA-CS	Binary	Nds=5	80%	219	231.1438	55.61306	-186.72	6.835E-03
		Nds=10	100%	37.7	24.4348	4.39185	-186.72	2.295E-03
		Nds=15	100%	31.9	52.4353	9.5062	-186.73	2.408E-03
	Real Value	Nds=5	24%	120.167	192.284	8.8425	-186.25	3.371E-01
		Nds=10	60%	29.25	45.7108	2.82	-186.58	1.008E-01
		Nds=15	80%	31.9375	16.575	1.706	-186.73	9.087E-03
Y-L (Binary)			100%	37.87	9.619	1.8844	-186.73	3.029E-03
Y-L (Real Value)			83.33%	205.72	132.9155	6.3411	-186.72	6.297E-02
GA (Binary)			53.33%	468.25	283.772	38.998	-186.50	1.602E-01
GA (Real Value)			0%	1000	0	41.622	-185.65	7.627E-01

*代表只計算有收斂的數值。當『收斂百分比』為 0 時，『平均收斂時間』取演算 1000 代的平均時間。

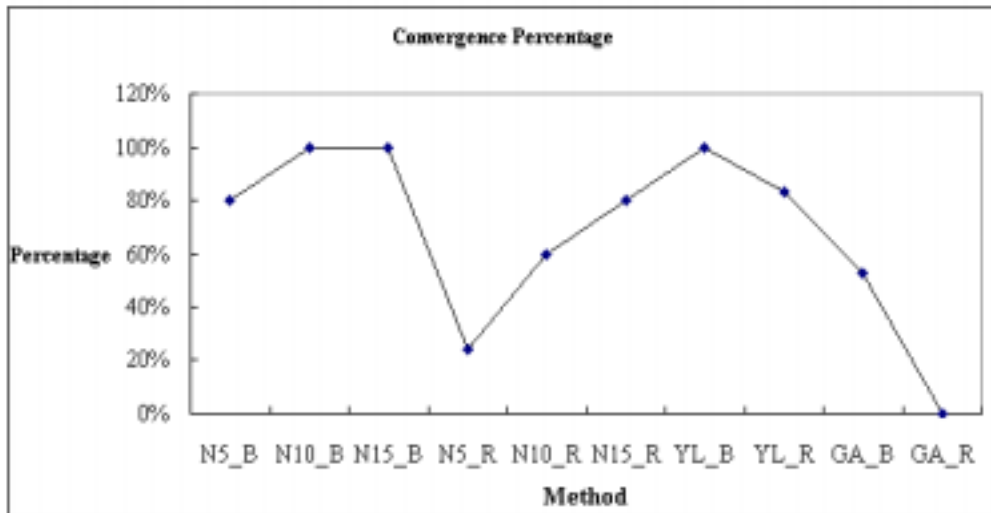


圖 4. 49 收斂率比較圖(算例七)

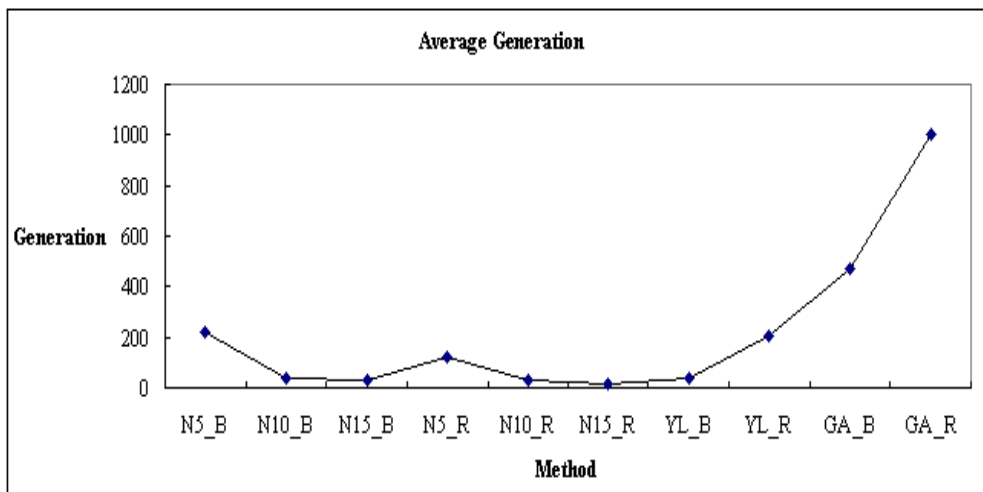


圖 4. 50 平均收斂代數比較圖(算例七)

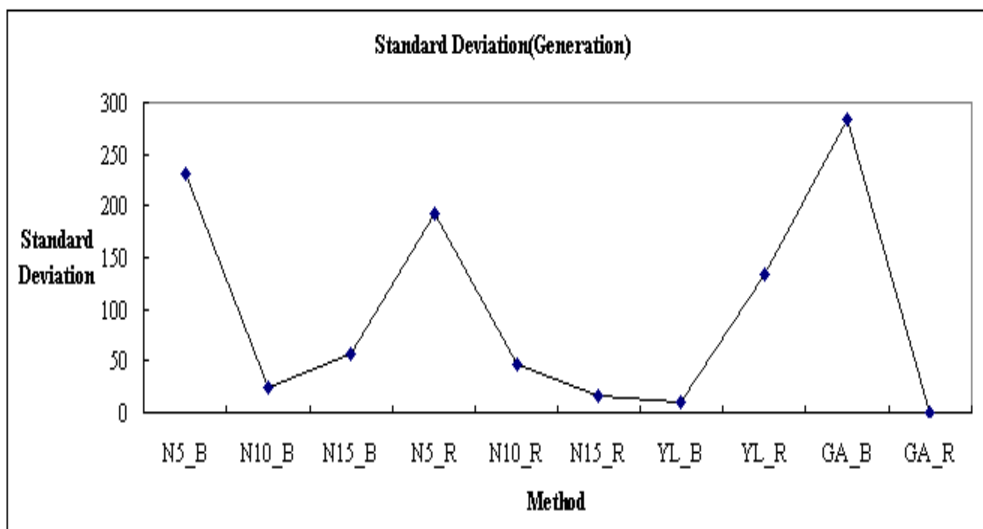


圖 4. 51 平均收斂代數之標準差比較圖(算例七)

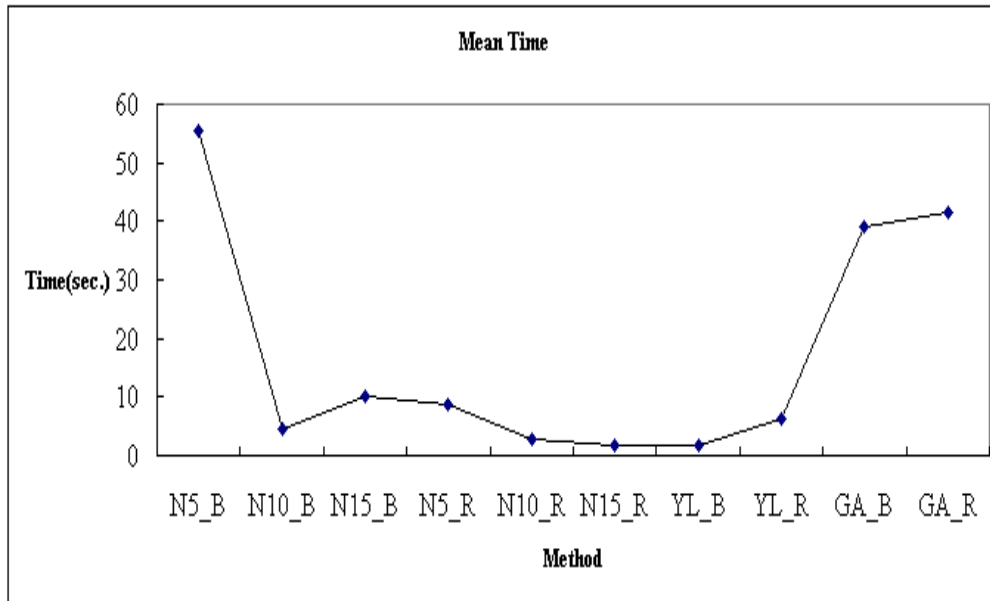


圖 4.52 達收斂時之平均運算時間比較圖(算例七)

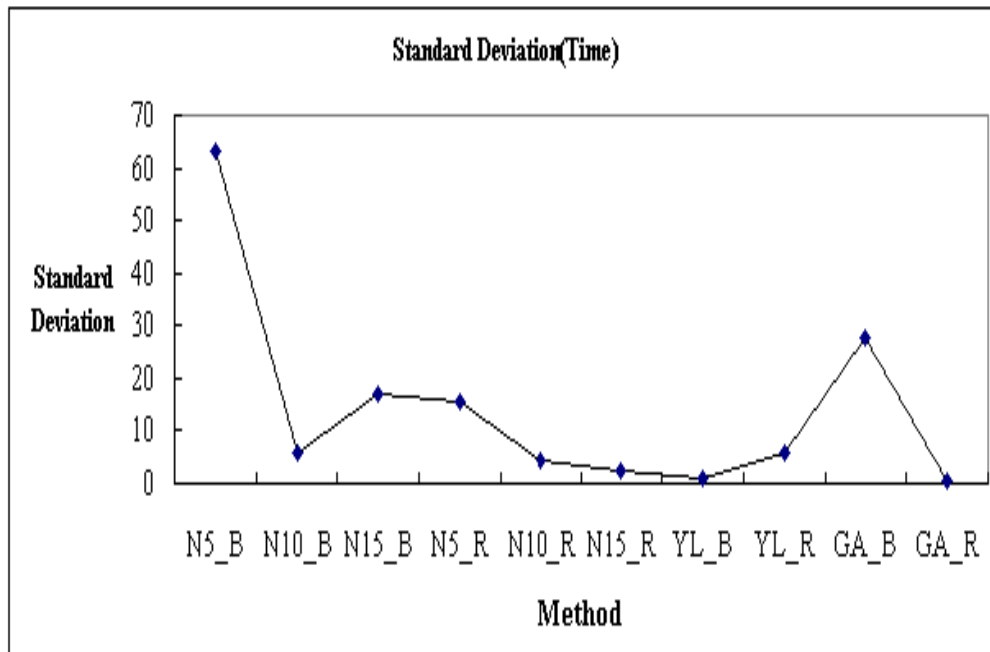


圖 4.53 平均運算時間之標準差比較圖(算例七)

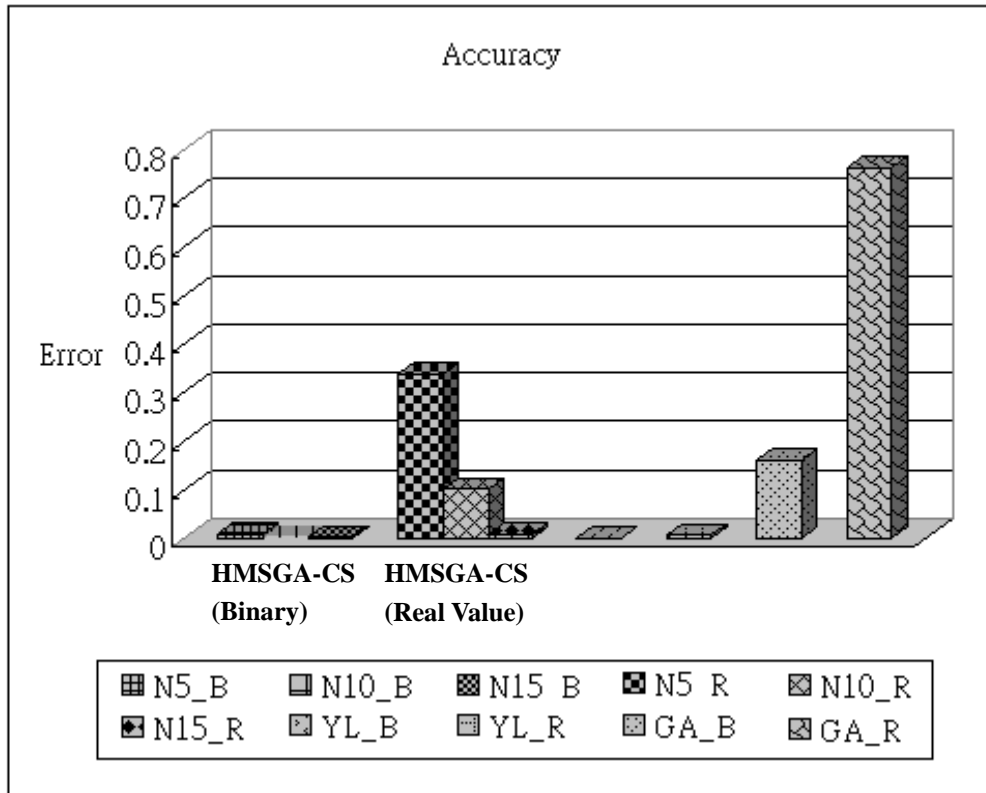


圖 4.54 與理想值之誤差比較圖(算例七)

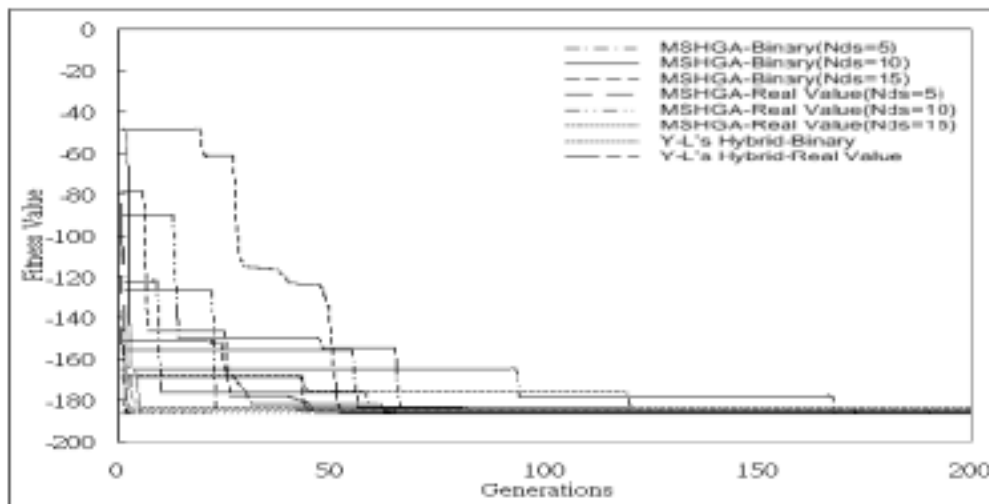


圖 4.55 代數收斂表示圖(算例七)

4.1.8 算例(八) : SinCos function

(1)目標函數

目標函數如(4-8)所示，此函數目的在求此函數的最小值。圖 4.56 為此目標函數之圖形。

$$\text{Min. } F(X) = \frac{1}{10} \sqrt{(x_1 - 25)^2 + (x_2 - 40)^2} + 10 - 10 \sin\left(\frac{\pi}{2} x_1\right) \cos\left(\frac{3\pi}{4} x_2\right)$$

(4- 8)

where $X_i \in [-256, 256]$

此函數之理想(Ideal)最佳值為 0 此函數有 20,480 個區域最佳解

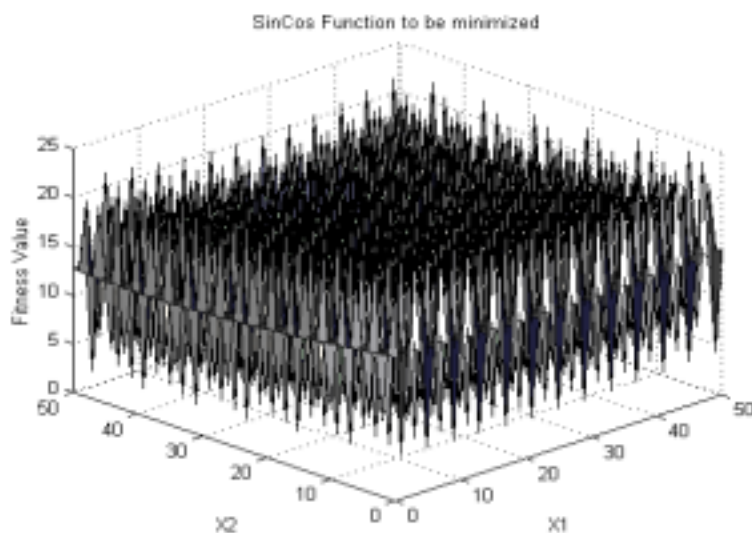


圖 4. 56 SinCos 之圖形

(2)測試環境與變數設定

此研究之相關變數如下：

- a. 族群大小：本範例母體大小為 42。
- b. 單一變數之基因數：單一變數之染色體長度為 26，精確度為 10^5 。
- c. 複製法則：輪盤法(Roulette)。
- d. 交配法則：以二進位制為編碼方式的交配方式為雙點交配。交配機率 $P_c = 0.4$ ；以十進位為編碼方式的交配方式為算術交配法 (Arithmetic Crossover)。
- e. 突變法則：以二進位制為編碼方式的突變為參考 2.1 節。突變機率 $P_m = 0.03$ ；以十進位為編碼方式的突變方式為均勻突變法 (Uniform Mutation)。
- f. 演化代數：在本範例中，我們取最大代數為 1000。
- g. 收斂條件：在本範例中，當與最佳值的誤差小於 0.001 時且連續

20 代皆如此，則視為收斂。

h. 測試次數：30 次。

i. 參數設定：在本範例中，菁英化單體數目為兩個；單體演化次數為 5 次、10 次和 15 次。

(3) 實驗結果與比較

圖 4. 57 至圖 4. 63 以及表 4. 21 為本研究實驗的結果。根據變異數分析(ANOVA)的結果(請參考附錄一表 A1. 22 及表 A1. 23), 方法的不同對收斂代數有顯著的影響且方法的不同對收斂時間有顯著的影響。

在表 4. 21 中，雖然 HMSGGA-CS 中有 5 組的『平均收斂代數』比二進位制(Binary)的 Y-L 混合法來的高，但我們必須注意另一數據，也就是『收斂百分比』，在該欄中，Y-L 混合法的『收斂百分比』皆不高，因此若以那些數據來平均或取標準差的話便能降低『平均收斂代數』和『標準差』的數據，所以若只看該欄位，結果並不準確；另外，雖然 Y-L 的計算時間比 HMSGGA-CS 短，但我們計算的基準是在以有達收斂標準的實驗為主，所以其結果也會受到影響。另一方面，在求出的『平均最佳值』中，Y-L 混合法及 GA 也不如 HMSGGA-CS 的精確。

表 4. 21 測試結果表(算例八)

Method		Convergence Percentage	Average* Generation	Standard Deviation (Generation)	Mean* Time(sec.) (Convergence)	Mean Value	Accuracy	
HMSGGA-CS	Binary	Nds=5	96.67%	124.157	126.971	29.679	0.0048	0.003395814
		Nds=10	90%	127.588	244.22	37.90	0.032	0.022650947
		Nds=15	100%	46.6	124.22	15.648	0.00026	0.000186834
	Real Value	Nds=5	80%	159.33	140.1	8.66	0.048	0.034106293
		Nds=10	60.00%	131.33	46.83	11.16	0.096	0.068131357
		Nds=15	73%	110.91	148.97	14.366	0.00871	0.061592153
Y-L (Binary)		36.67%	111.857	145.889	13.526	0.276	0.195028581	
Y-L (Real Value)		6.67%	320	0	9.514	0.504	0.356037257	
GA (Binary)		60.00%	306.94	235.294	24.909	0.10328	0.073033372	
GA (Real-value)		0%	1000	0	53.346	0.78149	0.552602597	

*代表只計算有收斂的數值。當『收斂百分比』為 0 時，『平均收斂時間』取演算 1000 代的平均時間。

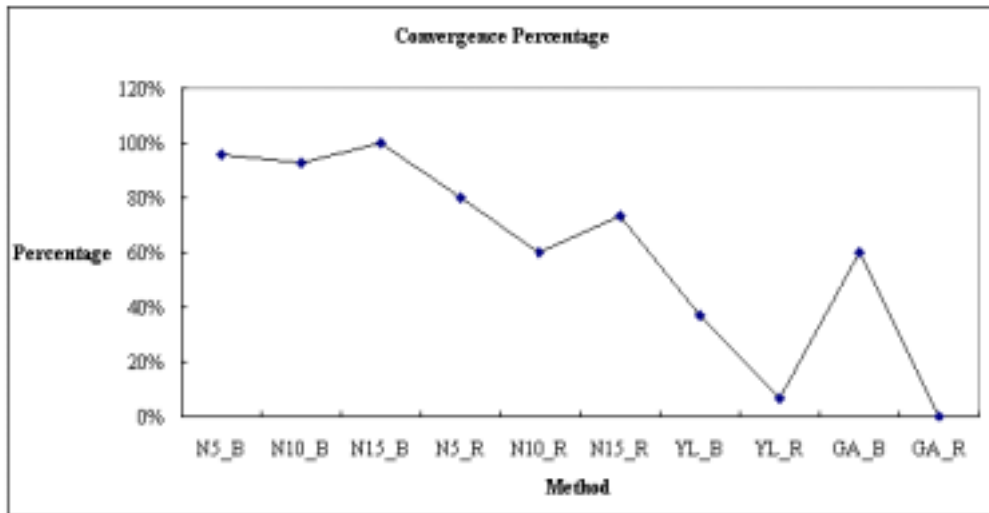


圖 4.57 收斂率比較圖(算例八)

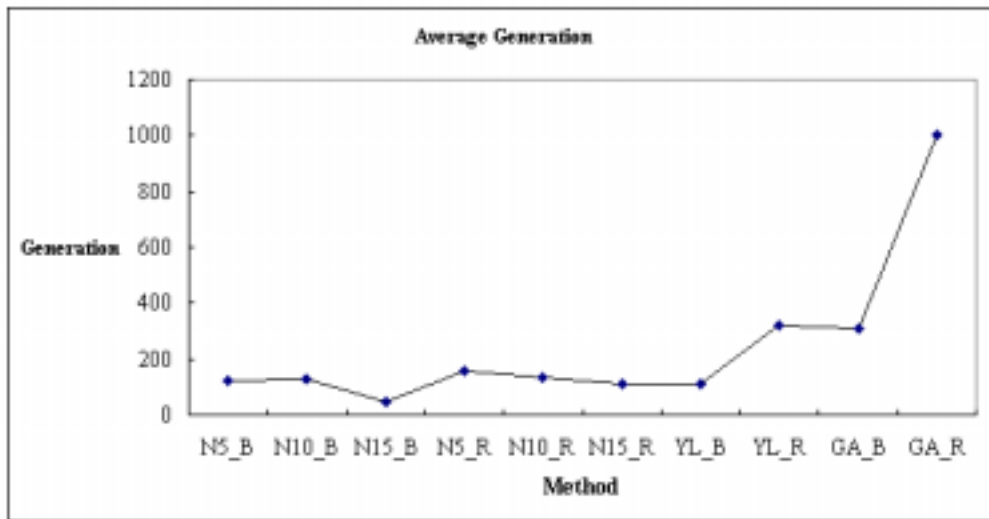


圖 4.58 平均收斂代數比較圖(算例八)



圖 4.59 平均收斂代數之標準差比較圖(算例八)

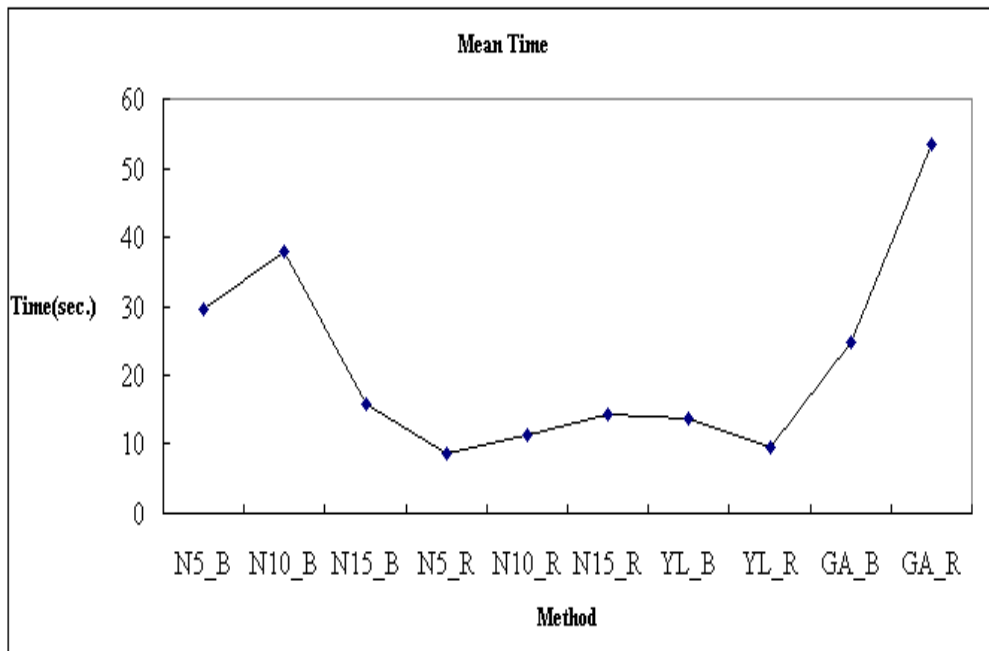


圖 4. 60 達收斂時之平均運算時間比較圖(算例八)

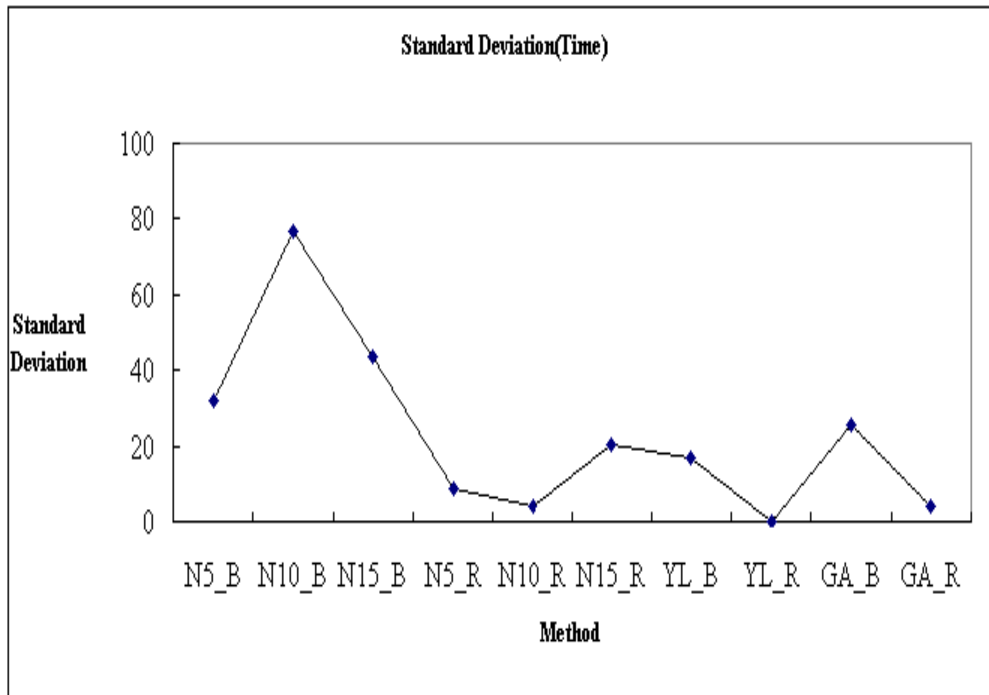


圖 4. 61 平均運算時間之標準差比較圖(算例八)

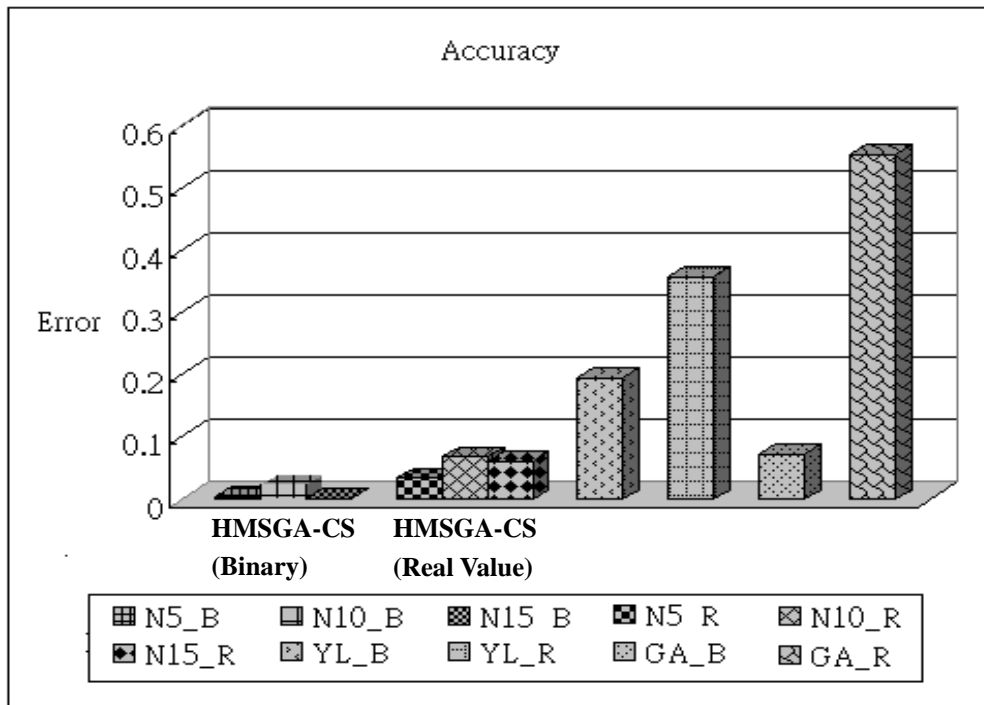


圖 4.62 與理想值之誤差比較圖(算例八)

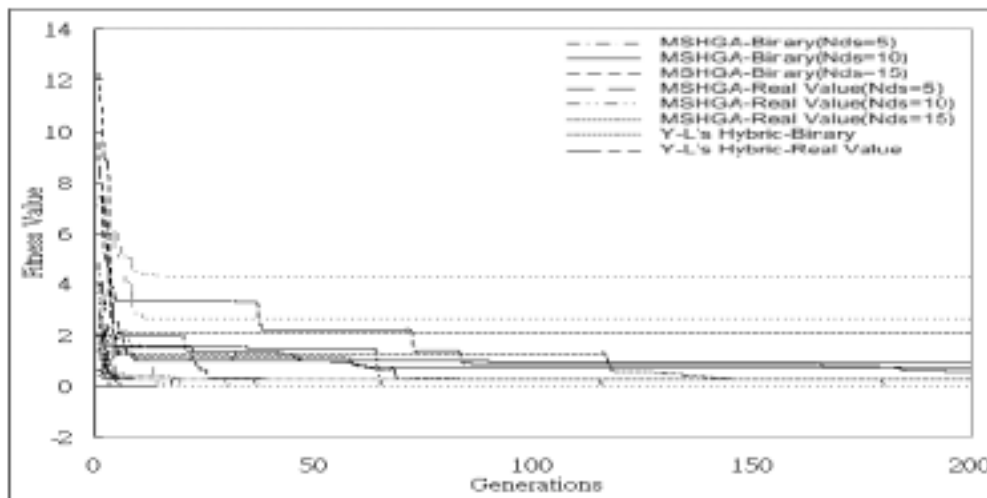


圖 4.63 代數收斂表示圖(算例八)

4.1.9 算例(九) : Goldstein Price function

(1) 目標函數

目標函數如(4-9)所示，此函數目的在求此函數的最小值。圖 4.64 為此目標函數之圖形。

Min.

$$F(X) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \\ \cdot [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$$

(4-9)

where $X_i \in [-2, 2]$

此函數之理想(Ideal)最佳值為 3。

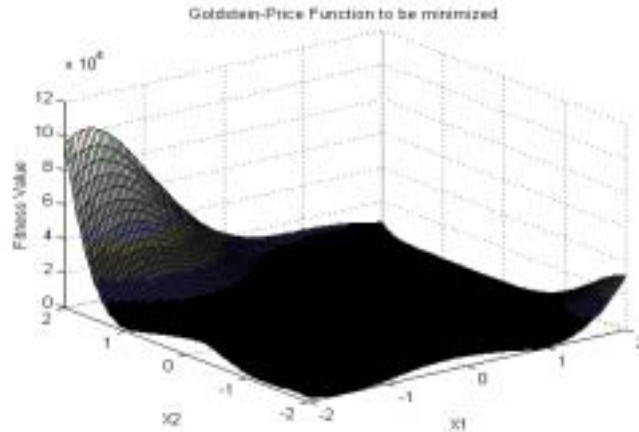


圖 4. 64 Goldstein-Price 之圖形

(2)測試環境與變數設定

此研究之相關變數如下：

- a. 族群大小：本範例母體大小為 42。
- b. 單一變數之基因數：單一變數之染色體長度為 19，精確度為 10^5 。
- c. 複製法則：輪盤法(Roulette)。
- d. 交配法則：以二進位制為編碼方式的交配方式為雙點交配。交配機率 $P_c = 0.4$ ；以十進位為編碼方式的交配方式為算術交配法 (Arithmetic Crossover)。
- e. 突變法則：以二進位制為編碼方式的突變為參考 2.1 節。突變機率 $P_m = 0.03$ ；以十進位為編碼方式的突變方式為均勻突變法 (Uniform Mutation)。
- f. 演化代數：在本範例中，我們取最大代數為 1000。
- g. 收斂條件：在本範例中，當與最佳值的誤差小於 0.001 時且連續 20 代皆如此，則視為收斂。
- h. 測試次數：30 次。

- i. 參數設定：在本範例中，菁英化單體數目為兩個；單體演化次數為 5 次、10 次和 15 次。

(3)實驗結果與比較

圖 4. 65 至圖 4. 71 以及表 4. 22 為本研究實驗的結果。根據變異數分析(ANOVA)的結果(請參考附錄一表 A1. 24 及表 A1. 26), 方法的不同對收斂代數有顯著的影響且方法的不同對收斂時間有顯著的影響。

在表 4. 22 中，以『收斂百分比』來說，除了實數型(Real Value)的 Y-L 及 GA 方法有較差的結果外，其他的組合都有接近 100%的收斂率。對於『平均收斂代數』來說，HMSGGA-CS 所有組合的結果皆比 Y-L 及 GA 的方法來的好且在『標準差』方面也比 Y-L 的來的低，顯示 HMSGGA-CS 比其它方法穩定。在『平均收斂時間』方面，HMSGGA-CS 的計算時間也比其它兩種方法的短。除了 GA 以外，在『平均最佳值』方面，各種組合的差異並不大。

另一方面，經變異數分析的結果顯示 HMSGGA-CS 面對不同的資料型態上與不同的單體演化代數不會對『平均收斂代數』(p-value=0.051>，請參考附錄一表 A1. 25)與『平均收斂時間』(p-value=0.381>，請參考附錄一表 A1. 27)有顯著的差異。

表 4. 22 測試結果表(算例九)

Method			Convergence Percentage	Average* Generation	Standard Deviation (Generation)	Mean* Time(sec.) (Convergence)	Mean Value	Accuracy
HMSGGA-CS	Binary	Nds=5	100.00%	28.333	14.40569	1.80527	3.00034	2.427E-04
		Nds=10	100.00%	25.200	2.51282	1.20440	3.00029	2.070E-04
		Nds=15	100.00%	24.600	3.94244	1.24573	3.00029	2.044E-04
	Real Value	Nds=5	100.00%	35.667	7.01699	0.76447	3.00039	2.758E-04
		Nds=10	100.00%	34.733	25.23169	1.34793	3.00034	2.374E-04
		Nds=15	100.00%	28.267	13.45079	1.09353	3.00035	2.445E-04
Y-L (Binary)			100.00%	95.733	236.35309	10.40567	3.00047	3.355E-04
Y-L (Real Value)			67.67%	422.300	380.55108	19.84940	3.00047	4.445E-03
GA (Binary)			96.67%	114.207	91.1578	6.1769	5.07162	1.91033
GA (Real Value)			10.00%	504.667	290.5173	13.6396	9.2	4.38447

*代表只計算有收斂的數值

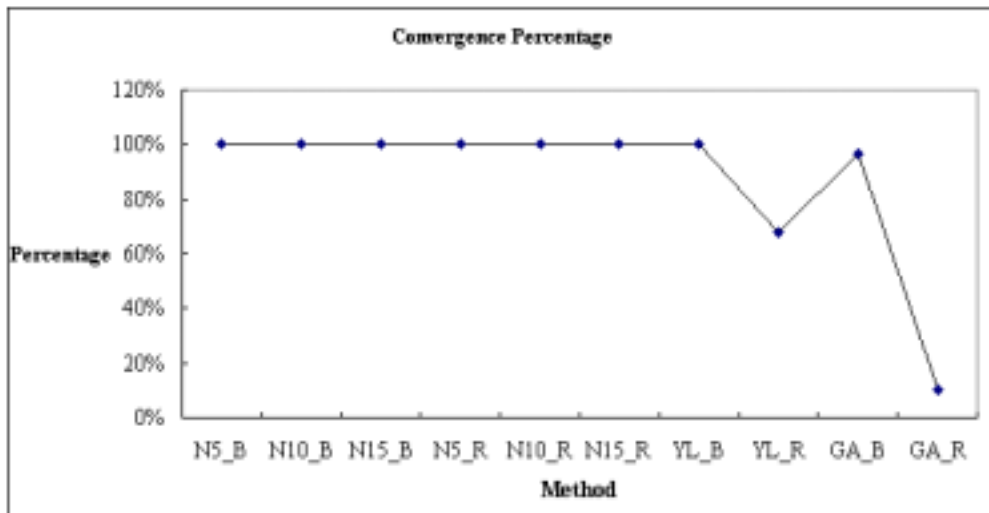


圖 4.65 收斂率比較圖(算例九)

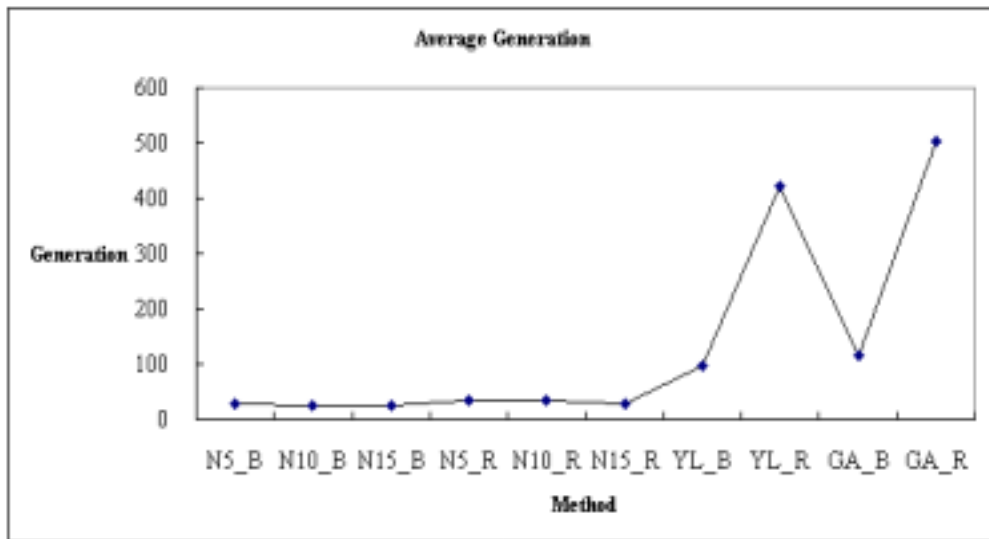


圖 4.66 平均收斂代數比較圖(算例九)

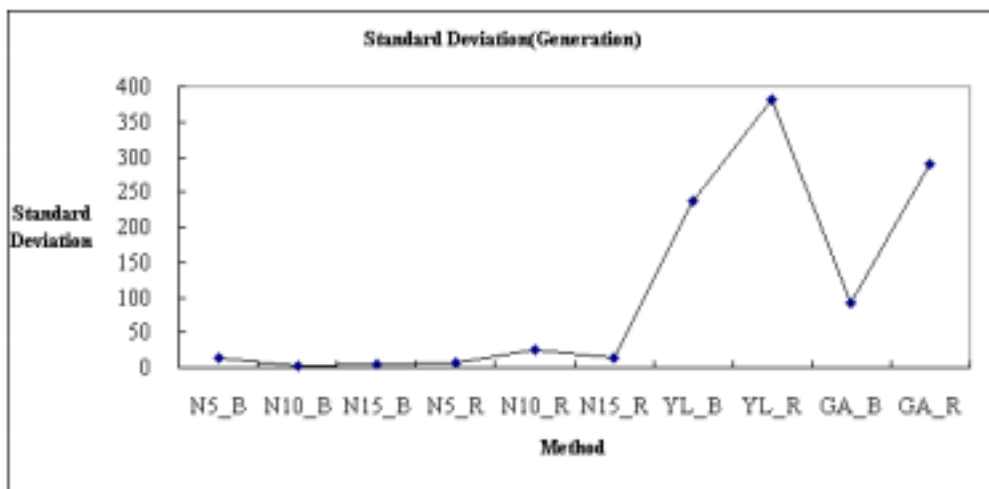


圖 4.67 平均收斂代數之標準差比較圖(算例九)

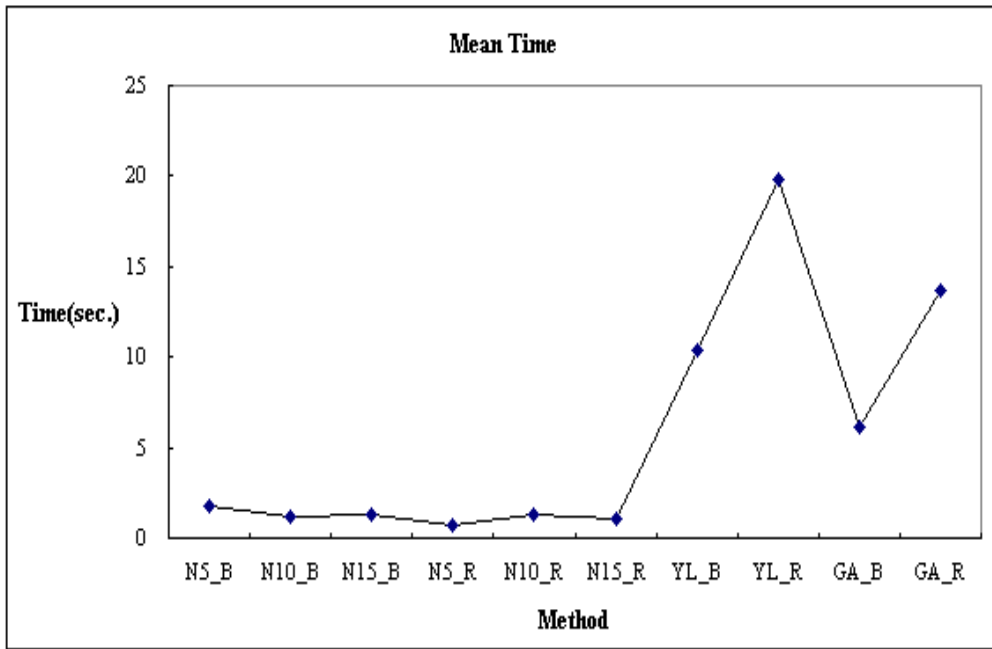


圖 4.68 達收斂時之平均運算時間比較圖(算例九)

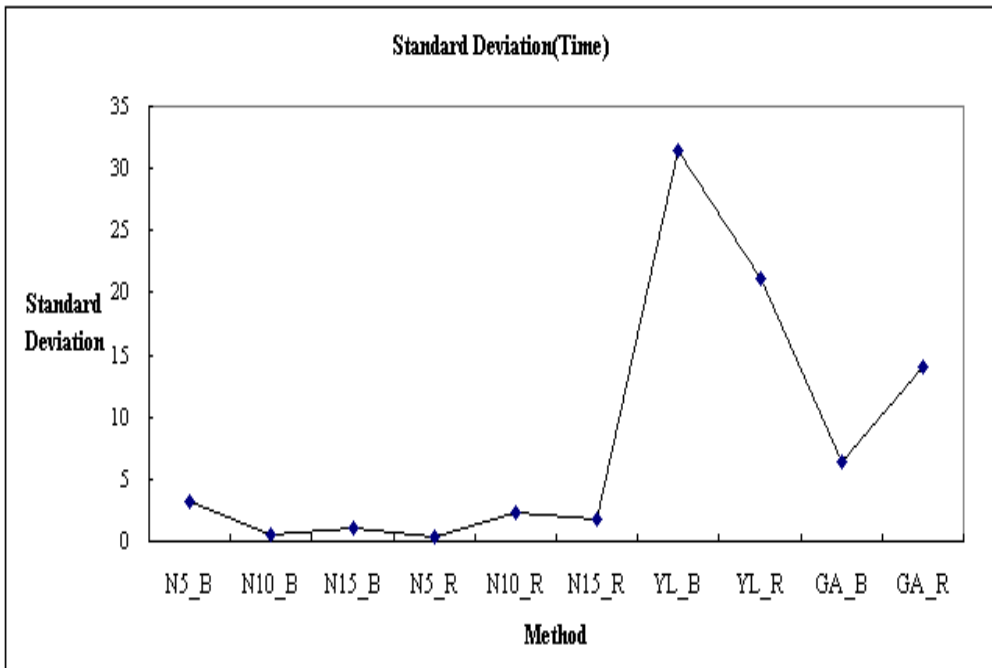


圖 4.69 平均運算時間之標準差比較圖(算例九)

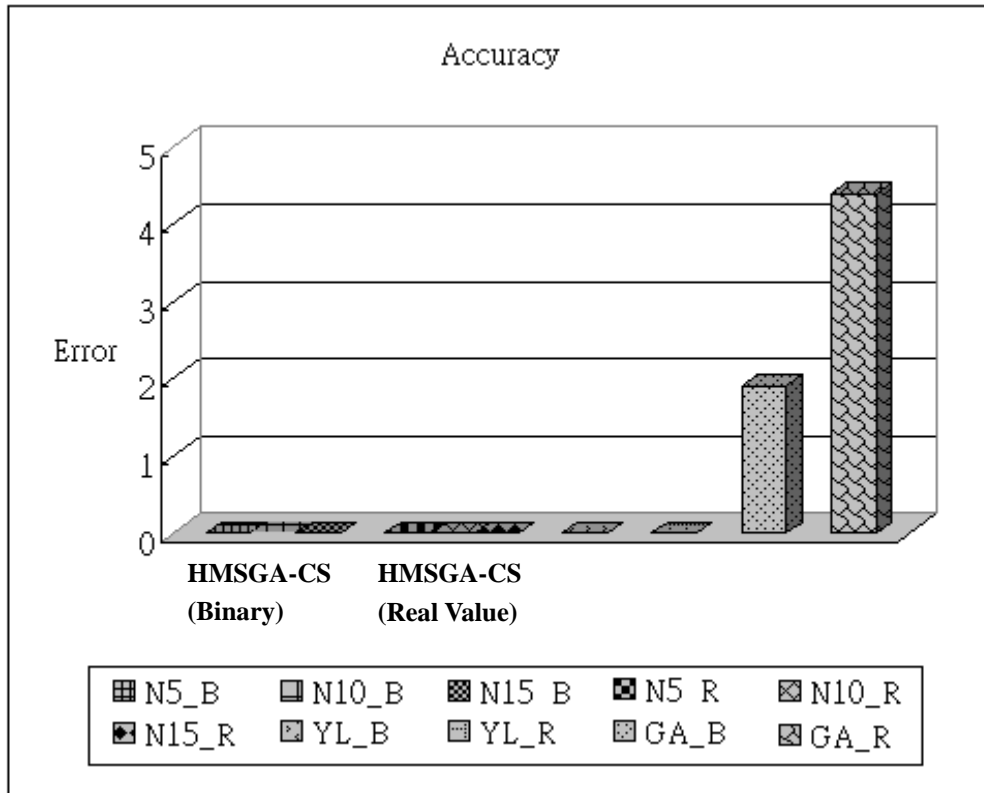


圖 4.70 與理想值之誤差比較圖(算例九)

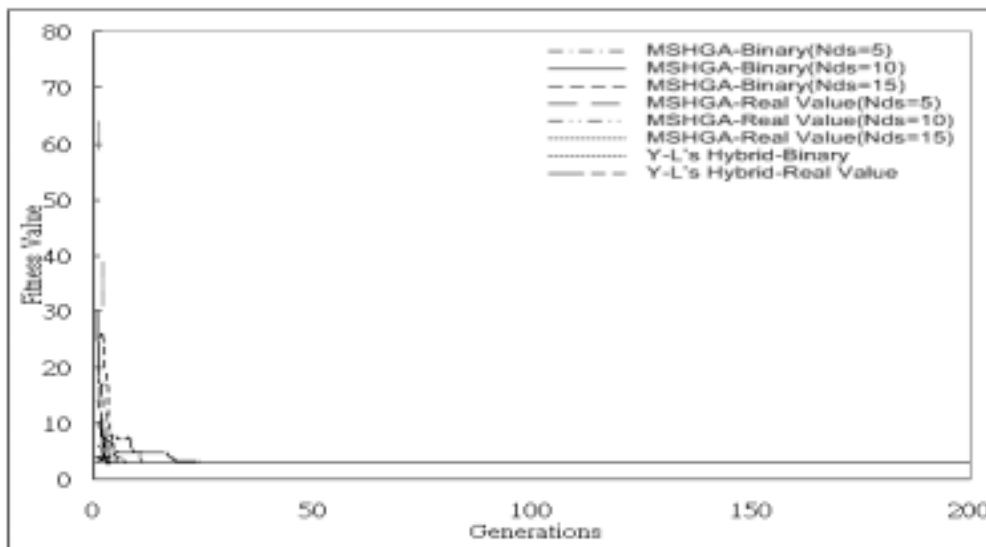


圖 4.71 代數收斂表示圖(算例九)

4.1.10 算例(十) : Bohachevsky function # 1

(1) 目標函數

目標函數如(4-10)所示,此函數目的在求此函數的最小值 圖 4.72

為此目標函數之圖形。

$$\text{Min. } F(X) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7 \quad (4-10)$$

where $X_i \in [-50, 50]$

此函數之理想(Ideal)最佳值為 0。

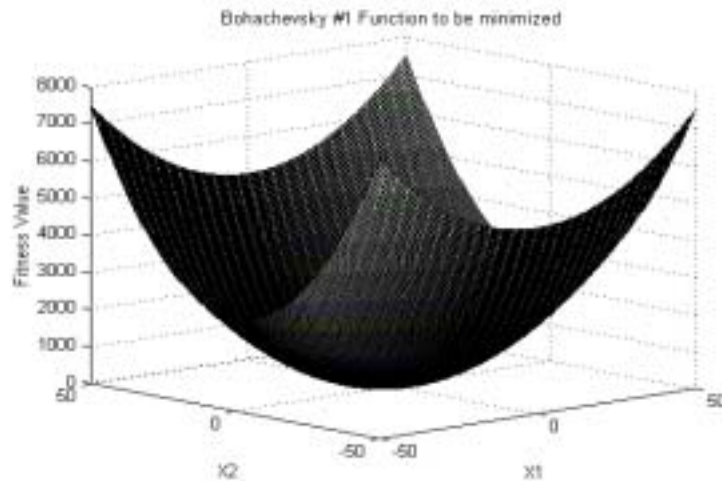


圖 4. 72 Bohachevsky # 1 之圖形

(2)測試環境與變數設定

此研究之相關變數如下：

- 族群大小：本範例母體大小為 42。
- 單一變數之基因數：單一變數之染色體長度為 23，精確度為 10^5 。
- 複製法則：輪盤法(Roulette)。
- 交配法則：以二進位制為編碼方式的交配方式為雙點交配。交配機率 $P_c = 0.4$ ；以十進位為編碼方式的交配方式為算術交配法 (Arithmetic Crossover)。
- 突變法則：以二進位制為編碼方式的突變為參考 2.1 節。突變機率 $P_m = 0.03$ ；以十進位為編碼方式的突變方式為均勻突變法 (Uniform Mutation)。
- 演化代數：在本範例中，我們取最大代數為 1000。
- 收斂條件：在本範例中，當與最佳值的誤差小於 0.001 時且連續 20 代皆如此，則視為收斂。

h. 測試次數：30 次。

i. 參數設定：在本範例中，菁英化單體數目為兩個；單體演化次數為 5 次、10 次和 15 次。

(3) 實驗結果與比較

圖 4. 73 至圖 4. 79 以及表 4. 23 為本研究實驗的結果。根據變異數分析(ANOVA)的結果(請參考附錄一表 A1. 28 及表 A1. 30), 方法的不同對收斂代數有顯著的影響且方法的不同對收斂時間有顯著的影響。

在表 4. 23 中，以『收斂百分比』來說，在實數型(Real Value)的 Y-L 及 GA 中，其比例偏低，不如其他的組合來的好。以『平均收斂代數』來說，HMSGGA-CS 的各種組合對其並無顯著的差異($p\text{-value}=0.331>$ ，請參考附錄一表 A1. 29)，且勝於 Y-L 的方法。以『標準差』值來說，HMSGGA-CS 在二進位制(Binary)的組合中有相當穩定的表現，但在實數型(Real Value)的組合中則略顯不佳，反而是在 Nds=5 時有較佳的結果。在『平均收斂時間』方面，HMSGGA-CS 皆能在短時間內達到目標且所有組合經變異數分析後證實並無顯著差異($p\text{-value}=0.574>$ ，請參考附錄一表 A1. 31)。

表 4. 23 測試結果表(算例十)

Method		Convergence Percentage	Average* Generation	Standard Deviation (Generation)	Mean* Time(sec.) (Convergence)	Mean Value	Accuracy	
HMSGGA-CS	Binary	Nds=5	100.00%	23.6667	1.1751	0.7625	0.0003	1.93284E-04
		Nds=10	100.00%	22.8000	1.1464	0.6716	0.0002	1.09851E-04
		Nds=15	100.00%	22.4667	1.8074	0.6770	0.0002	1.71466E-04
	Real Value	Nds=5	93.33%	27.0714	7.7506	0.3119	0.0035	2.509559E-03
		Nds=10	100.00%	26.5333	14.8077	0.5073	0.0001	8.92169E-05
		Nds=15	100.00%	26.9333	15.2759	0.7677	0.0002	1.34163E-04
Y-L (Binary)		100.00%	35.7333	13.0245	1.4321	0.0005	3.6797E-04	
Y-L (Real Value)		40.00%	93.1667	135.0650	1.9595	0.0468	3.312373E-02	
GA (Binary)		100%	45.83	8.5423	2.93	0.0005	3.331E-04	
GA (Real Value)		3%	337	0	35.534	0.1449	0.10248	

*代表只計算有收斂的數值

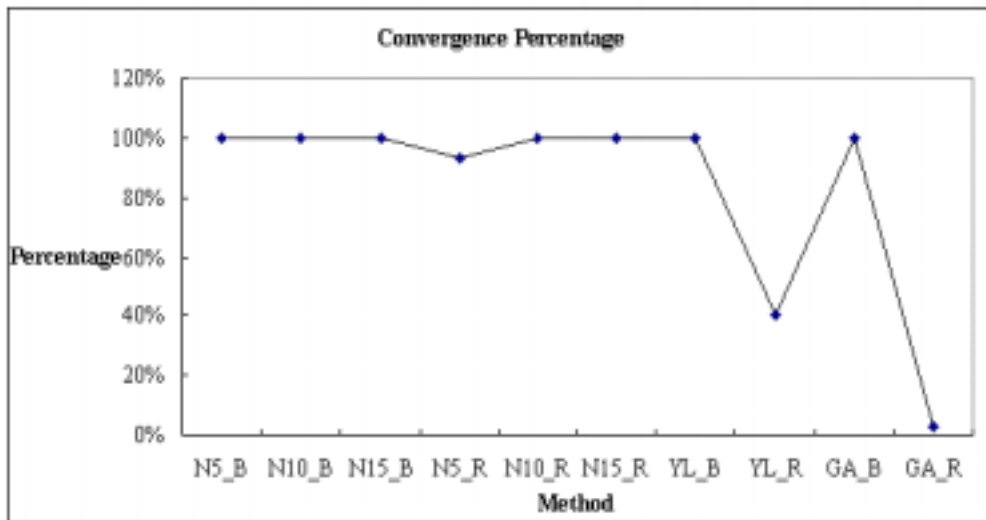


圖 4.73 收斂率比較圖(算例十)

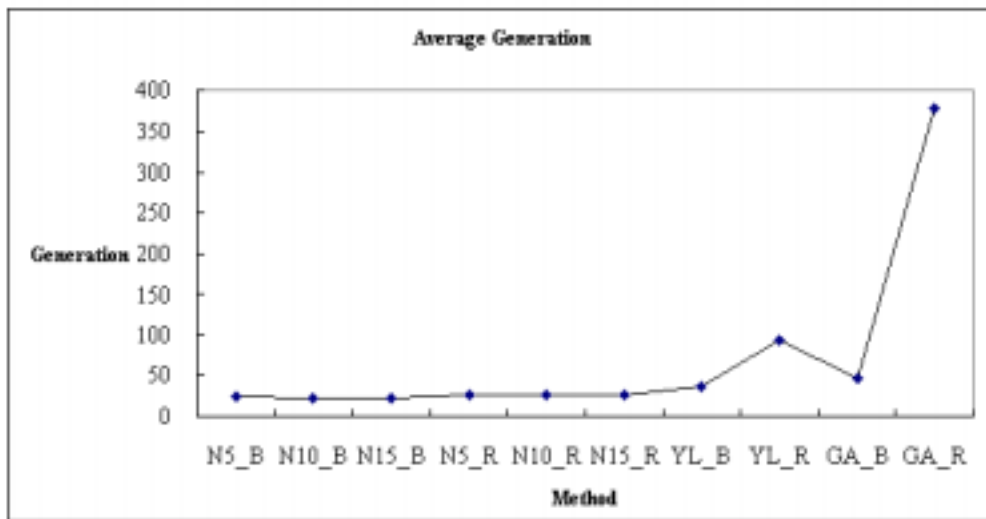


圖 4.74 平均收斂代數比較圖(算例十)

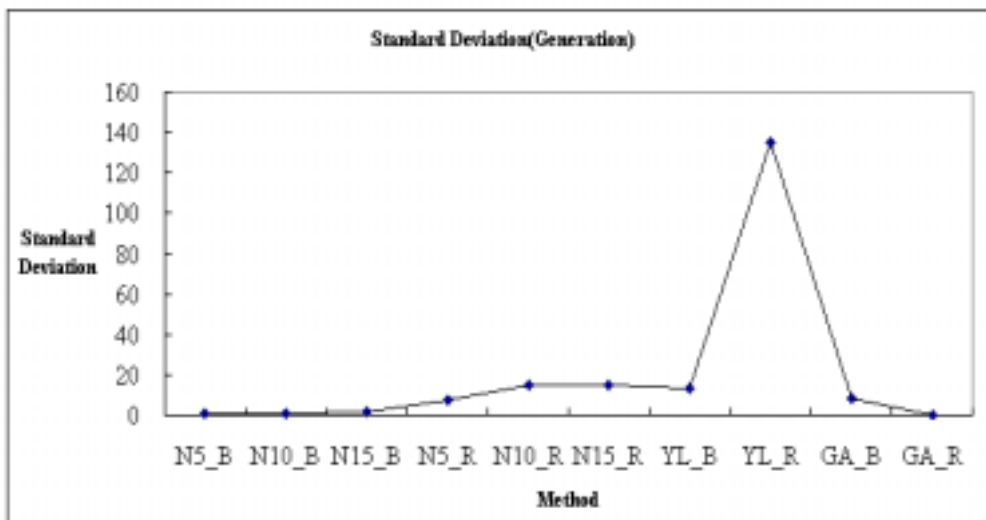


圖 4.75 平均收斂代數之標準差比較圖(算例十)

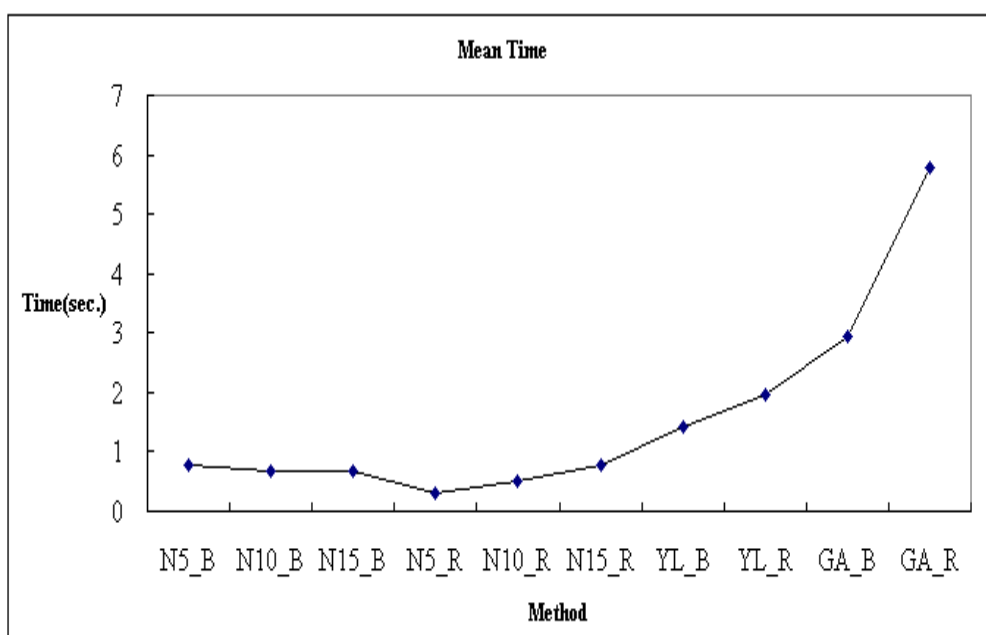


圖 4.76 達收斂時之平均運算時間比較圖(算例十)

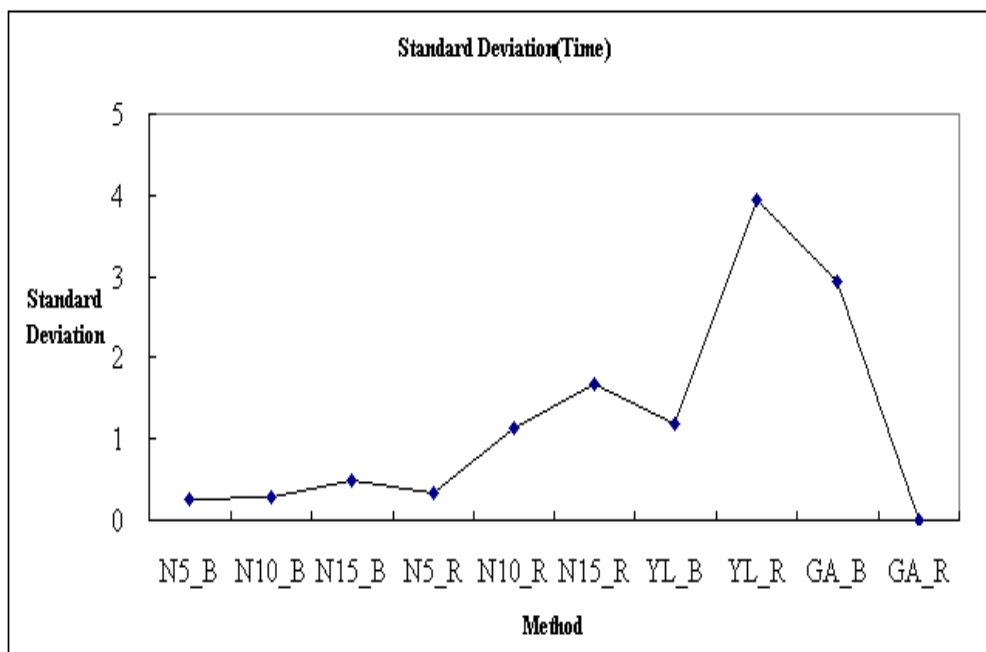


圖 4.77 平均運算時間之標準差比較圖(算例十)

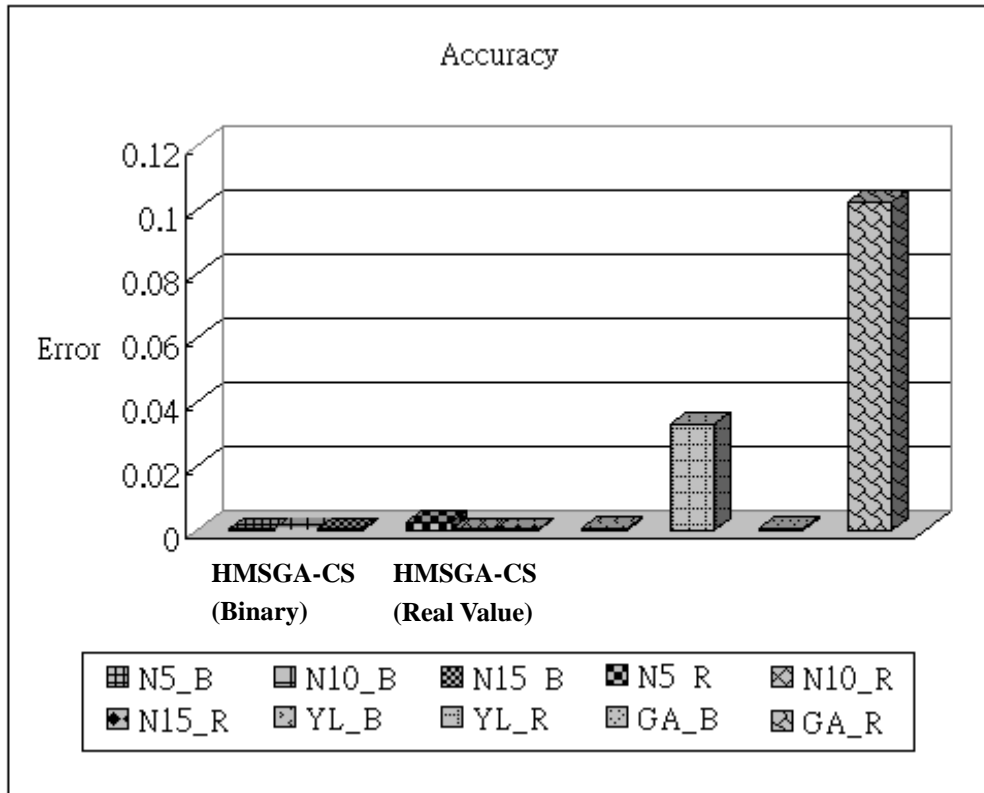


圖 4.78 與理想值之誤差比較圖(算例十)

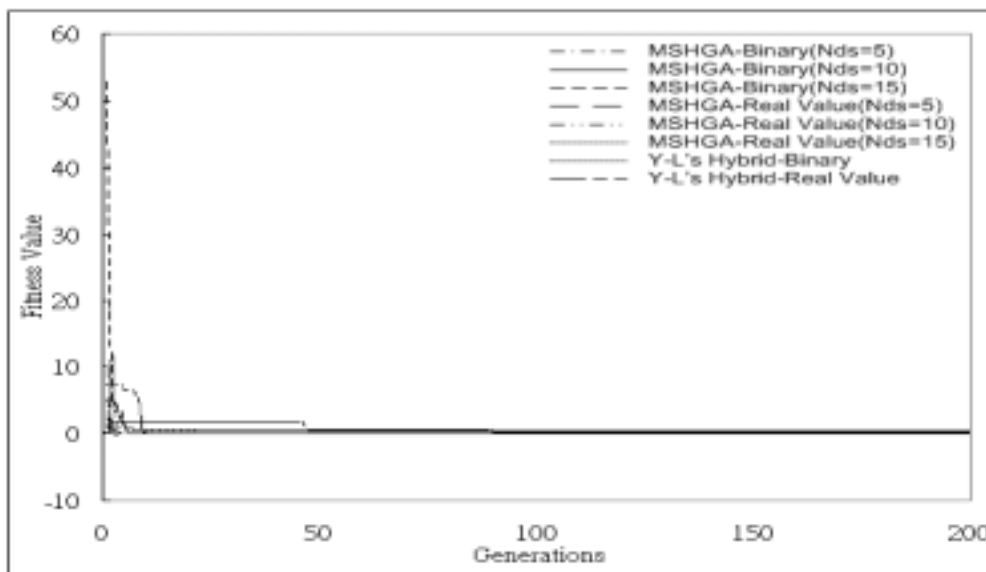


圖 4.79 代數收斂表示圖(算例十)

4.1.11 算例(十一) : Easom function

(1) 目標函數

目標函數如(4-11)所示,此函數目的在求此函數的最小值 圖 4.80

為此目標函數之圖形。

$$\text{Min. } F(X) = -\cos(x_1) \cos(x_2) e^{-(x_1-\pi)^2 - (x_2-\pi)^2} \quad (4-11)$$

Where $X_i \in [-100, 100]$

此函數之理想(Ideal)最佳值為-1。

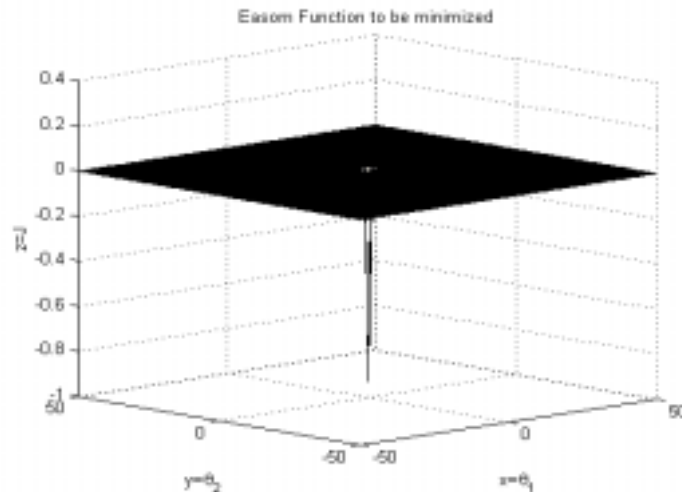


圖 4. 80 Easom 之圖形

(2)測試環境與變數設定

此研究之相關變數如下：

- 族群大小：本範例母體大小為 42。
- 單一變數之基因數：單一變數之染色體長度為 25，精確度為 10^5 。
- 複製法則：輪盤法(Roulette)。
- 交配法則：以二進位制為編碼方式的交配方式為雙點交配。交配機率 $P_c = 0.4$ ；以十進位為編碼方式的交配方式為算術交配法 (Arithmetic Crossover)。
- 突變法則：以二進位制為編碼方式的突變為參考 2.1 節。突變機率 $P_m = 0.03$ ；以十進位為編碼方式的突變方式為均勻突變法 (Uniform Mutation)。
- 演化代數：在本範例中，我們取最大代數為 1000。
- 收斂條件：在本範例中，當與最佳值的誤差小於 0.001 時且連續 20 代皆如此，則視為收斂。

h. 測試次數：30 次。

i. 參數設定：在本範例中，菁英化單體數目為兩個；單體演化次數為 5 次、10 次和 15 次。

(3) 實驗結果與比較

圖 4. 81 至圖 4. 87 以及表 4. 24 為本研究實驗的結果。根據變異數分析(ANOVA)的結果(請參考附錄一表 A1. 32 及表 A1. 33), 方法的不同對收斂代數有顯著的影響且方法的不同對收斂時間有顯著的影響。

在『收斂百分比』方面，實數型(Real Value)的 Y-L 方法達到目標值的次數並不多，GA 更是在 30 次的實驗中皆無達到所設定的停止條件。在『平均收斂代數』方面，HMSGGA-CS 的組合結果皆比 Y-L 及 GA 的方法來的好，且在二進位制(Binary)的 Nds=10 和 Nds=15 的組合中以及十進位制(Real Value)的各種組合中，經變異數分析後其結果並無顯著差異(p-value=0.064> , p-value=0.074>)。在『平均收斂時間』方面，HMSGGA-CS 在十進位制(Real Value)的組合中比 Y-L 的方法低，且其結果不顯著(p-value=0.775>)。

表 4. 24 測試結果表(算例十一)

Method		Convergence Percentage	Average* Generation	Standard Deviation (Generation)	Mean* Time(sec.) (Convergence)	Mean Value	Accuracy	
HMSGGA-CS	Binary	Nds=5	100.00%	44.0000	21.9013	3.8695	-1.0000	0
		Nds=10	100.00%	28.6800	2.4956	2.2759	-1.0000	0
		Nds=15	100.00%	25.5263	2.6113	3.1384	-1.0000	0
	Real Value	Nds=5	100.00%	40.3333	27.472	0.9234	-1.0000	1.4901E-08
		Nds=10	100.00%	29.2000	4.1952	0.7077	-1.0000	0
		Nds=15	100.00%	27.2667	6.0293	0.7978	-1.0000	1.4901E-08
Y-L (Binary)		100.00%	60.2000	23.5590	3.8709	-1.0000	0	
Y-L (Real Value)		40.00%	167.1667	104.5111	3.5683	-0.6665	0.2358	
GA (Binary)		0%	1000	0	90.902	-0.4817	0.3664	
GA (Real Value)		0%	1000	0	40.320	-0.4306	0.4026	

*代表只計算有收斂的數值。當『收斂百分比』為 0 時，『平均收斂時間』取演算 1000 代的平均時間。

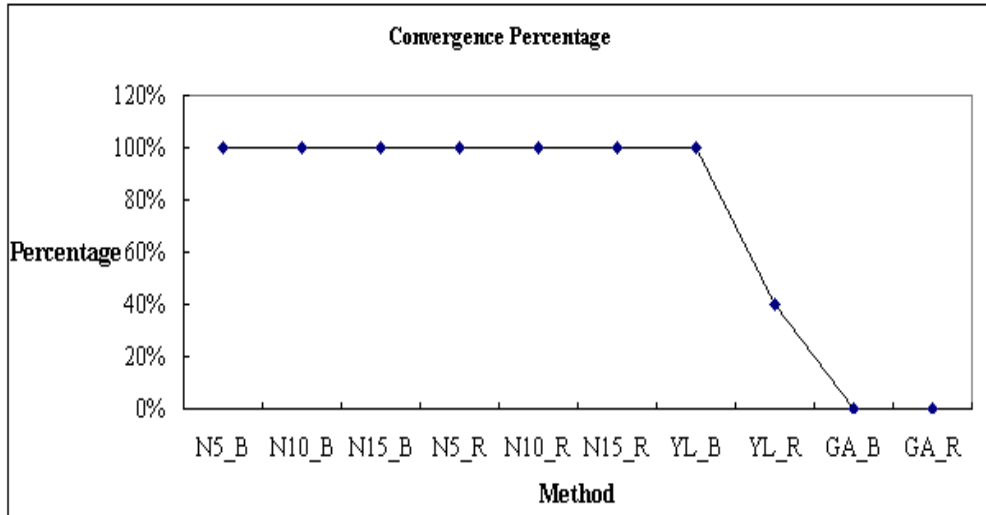


圖 4. 81 收斂率比較圖(算例十一)

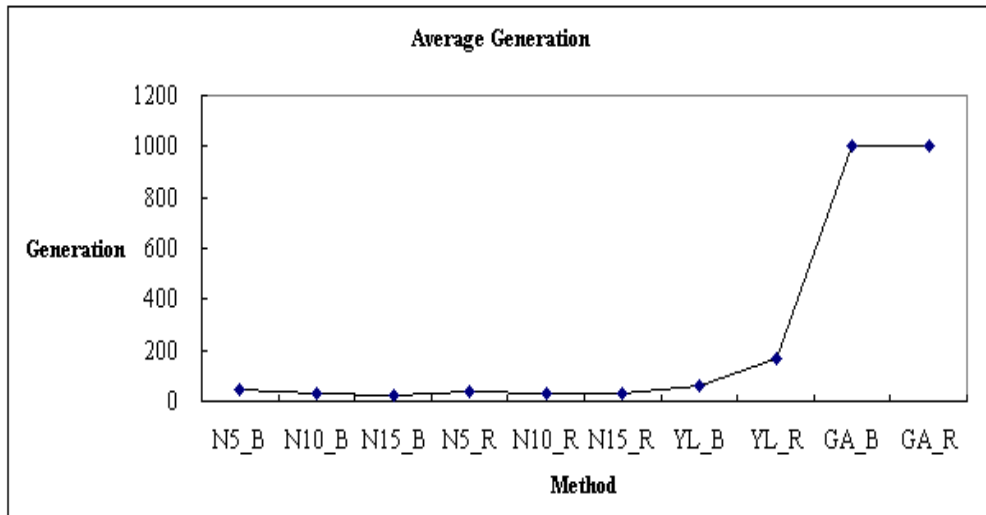


圖 4. 82 平均收斂代數比較圖(算例十一)

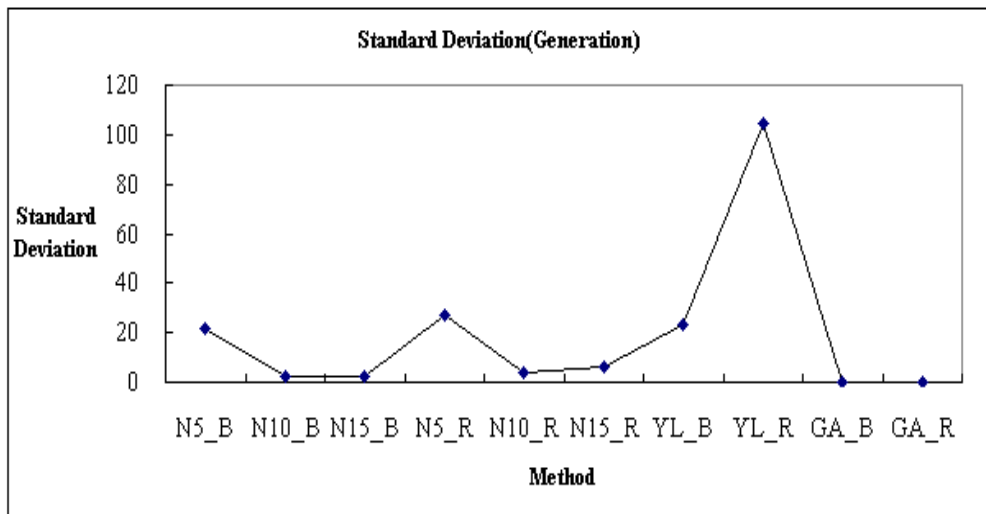


圖 4. 83 平均收斂代數之標準差比較圖(算例十一)

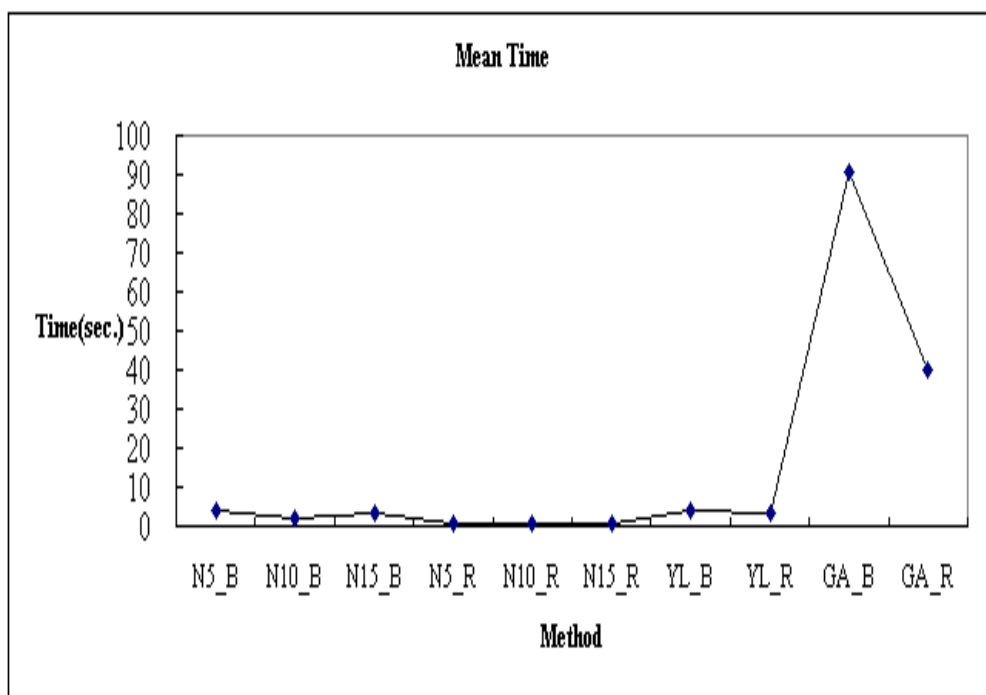


圖 4.84 達收斂時之平均運算時間比較圖(算例十一)

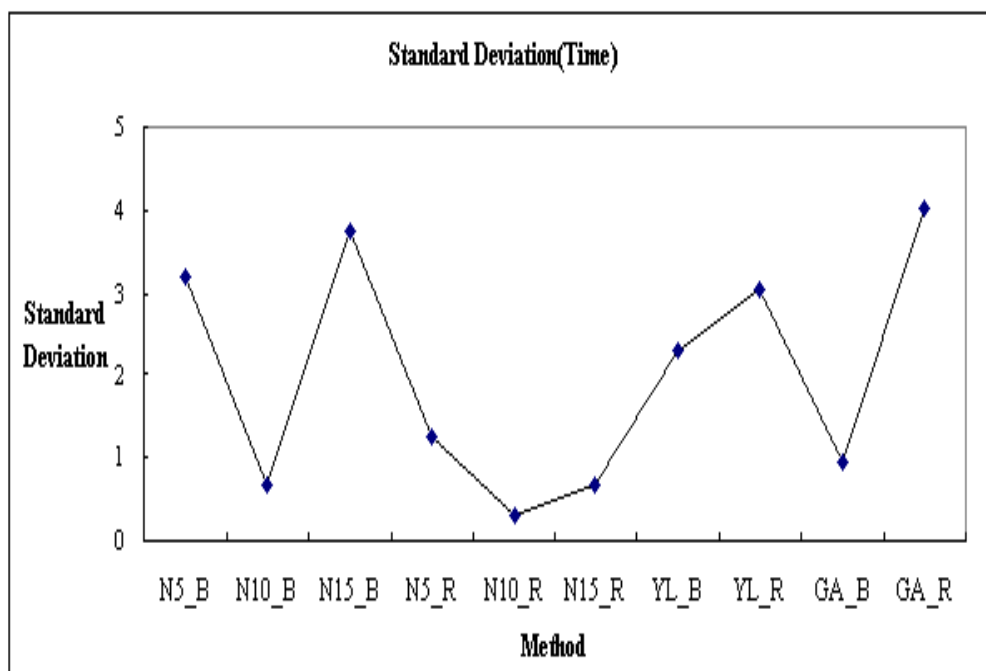


圖 4.85 平均運算時間之標準差比較圖(算例十一)

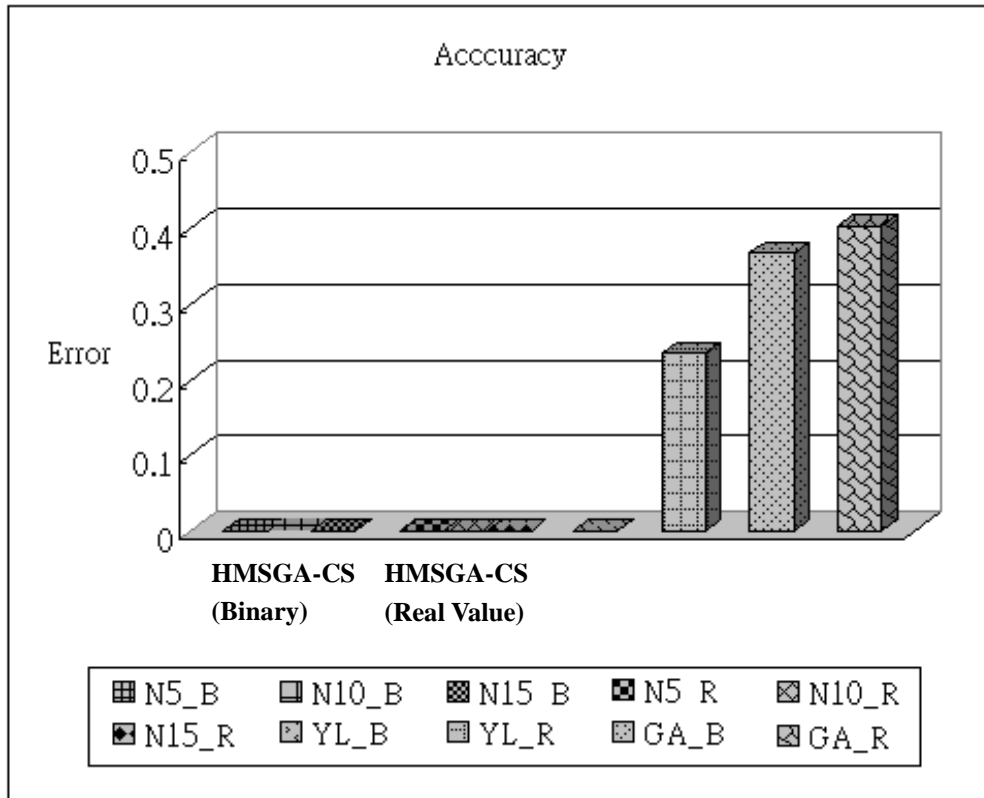


圖 4.86 與理想值之誤差比較圖(算例十一)

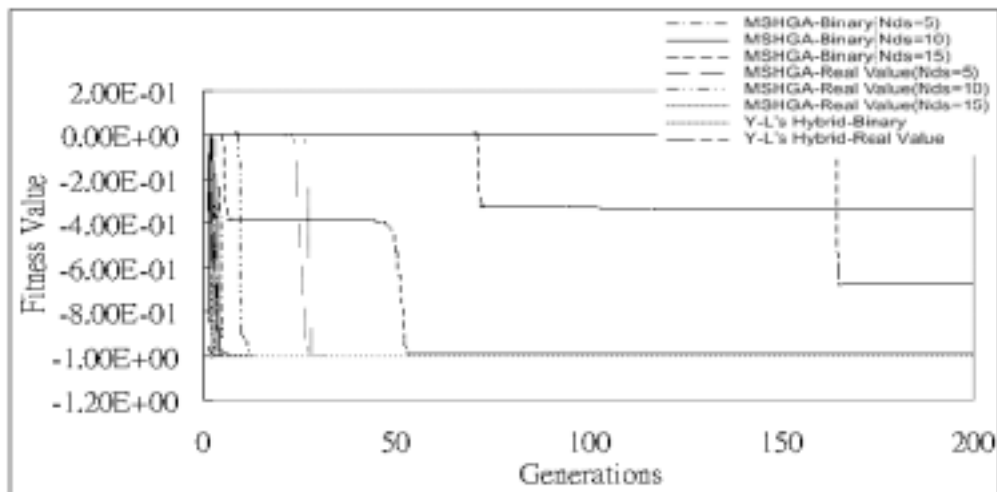


圖 4.87 代數收斂表示圖(算例十一)

4.1.12 算例(十二) : SQRN5 function

(1) 目標函數

目標函數如(4-12)所示，此函數目的在求此函數的最小值。

$$\text{Min. } F(X) = -\sum_{j=1}^5 \frac{1}{\sum_{i=1}^4 (x_i - a_{ij})^2 + c_j} \quad (4-12)$$

where $X_i \in [0,10]$

表 4. 25 SQRN5 function 之參數對照表

j	a_{1j}	a_{2j}	a_{3j}	a_{4j}	c_j
1	4.0	4.0	4.0	4.0	0.1
2	1.0	1.0	1.0	1.0	0.2
3	8.0	8.0	8.0	8.0	0.2
4	6.0	6.0	6.0	6.0	4.0
5	3.0	7.0	3.0	7.0	0.6
6	2.0	9.0	2.0	9.0	0.6
7	5.0	5.0	3.0	3.0	0.3
8	8.0	1.0	8.0	1.0	0.7
9	6.0	2.0	6.0	2.0	0.5
10	7.0	3.6	7.0	3.6	0.5

此函數之理想(Ideal)最佳值為-10.1527。

(2)測試環境與變數設定

此研究之相關變數如下：

- 族群大小：本範例母體大小為 40。
- 單一變數之基因數：單一變數之染色體長度為 20，精確度為 10^5 。
- 複製法則：輪盤法(Roulette)。
- 交配法則：以二進位制為編碼方式的交配方式為雙點交配。交配機率 $P_c = 0.4$ ；以十進位為編碼方式的交配方式為算術交配法 (Arithmetic Crossover)。
- 突變法則：以二進位制為編碼方式的突變為參考 2.1 節。突變機率 $P_m = 0.03$ ；以十進位為編碼方式的突變方式為均勻突變法 (Uniform Mutation)。
- 演化代數：在本範例中，我們取最大代數為 1000。

- g. 收斂條件：在本範例中，當與最佳值的誤差小於 0.001 時且連續 20 代皆如此，則視為收斂。
- h. 測試次數：30 次。
- i. 參數設定：在本範例中，菁英化單體數目取 0%、50% 和 100% 來比較；單體演化次數為 5 次、10 次和 15 次。

(3) 實驗結果

圖 4. 88 至圖 4. 94 以及表 4. 26 為本研究取菁英化數目為兩個單體的實驗結果。根據變異數分析(ANOVA)的結果(請參考附錄一表 A1. 34 及表 A1. 36)，方法的不同對收斂代數有顯著的影響且方法的不同對收斂時間有顯著的影響。

在 Y-L 方法中，其『收斂百分比』顯然沒有比 HMSGGA-CS 來的穩定，且『平均收斂代數』及『平均收斂時間』也比我們來的多。在實數型(Real Value)的 HMSGGA-CS 中單體的數目不同對於『平均收斂代數』(p-value=0.699> ，請參考附錄一表 A1. 35)及『平均收斂時間』(p-value=0.845> ，請參考附錄一表 A1. 37)並沒有顯著的影響。

在本範例中，本研究將以實驗設計來驗證參數對實驗結果的影響。表 4. 27 我們的測試組合表與數據。

表 4. 26 測試結果表(算例十二)

Method			Convergence Percentage	Average* Generation	Standard Deviation (Generation)	Mean* Time(sec.) (Convergence)	Mean Value	Accuracy
HMSGGA -CS	Binary	Nds=5	46.67%	49.1429	24.7972	7.8984	-7.6621	1.7611
		Nds=10	77%	28.134	4.3829	2.4602	-8.73917	0.9995
		Nds=15	66.67%	23.2	28.5687	1.2472	-7.99644	1.5247
	Real Value	Nds=5	80%	29.7917	3.4133	0.5666	-9.1585	0.7030
		Nds=10	86.67%	27.2692	3.9044	0.7834	-9.3331	0.5795
		Nds=15	86.67%	25.2308	3.7769	0.8647	-9.3928	0.5373
Y-L (Binary)			36.67%	80	20.8038	7.3643	-5.8848	3.0179
Y-L (Real Value)			16.67%	488.4	179.3929	22.2082	-5.2971	3.4334
GA (Binary)			0%	1000	0	135.895	-5.5208	3.2752
GA (Real Value)			0%	1000	0	46.639	-5.24	3.4738

*代表只計算有收斂的數值。當『收斂百分比』為 0 時，『平均收斂時間』取演算 1000 代的平均時間。

表 4.27 參數實驗設計組合表(算例十二)

變數 組別	資料型態	菁英化單體數目	單體演化次數	演化代數	時間(秒)
第 1 組	Binary	0%	Nds=5	241	41.69
第 2 組				78	10.976
第 3 組				333	62.69
第 4 組				29	1.692
第 5 組				510	101.125
第 6 組				22	4.847
第 7 組				50	8.134
第 8 組				81	11.431
第 9 組				103	12.974
第 10 組				135	14.317
第 11 組			Nds=10	26	1.352
第 12 組				26	1.332
第 13 組				74	12.618
第 14 組				43	5.287
第 15 組				26	1.372
第 16 組				112	21.662
第 17 組				72	12.097
第 18 組				27	1.572
第 19 組				28	1.863
第 20 組				27	1.602
第 21 組			Nds=15	88	19.869
第 22 組				234	62.289
第 23 組				24	1.643
第 24 組				26	2.473
第 25 組				29	2.203
第 26 組				28	1.102
第 27 組				24	1.372
第 28 組				25	1.091
第 29 組				25	1.402
第 30 組				34	3.786
第 31 組	50%	Nds=5	58	9.864	
第 32 組			28	2.043	
第 33 組			29	2.313	
第 34 組			30	2.574	
第 35 組			31	2.844	
第 36 組			34	3.735	
第 37 組			30	5.478	
第 38 組			41	11.716	

第 39 組				35	8.513
第 40 組				36	9.143
第 41 組			Nds=10	29	3.165
第 42 組				28	2.814
第 43 組				25	1.853
第 44 組				25	1.582
第 45 組				31	3.385
第 46 組				24	1.222
第 47 組				28	2.453
第 48 組				27	2.123
第 49 組				25	1.472
第 50 組				30	2.965
第 51 組			Nds=15	24	1.472
第 52 組				37	5.698
第 53 組				38	5.999
第 54 組				27	2.313
第 55 組				28	2.724
第 56 組				31	3.775
第 57 組				25	1.683
第 58 組				24	1.352
第 59 組				41	7.25
第 60 組				24	1.382
第 61 組		ALL	Nds=5	34	3.555
第 62 組				36	4.036
第 63 組				64	11.386
第 64 組				62	10.876
第 65 組				31	2.804
第 66 組				29	2.294
第 67 組				39	4.897
第 68 組				31	2.834
第 69 組				30	4.726
第 70 組				29	4.317
第 71 組			Nds=10	29	2.163
第 72 組				27	4.377
第 73 組				35	1.742
第 74 組				26	1.162
第 75 組				24	1.722
第 76 組				26	2.314
第 77 組				28	7.901
第 78 組				46	9.694
第 79 組				52	7.611

第 80 組				45	1.802
第 81 組			Nds=15	26	2.023
第 82 組				28	2.704
第 83 組				24	1.342
第 84 組				26	2.063
第 85 組				32	4.106
第 86 組				26	2.063
第 87 組				25	1.732
第 88 組				25	1.723
第 89 組				24	1.362
第 90 組				25	1.692
第 91 組	Real Value		Nds=5	31	0.611
第 92 組				27	0.39
第 93 組				32	0.65
第 94 組				26	0.341
第 95 組				33	0.701
第 96 組				28	0.441
第 97 組				29	0.48
第 98 組				27	0.391
第 99 組				28	0.42
第 100 組				26	0.331
第 101 組			Nds=10	25	0.501
第 102 組				25	0.5
第 103 組				26	0.581
第 104 組				29	0.852
第 105 組				26	0.58
第 106 組				34	1.332
第 107 組				25	0.501
第 108 組				30	0.951
第 109 組				27	0.661
第 110 組				24	0.421
第 111 組			Nds=15	24	0.57
第 112 組				23	0.421
第 113 組				23	0.441
第 114 組				23	0.42
第 115 組				25	0.711
第 116 組				24	0.561
第 117 組				44	3.454
第 118 組				26	0.862
第 119 組				24	0.581
第 120 組				23	0.451

第 121 組				31	0.661
第 122 組				26	0.35
第 123 組				29	0.511
第 124 組				27	0.4
第 125 組			Nds=5	26	0.331
第 126 組				33	0.751
第 127 組				27	0.401
第 128 組				27	0.431
第 129 組				32	0.671
第 130 組				21	0.06
第 131 組				24	0.421
第 132 組				23	0.3
第 133 組				25	0.491
第 134 組				29	0.891
第 135 組			Nds=10	35	1.492
第 136 組		50%		27	0.681
第 137 組				24	0.401
第 138 組				24	0.411
第 139 組				24	0.411
第 140 組				27	0.721
第 141 組				24	0.601
第 142 組				24	0.711
第 143 組				32	1.883
第 144 組				32	1.972
第 145 組			Nds=15	23	0.441
第 146 組				32	1.763
第 147 組				26	0.871
第 148 組				26	0.882
第 149 組				31	1.582
第 150 組				23	0.431
第 151 組		100%		30	0.571
第 152 組				31	0.651
第 153 組				29	0.531
第 154 組				28	0.47
第 155 組			Nds=5	25	0.301
第 156 組				33	0.761
第 157 組				28	0.471
第 158 組				36	0.992
第 159 組				28	0.491
第 160 組				26	0.36
第 161 組			Nds=10	26	0.631

第 162 組				24	0.401
第 163 組				27	0.701
第 164 組				24	0.401
第 165 組				25	0.5
第 166 組				23	0.311
第 167 組				27	0.701
第 168 組				27	0.711
第 169 組				34	1.453
第 170 組				25	0.53
第 171 組			Nds=15	24	0.611
第 172 組				24	0.591
第 173 組				27	1.012
第 174 組				23	0.43
第 175 組				26	0.861
第 176 組				24	0.571
第 177 組				24	0.561
第 178 組				23	0.421
第 179 組				23	0.42
第 180 組				24	0.571

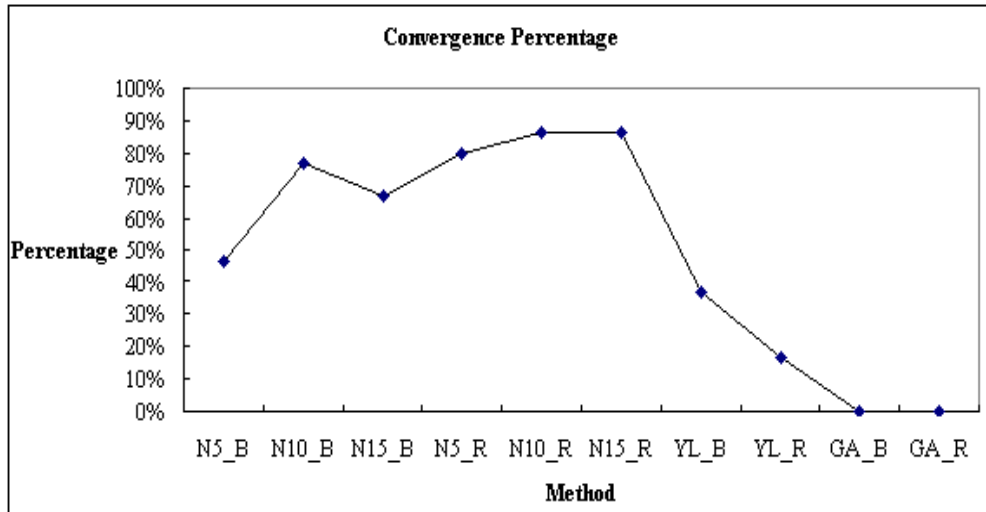


圖 4. 88 收斂率比較圖(算例十二)

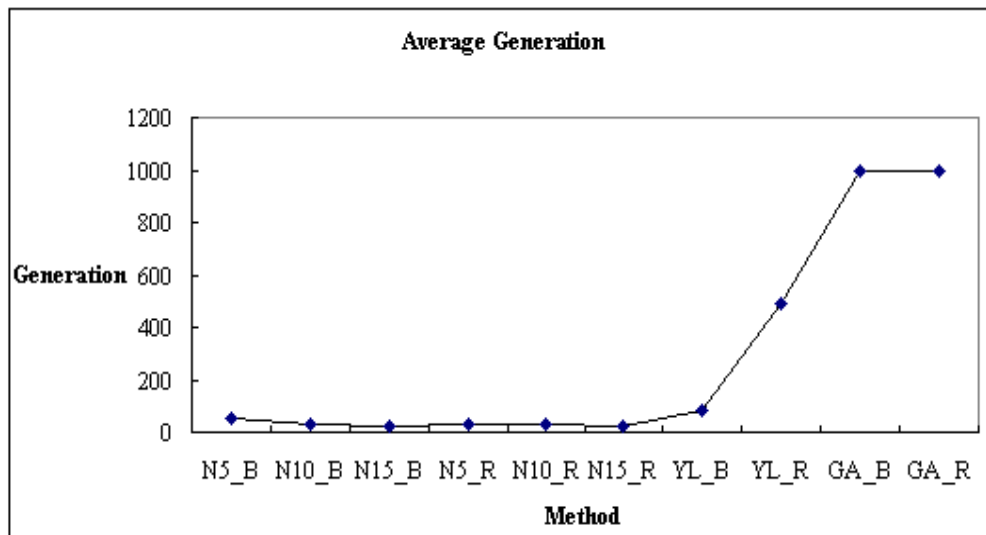


圖 4. 89 平均收斂代數表示圖(算例十二)

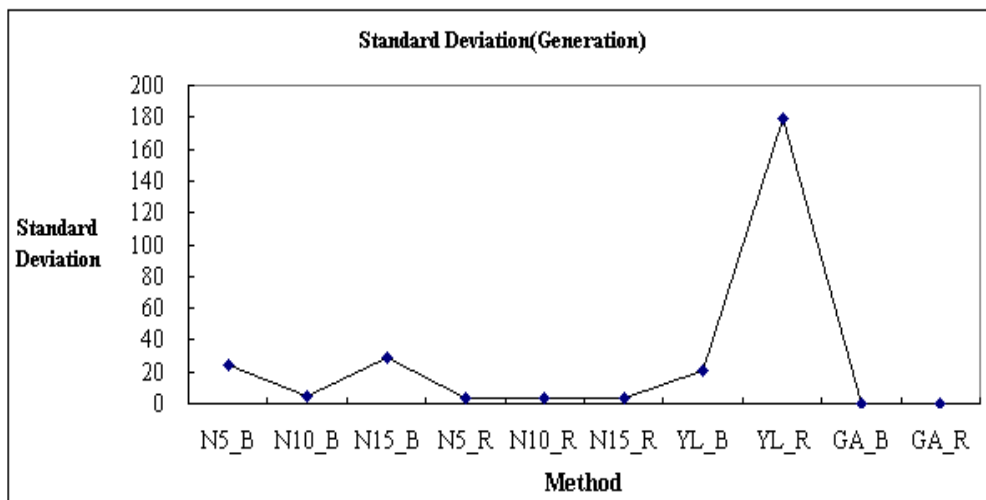


圖 4. 90 平均收斂代數之標準差比較圖(算例十二)

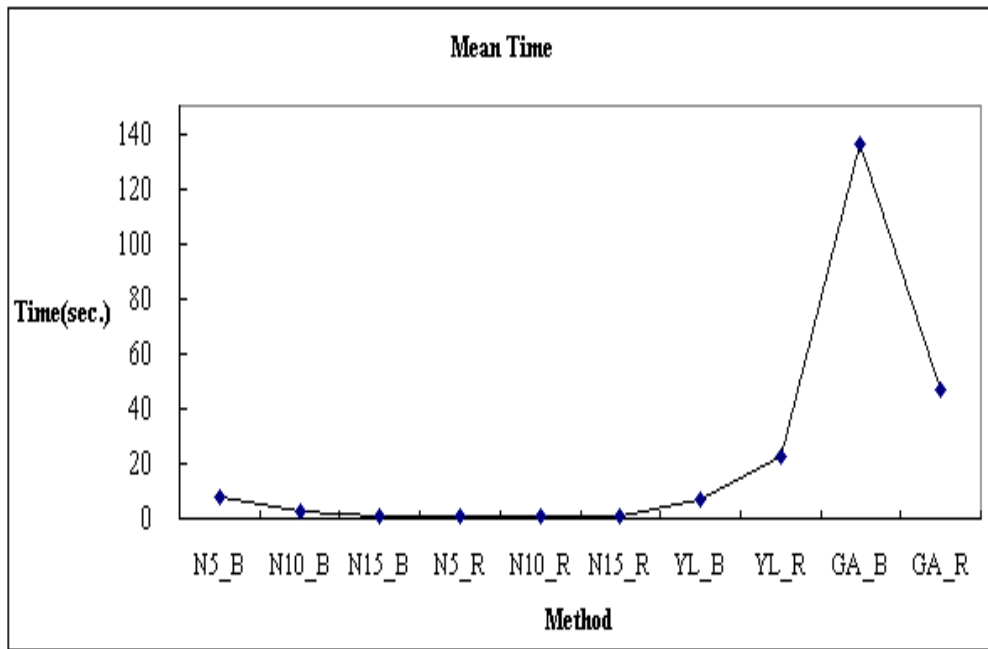


圖 4.91 達收斂時之平均運算時間比較圖(算例十二)

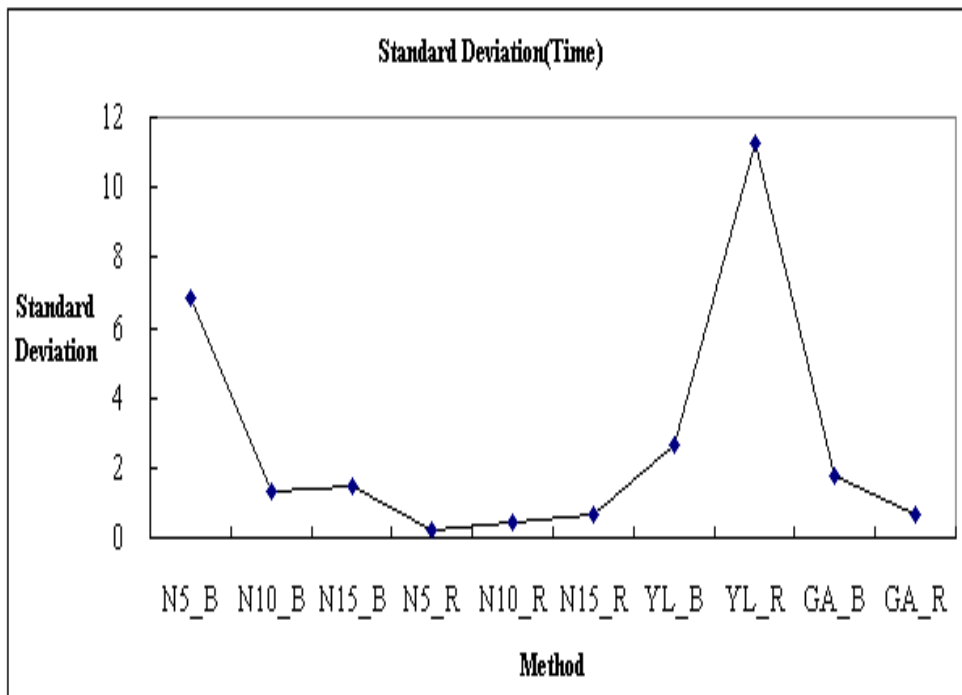


圖 4.92 平均運算時間之標準差比較圖(算例十二)

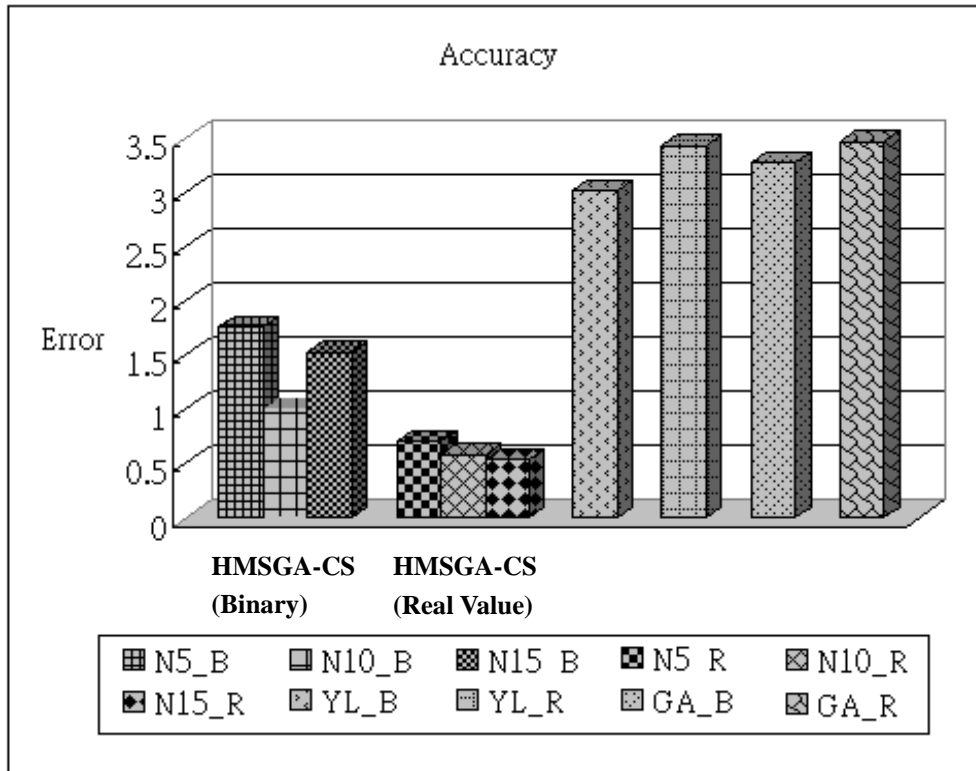


圖 4.93 與理想值之誤差比較圖(算例十二)

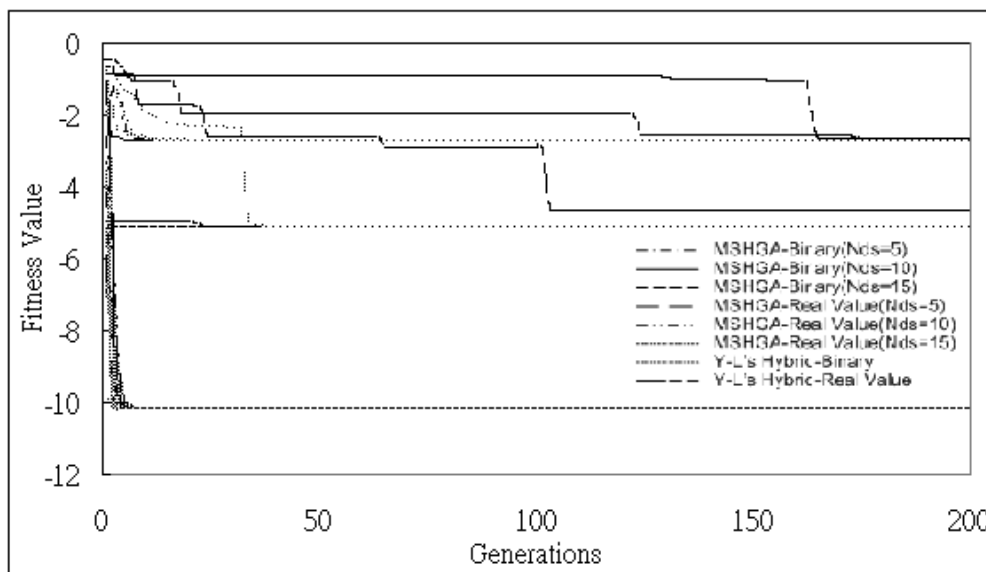


圖 4.94 代數收斂表示圖(算例十二)

(4) 實驗數據分析

1. 收斂代數分析及探討

表 4.28 為針對各參數對收斂代數的數據分析結果。在分析方

面，我們取顯著水準 $\alpha=0.05$ 來檢定是否有顯著關係。

表 4. 28 收斂代數之變異數分析表(算例十二)

變異來源	自由 度	均方	誤差均方	F 值	P-value
資料型態	1	23529.8	1706.789	13.78601	0.000281
菁英化數目	2	14905.14	1706.789	8.732855	0.00025
單體演化次數	2	9541.272	1706.789	5.590189	0.004491
資料型態 vs 菁英化數目	2	14523.82	1706.789	8.509439	0.000306
資料型態 vs 單體演化次數	2	7478.317	1706.789	4.381512	0.014022
菁英化數目 vs 單體演化次數	4	5684.172	1706.789	3.330331	0.011876
資料型態 vs 菁英化數目 vs 單體演化次數	4	5753.133	1706.789	3.370735	0.011129
誤差值	162				

首先，先就單因子來探討對收斂代數的影響：

a. 資料型態的不同對收斂代數有顯著的影響。

在表 4. 29 中，對於本範例而言，實數型的架構比二進位制的收斂代數來的好。

b. 菁英化數目的不同對收斂代數有顯著的影響。

在表 4. 29 中，50%的菁英化策略其結果比 0%的來的好，不過注意的是並非比例越高，收斂代數越少，50%和 100%的差異並不大。

c. 單體演化次數的不同對收斂代數有顯著的影響。

在表 4. 29 中，單體演化 10 代的結果比 5 代來的好，不過當單體演化 15 待時，其結果與 10 代並無太大差異。

表 4. 29 單因子平均代數比較表(算例十二)

參數	資料型態		菁英化比例			單體演化代數		
	Binary	Real Value	0%	50%	100%	5	10	15
平均代數	49.86	26.99	51.62	28.95	29.7	52.98	31.1	31.18

接下來檢視各因子之間是否存在交互作用：

a. 資料型態與菁英化數目之間的交互關係顯著。

在表 4. 30 中，實數型的組合比二進位制的組合結果來的好，但收斂代數並不會因為菁英化比例提高而造成較少的收斂代數，反而是差不多的形式；而二進位制的資料型態則是隨著菁英化比例增加，而有較好的結果。

b. 資料型態與單體演化次數之間的交互關係顯著。

在表 4. 30 中，實數型的資料型態搭配的單體演化次數其結果比二進位制的資料型態來的好，不過其間的差異並不大；而二進位制的資料型態在 10 代與 15 代的結果便差異不大。

c. 菁英化數目與單體演化次數之間的交互關係顯著。

在表 4. 31 中，菁英化的比例越高與單體演化的次數越高並不會絕對造成收斂代數的減少，組合 5 的結果就比組合 8 來的好。

d. 資料型態、菁英化數目與單體演化次數之間的交互關係顯著。

在表 4. 32 中，我們可以看出再組合 1 到組合 9 有顯著差異外，其餘的組合就比較不受其參數所影響。

表 4. 30 雙因子平均代數比較表(算例十二)

平均代數 \ 因子	資料型態 vs 菁英化數目		資料型態 vs 單體演化次數	
	組合 1	B vs 0%	86	B vs 5
組合 2	B vs 50%	30.76667	B vs 10	35.7
組合 3	B vs 100%	32.8	B vs 15	36.567
組合 4	R vs 0%	27.23333	R vs 5	28.667
組合 5	R vs 50%	27.13333	R vs 10	26.5
組合 6	R vs 100%	26.6	R vs 15	25.8

表 4. 31 雙因子平均代數比較表(續)

平均代數 \ 因子	菁英化數目 vs 單體演化次數	
	組合 1	0% vs 5
組合 2	0% vs 10	36.6
組合 3	0% vs 15	39.8
組合 4	50% vs 5	31.55
組合 5	50% vs 10	26.7
組合 6	50% vs 15	28.6
組合 7	100% vs 5	33.95
組合 8	100% vs 10	30
組合 9	100% vs 15	25.15

表 4. 32 三因子平均代數比較表(算例十二)

平均時間 \ 因子	資料型態 vs 菁英化數目 vs 單體演化次數	
組合 1	Binary vs 0% vs Nds=5	158.2
組合 2	Binary vs 0% vs Nds=10	46.1
組合 3	Binary vs 0% vs Nds=15	53.7
組合 4	Binary vs 50% vs Nds=5	35.2
組合 5	Binary vs 50% vs Nds=10	27.2
組合 6	Binary vs 50% vs Nds=15	29.9
組合 7	Binary vs All vs Nds=5	38.5
組合 8	Binary vs All vs Nds=10	33.8
組合 9	Binary vs All vs Nds=15	26.1
組合 10	Real-Val vs 0% vs Nds=5	28.7
組合 11	Real-Val vs 0% vs Nds=10	27.1
組合 12	Real-Val vs 0% vs Nds=15	25.9
組合 13	Real-Val vs 50% vs Nds=5	27.9
組合 14	Real-Val vs 50% vs Nds=10	26.2
組合 15	Real-Val vs 50% vs Nds=15	27.3
組合 16	Real-Val vs All vs Nds=5	29.4
組合 17	Real-Val vs All vs Nds=10	26.2
組合 18	Real-Val vs All vs Nds=15	24.2

4.1.13算例(十三) : SQRN7 function

(1)目標函數

目標函數如(4-13)所示，此函數目的在求此函數的最小值。

$$\text{Min. } F(X) = - \sum_{j=1}^7 \frac{1}{\sum_{i=1}^4 (x_i - a_{ij})^2 + c_j} \quad (4-13)$$

where $X_i \in [0,10]$

此函數之理想(Ideal)最佳值為-10.4028。其餘參數值請參照表 4. 25。

(2)測試環境與變數設定

此研究之相關變數如下：

- a. 族群大小：本範例母體大小為 40。
- b. 單一變數之基因數：單一變數之染色體長度為 20，精確度為 10^5 。
- c. 複製法則：輪盤法(Roulette)。
- d. 交配法則：以二進位制為編碼方式的交配方式為雙點交配。交配機率 $P_c=0.4$ ；以十進位為編碼方式的交配方式為算術交配法 (Arithmetic Crossover)。
- e. 突變法則：以二進位制為編碼方式的突變為參考 2.1 節。突變機率 $P_m=0.03$ ；以十進位為編碼方式的突變方式為均勻突變法 (Uniform Mutation)。
- f. 演化代數：在本範例中，我們取最大代數為 1000。
- g. 收斂條件：在本範例中，當與最佳值的誤差小於 0.001 時且連續 20 代皆如此，則視為收斂。
- h. 測試次數：30 次。
- i. 參數設定：在本範例中，菁英化單體數目為兩個；單體演化次數為 5 次、10 次和 15 次。

(3)實驗結果與比較

圖 4. 95 至圖 4. 101 以及表 4. 33 為本研究實驗的結果。根據變異數分析(ANOVA)的結果(請參考附錄一表 A1. 38 及表 A1. 40)，方法的不同對收斂代數有顯著的影響且方法的不同對收斂時間有顯著的影響。

在表 4. 33 中，以『收斂百分比』來說，Y-L 的方法及 GA 達成目標的比例很低。以『平均收斂代數』來看，HMSGGA-CS 的組合所測試出來的結果皆比 Y-L 及 GA 的方法好。在 HMSGGA-CS 中不同的組合對於『平均收斂代數』(p-value=0.063>，請參考附錄一表 A1. 39)及『平均收斂時間』(p-value=0.099>，請參考附錄一表 A1. 41)並沒有顯著的影響。

表 4.33 測試結果表(算例十三)

Method		Convergence Percentage	Average* Generation	Standard Deviation (Generation)	Mean* Time(sec.) (Convergence)	Mean Value	Accuracy	
HMSGA -CS	Binary	Nds=5	100.00%	38.3333	11.3242	4.9985	-10.4017	0.0004
		Nds=10	100.00%	28.1333	5.2626	2.4722	-10.4020	0.0002
		Nds=15	100.00%	25.0000	3.1168	1.7772	-10.4021	0.0002
	Real Value	Nds=5	86.67%	28.1538	2.5770	0.5447	-9.3818	0.7216
		Nds=10	100.00%	25.4000	1.6818	0.6202	-10.4021	0.0002
		Nds=15	100.00%	24.6667	3.2660	0.8058	-10.4021	0.0001
Y-L (Binary)		36.67%	57.9091	12.9031	4.7066	-5.7297	3.3040	
Y-L (Real Value)		13.33%	685.25	249.3236	36.377	-4.9381	3.8638	
GA (Binary)		0%	1000	0	148.548	-5.614	3.3858	
GA (Real Value)		0%	1000	0	53.998	-5.431	3.5150	

*代表只計算有收斂的數值。當「收斂百分比」為 0 時，「平均收斂時間」取演算 1000 代的平均時間。

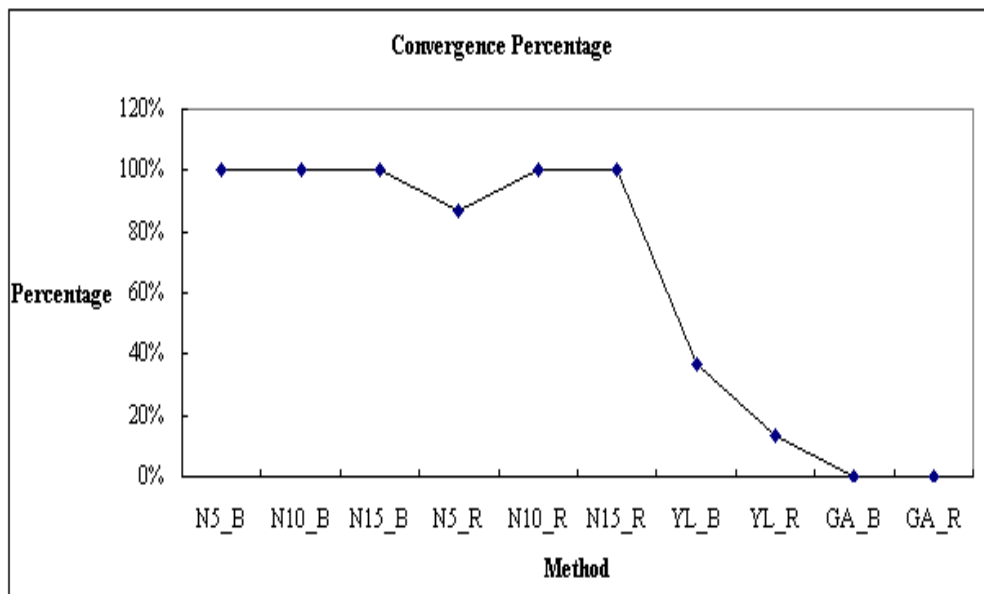


圖 4.95 收斂率比較圖(算例十三)

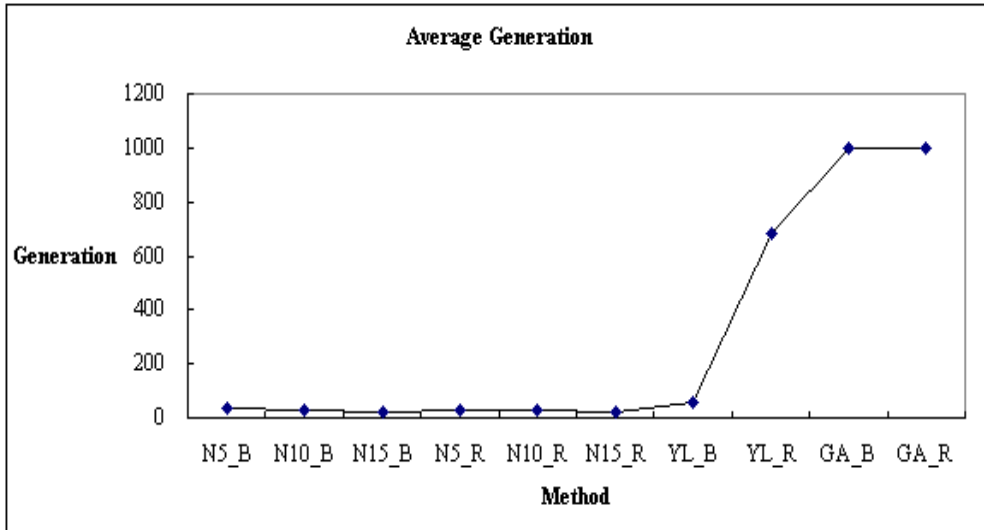


圖 4.96 平均收斂代數比較圖(算例十三)

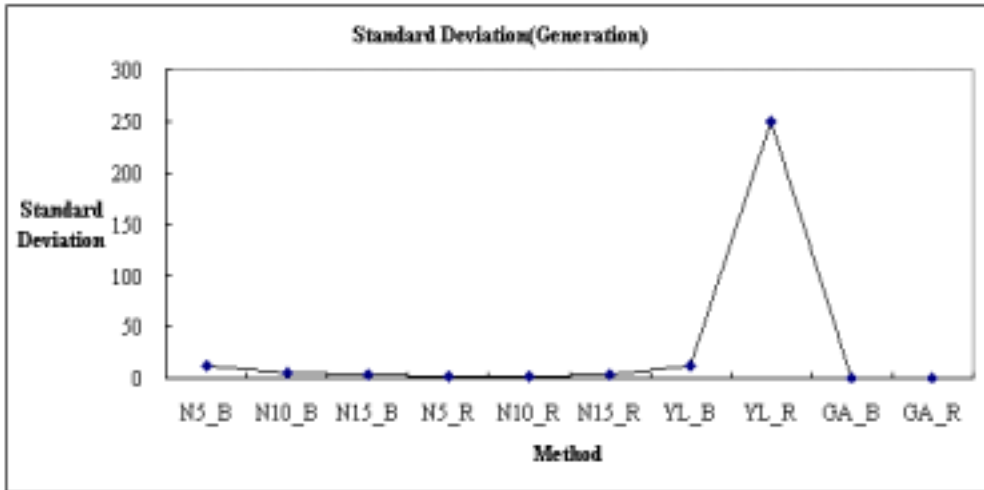


圖 4.97 平均收斂代數之標準差比較圖(算例十三)

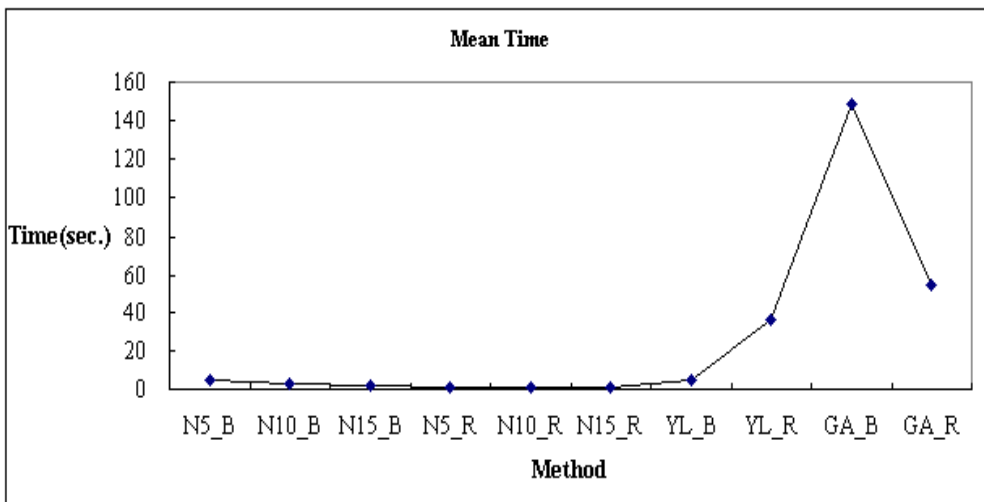


圖 4.98 達收斂時之平均運算時間比較圖(算例十三)

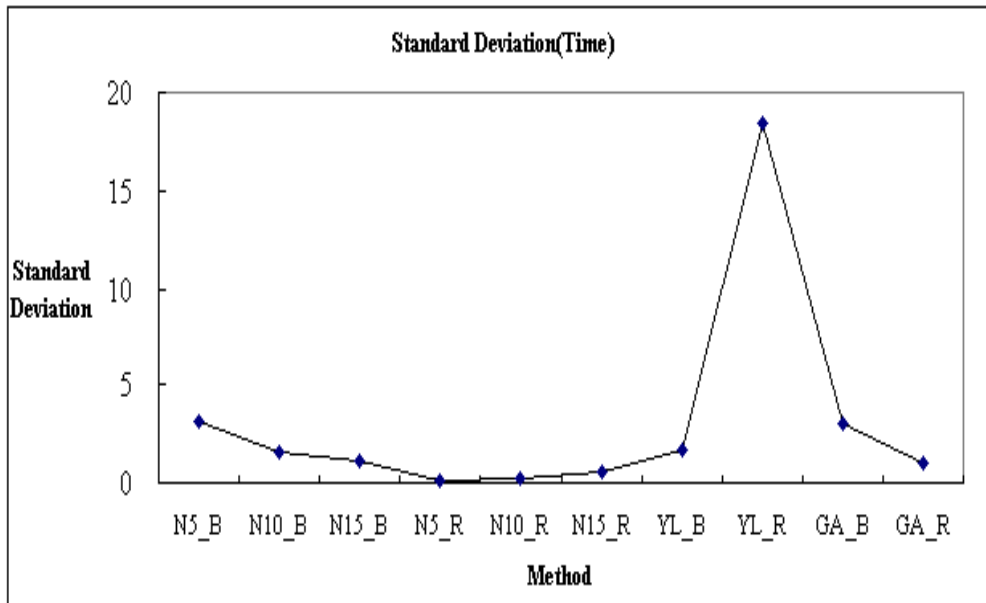


圖 4.99 平均運算時間之標準差比較圖(算例十三)

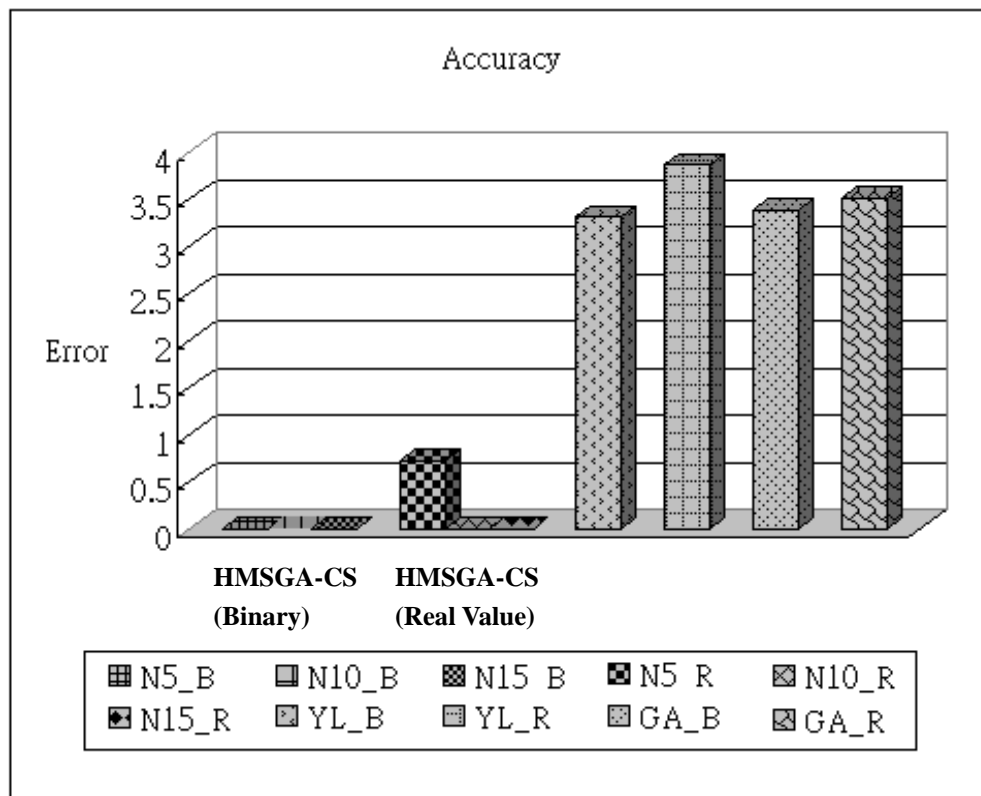


圖 4.100 與理想值之誤差比較圖(算例十三)

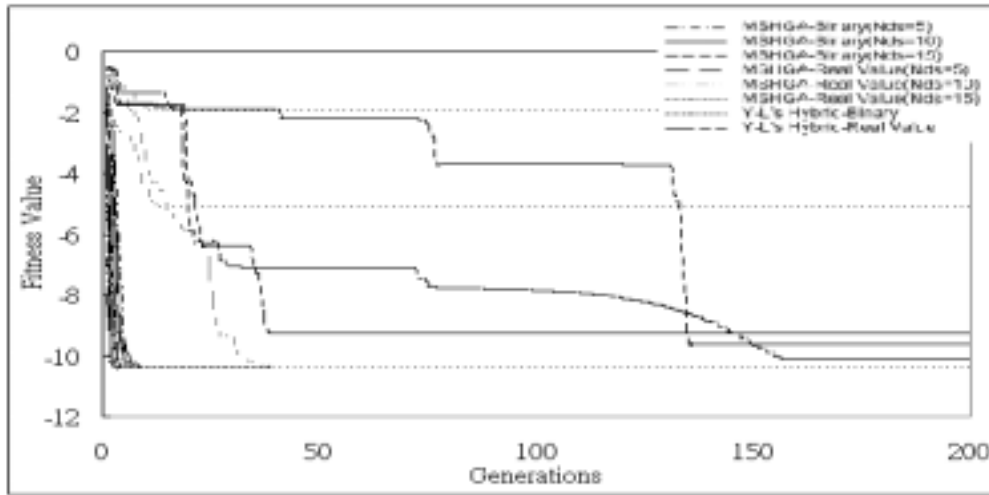


圖 4.101 代數收斂表示圖(算例十三)

4.1.14 算例(十四)：SQRN10 function

(1) 目標函數

目標函數如(4-14)所示，此函數目的在求此函數的最小值。

$$\text{Min. } F(X) = -\sum_{j=1}^{10} \frac{1}{\sum_{i=1}^4 (x_i - a_{ij})^2 + c_j} \quad (4-14)$$

where $X_i \in [0,10]$

此函數之理想(Ideal)最佳值為-10.5359。其餘參數值請參照表 4.25。

(2) 測試環境與變數設定

此研究之相關變數如下：

- 族群大小：本範例母體大小為 40。
- 單一變數之基因數：單一變數之染色體長度為 20，精確度為 10^5 。
- 複製法則：輪盤法(Roulette)。
- 交配法則：以二進位制為編碼方式的交配方式為雙點交配。交配機率 $P_c = 0.4$ ；以十進位為編碼方式的交配方式為算術交配法(Arithmetic Crossover)。
- 突變法則：以二進位制為編碼方式的突變為參考 2.1 節。突變機率 $P_m = 0.03$ ；以十進位為編碼方式的突變方式為均勻突變法(Uniform Mutation)。

- f. 演化代數：在本範例中，我們取最大代數為 1000。
- g. 收斂條件：在本範例中，當與最佳值的誤差小於 0.001 時且連續 20 代皆如此，則視為收斂。
- h. 測試次數：30 次。
- i. 參數設定：在本範例中，菁英化單體數目為兩個；單體演化次數為 5 次、10 次和 15 次。

(3) 實驗結果與比較

圖 4. 102 至圖 4. 108 以及表 4. 34 為本研究實驗的結果。根據變異數分析(ANOVA)的結果(請參考附錄一表 A1. 42 及表 A1. 43), 方法的不同對收斂代數有顯著的影響且方法的不同對收斂時間有顯著的影響。

在表 4. 34 中，Y-L 的方法及 GA 在『收斂百分比』的項目中比例皆不高甚至為 0，顯示其穩定性不高。以『平均收斂代數』來說，HMSGGA-CS 的所有組合皆比 Y-L 及 GA 的方法來的佳，且二進位制(Binary)與 Nds=10 和 Nds=15 的組合以及十進位制(Real Value)與 Nds=10 和 Nds=15 的組合經變異數分析(ANOVA)後其結果並無顯著差異($p\text{-value}=0.1 > \alpha$)。以『平均收斂時間』來說，HMSGGA-CS 中的所有組合並無顯著差異($p\text{-value}=0.0915 > \alpha$)。

表 4. 34 測試結果表(算例十四)

Method		Convergence Percentage	Average* Generation	Standard Deviation (Generation)	Mean* Time(sec.) (Convergence)	Mean Value	Accuracy	
HMSGGA-CS	Binary	Nds=5	100.00%	31.6667	4.0649	3.0108	-10.5353	0.0003
		Nds=10	100.00%	27.4000	5.3157	2.4710	-10.5355	0.0002
		Nds=15	100.00%	24.4667	2.3258	1.7545	-10.5356	0.0002
	Real Value	Nds=5	86.67%	32.9231	8.5485	1.0862	-9.6671	0.6143
		Nds=10	100.00%	25.8667	2.6957	0.8346	-10.5355	0.0002
		Nds=15	100.00%	24.9333	2.2824	1.0528	-10.5355	0.0002
Y-L (Binary)		46.67%	59.7143	11.5187	5.3641	-6.5297	2.8327	
Y-L (Real Value)		16.67%	505.4	192.8595	27.8038	-5.5971	3.4922	
GA (Binary)		0%	1000	0	143.93	-5.501	3.5598	
GA (Real Value)		0%	1000	0	59.122	-5.125	3.8261	

*代表只計算有收斂的數值。當『收斂百分比』為 0 時，『平均收斂時間』取演算 1000 代的平均時間。

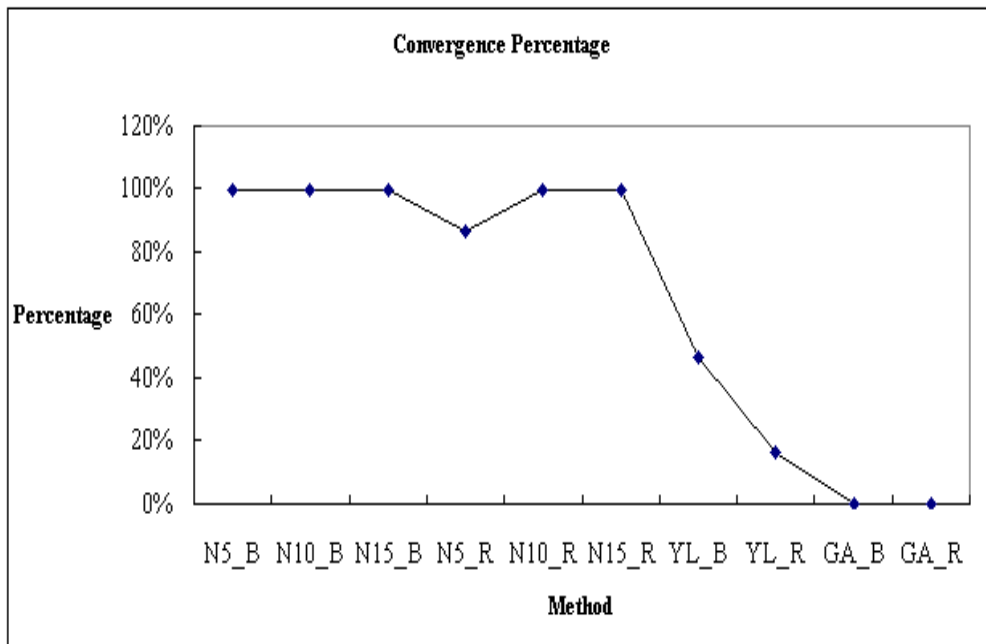


圖 4. 102 收斂率比較圖(算例十四)

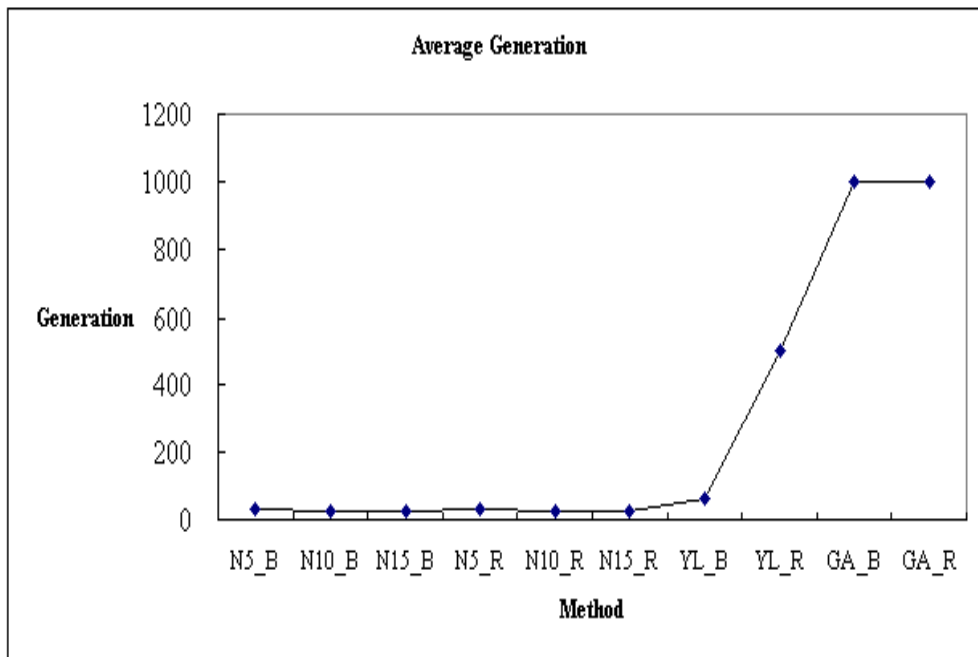


圖 4. 103 平均收斂代數比較圖(算例十四)

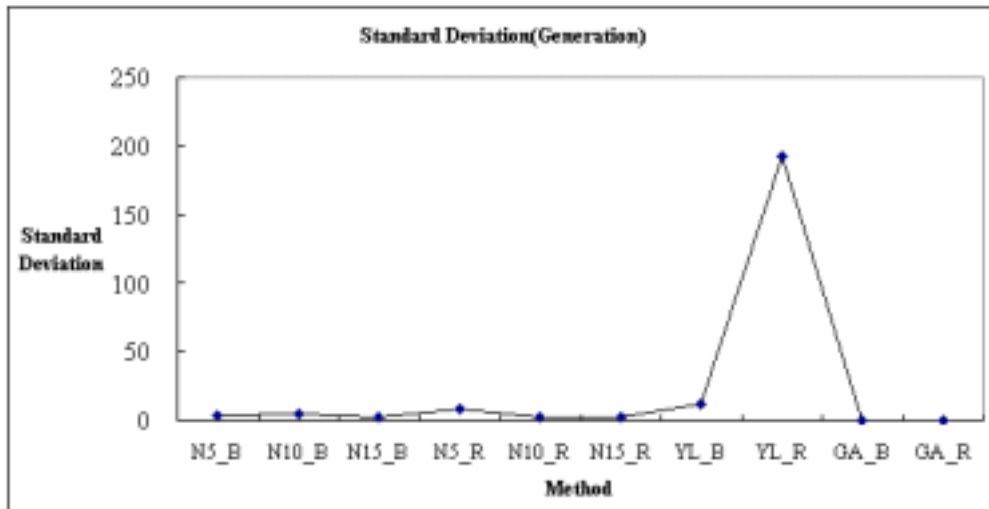


圖 4.104 平均收斂代數之標準差比較圖(算例十四)

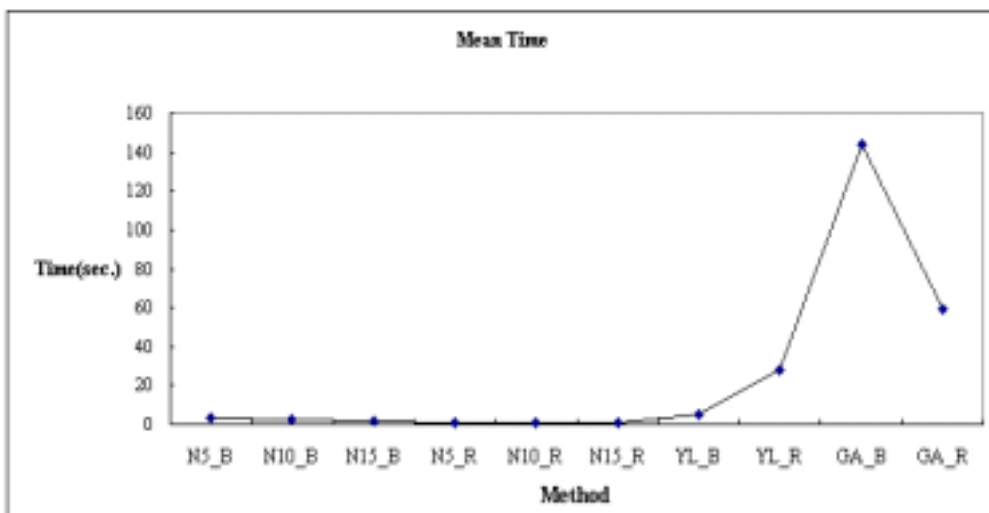


圖 4.105 達收斂時之平均運算時間比較圖(算例十四)

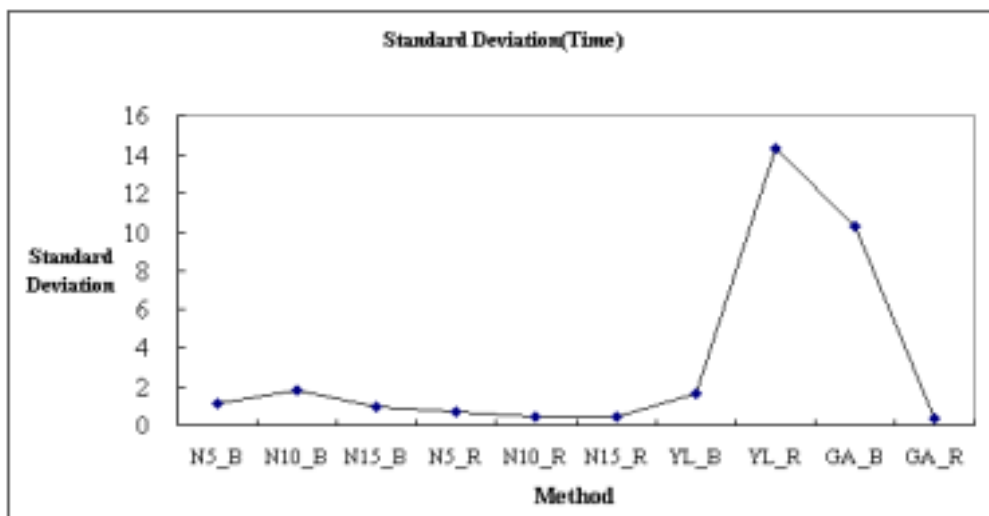


圖 4.106 平均運算時間之標準差比較圖(算例十四)

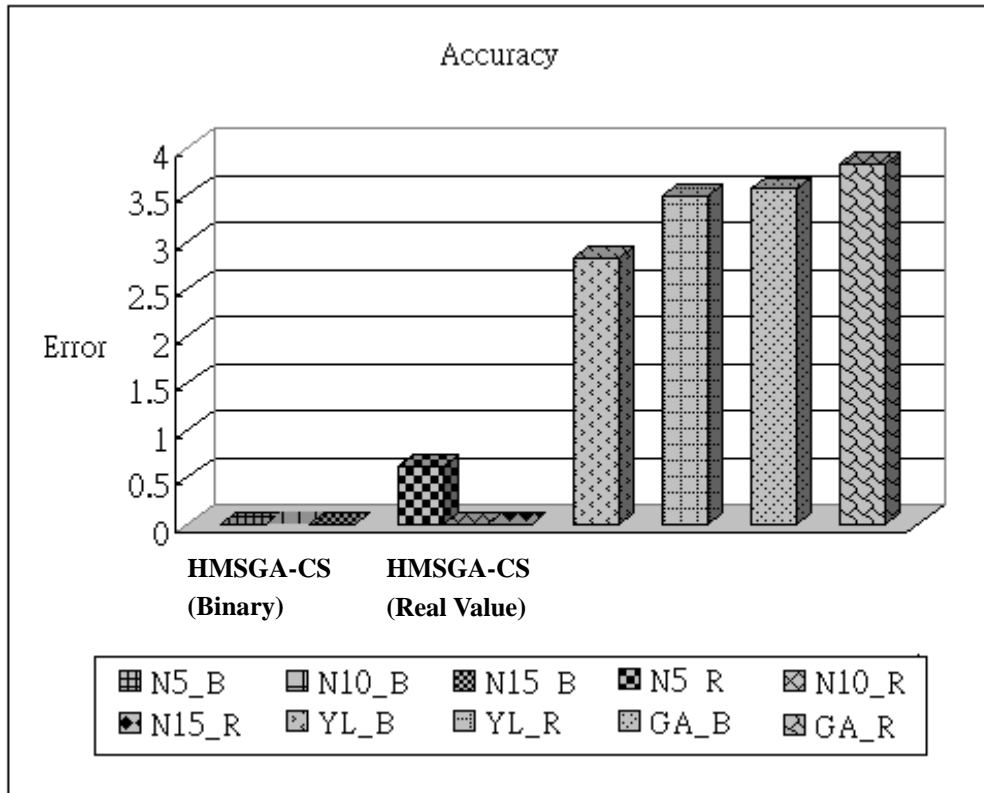


圖 4. 107 與理想值之誤差比較圖(算例十四)

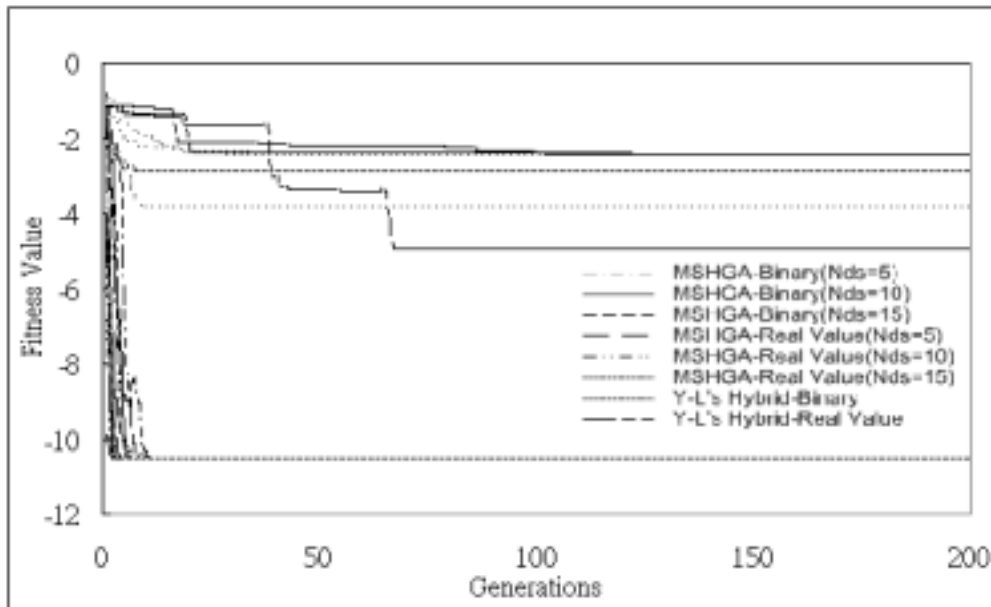


圖 4. 108 代數收斂表示圖(算例十四)

4.1.15算例(十五)：Colville function

(1)目標函數

目標函數如(4-15)所示，此函數目的在求此函數的最小值。

Min.

$$\begin{aligned} F(X) = & 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 \\ & + (1 - x_3)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) \\ & + 19.8(x_2 - 1)(x_4 - 1) \end{aligned} \quad (4-15)$$

where $X_i \in [0,10]$

此函數之理想(Ideal)最佳值為 0。

(2)測試環境與變數設定

此研究之相關變數如下：

- a. 族群大小：本範例母體大小為 40。
- b. 單一變數之基因數：單一變數之染色體長度為 21，精確度為 10^5 。
- c. 複製法則：輪盤法(Roulette)。
- d. 交配法則：以二進位制為編碼方式的交配方式為雙點交配。交配機率 $P_c = 0.4$ ；以十進位為編碼方式的交配方式為算術交配法(Arithmetic Crossover)。
- e. 突變法則：以二進位制為編碼方式的突變為參考 2.1 節。突變機率 $P_m = 0.03$ ；以十進位為編碼方式的突變方式為均勻突變法(Uniform Mutation)。
- f. 演化代數：在本範例中，我們取最大代數為 1000。
- g. 收斂條件：在本範例中，當與最佳值的誤差小於 0.001 時且連續 20 代皆如此，則視為收斂。
- h. 測試次數：30 次。
- i. 參數設定：在本範例中，菁英化單體數目為兩個；單體演化次數為 5 次、10 次和 15 次。

(3)實驗結果與比較

圖 4. 109 至圖 4. 115 以及表 4. 35 為本研究實驗的結果。根據變異數分析(ANOVA)的結果(請參考附錄一表 A1. 44 及表 A1. 45)，方法的不同對收斂代數有顯著的影響且方法的不同對收斂時間有顯著的影響。

在表 4. 35 中，以『收斂百分比』來說，Y-L 的方法及 GA 在 30

次的實驗中皆並未達到本研究所設定的目標，且由實數型(Real Value)的 Y-L 方法及 GA 所求出的最適值偏大，離理想值的目標還很遠，因此，若要達成所設定的收斂條件則需要更久的時間及代數。在『平均收斂代數』方面，經變異數分析(ANOVA)後，二進位制(Binary)的 HMSGGA-CS 在 Nds=10 和 Nds=15 的組合中，對其並無顯著影響(p-value=0.0615>)，而十進位制(Real Value)的 HMSGGA-CS 則在三種組合中對其皆無顯著的影響(p-value=0.0602>)。在『平均收斂時間』方面，HMSGGA-CS 中的各種組合對收斂時間皆無顯著的影響(p-value=0.345>)。

表 4.35 測試結果表(算例十五)

Method		Convergence Percentage	Average* Generation	Standard Deviation (Generation)	Mean* Time(sec.) (Convergence)	Mean Value	Accuracy	
HMSGGA-CS	Binary	Nds=5	100.00%	30.4	2.8234	2.5677	0.0006	0.0004
		Nds=10	100.00%	27.0667	1.9445	1.9421	0.0004	0.0003
		Nds=15	100.00%	25.3333	2.8452	1.6057	0.0003	0.0002
	Real Value	Nds=5	86.67%	82.4	128.9151	3.5077	0.0004	0.0003
		Nds=10	100.00%	37.3333	20.3634	1.2812	0.0004	0.0003
		Nds=15	100.00%	26.2	8.8415	0.6436	0.0002	0.0002
Y-L (Binary)		0%	1000	0	169.3715	0.8223	0.5815	
Y-L (Real Value)		0%	1000	0	73.5117	8.6097	6.0879	
GA (Binary)		0%	1000	0	158.1267	9.1376	6.4612	
GA (Real Value)		0%	1000	0	71.0048	113.295	80.113	

*代表只計算有收斂的數值。當『收斂百分比』為 0 時，『平均收斂時間』取演算 1000 代的平均時間。

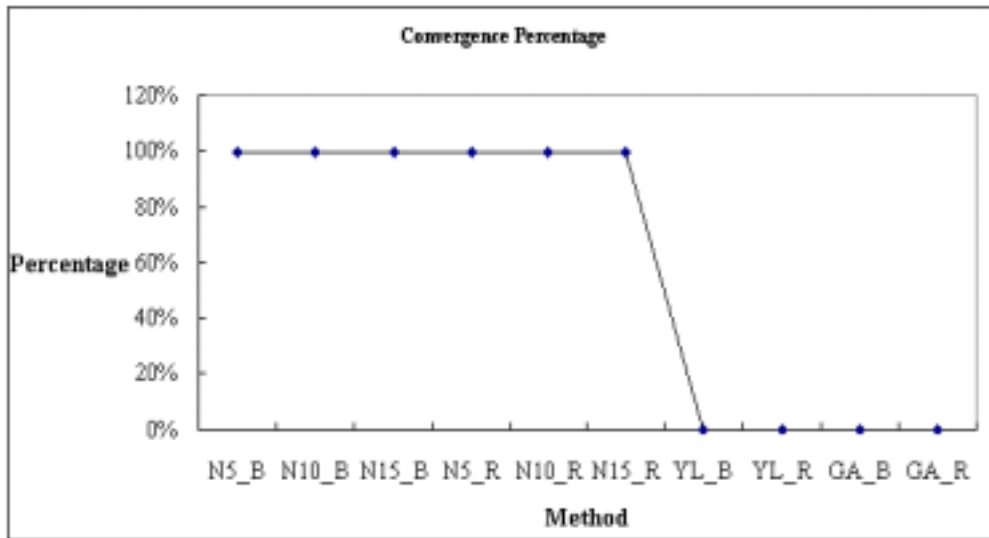


圖 4. 109 收斂率比較圖(算例十五)

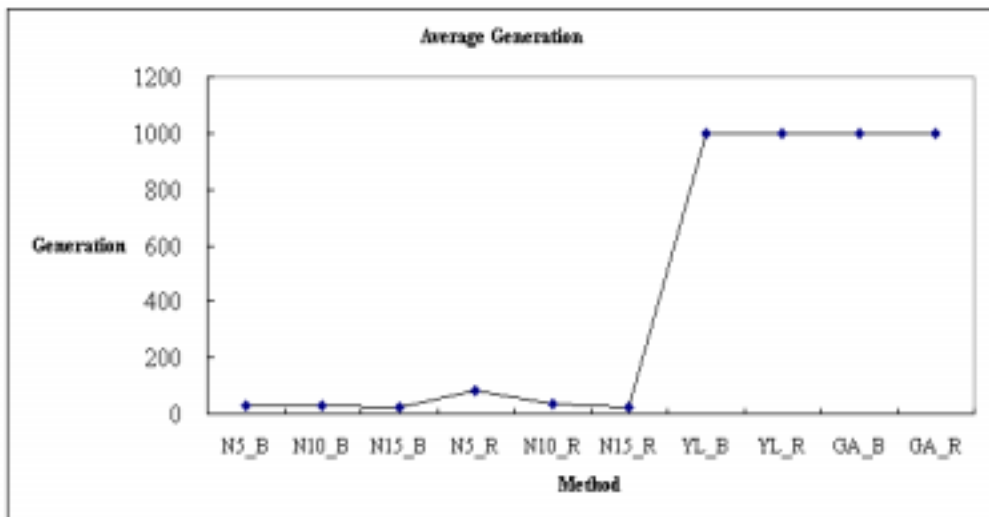


圖 4. 110 平均收斂代數比較圖(算例十五)

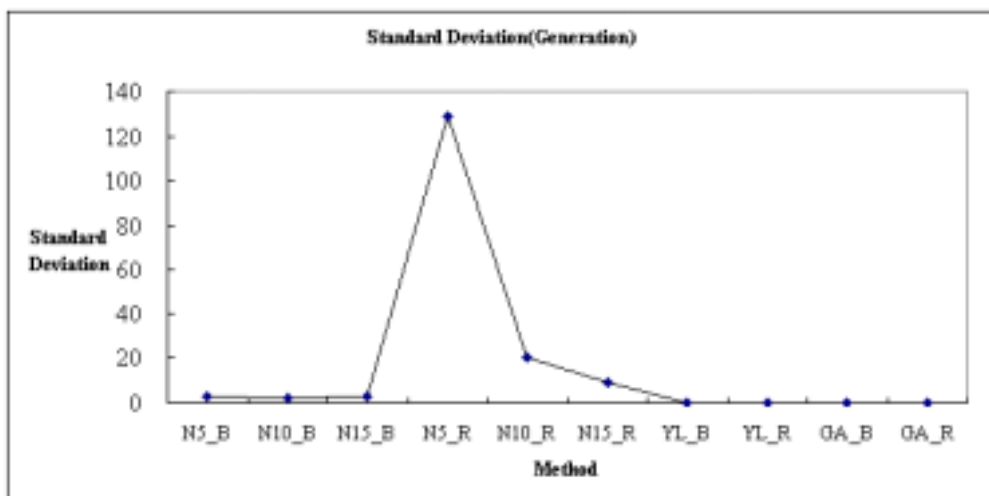


圖 4. 111 平均收斂代數之標準差比較圖(算例十五)

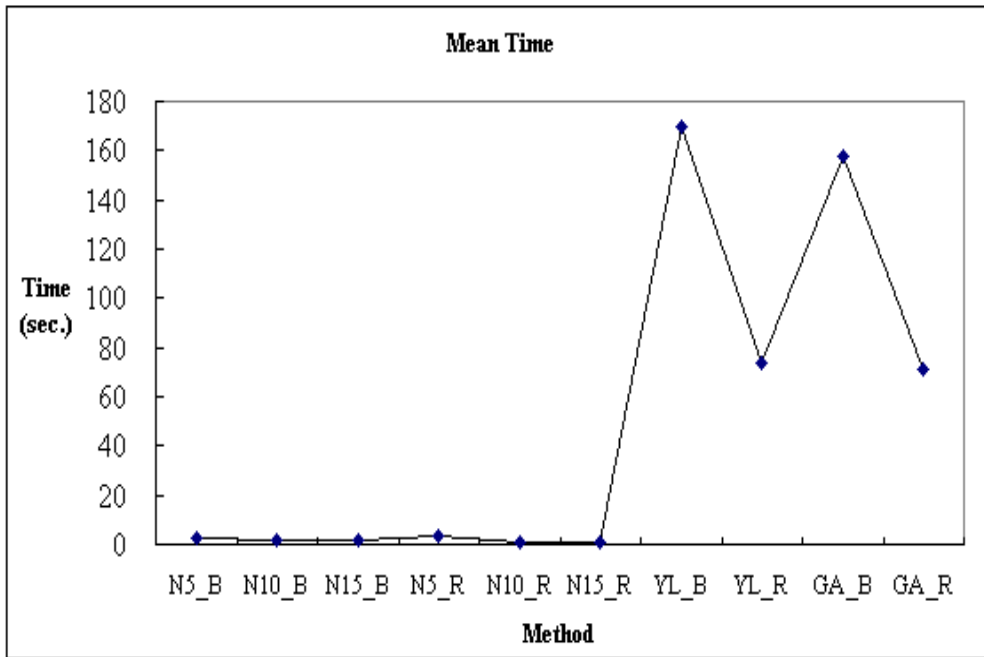


圖 4.112 達收斂時之平均運算時間比較圖(算例十五)

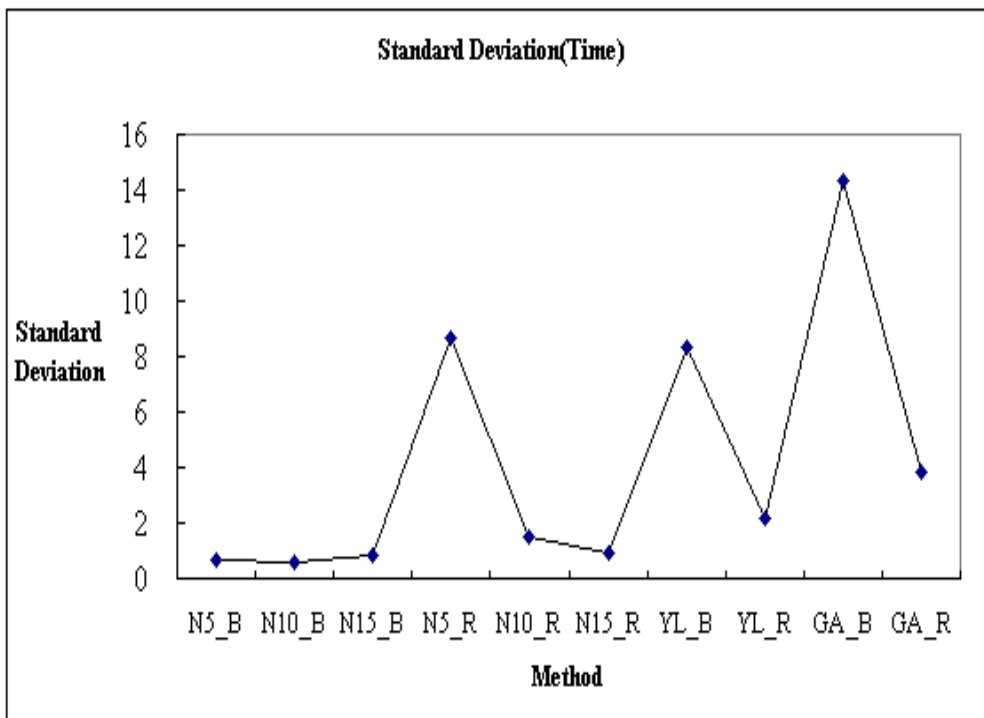


圖 4.113 平均運算時間之標準差比較圖(算例十五)

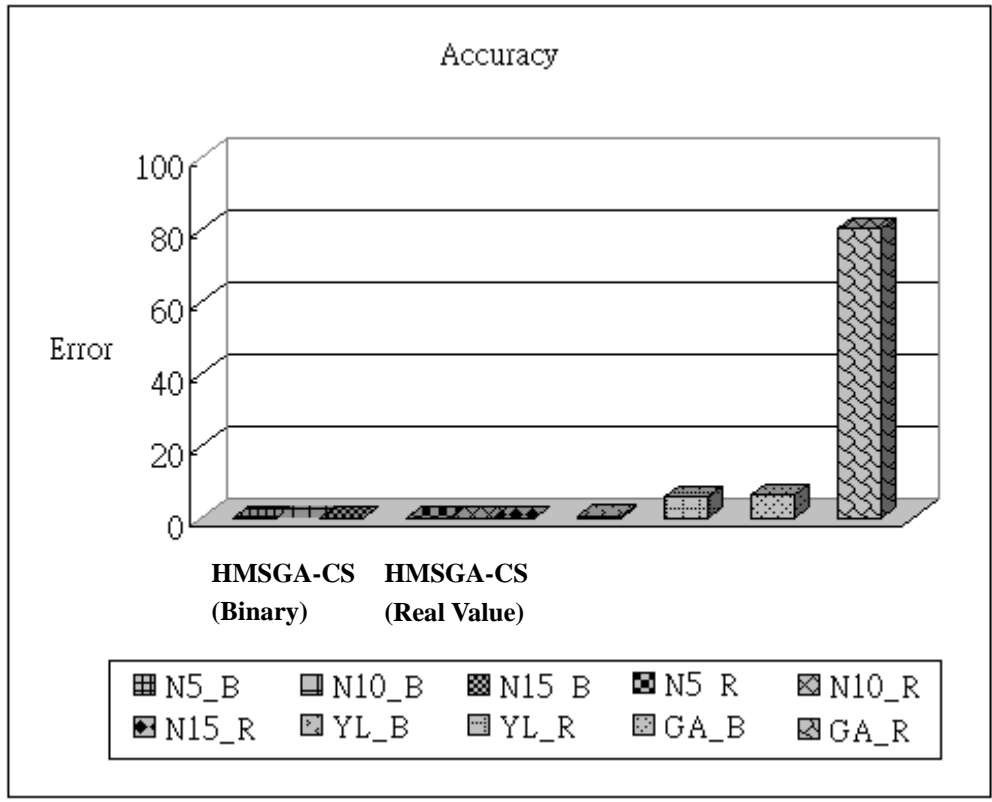


圖 4. 114 與理想值之誤差比較圖(算例十五)

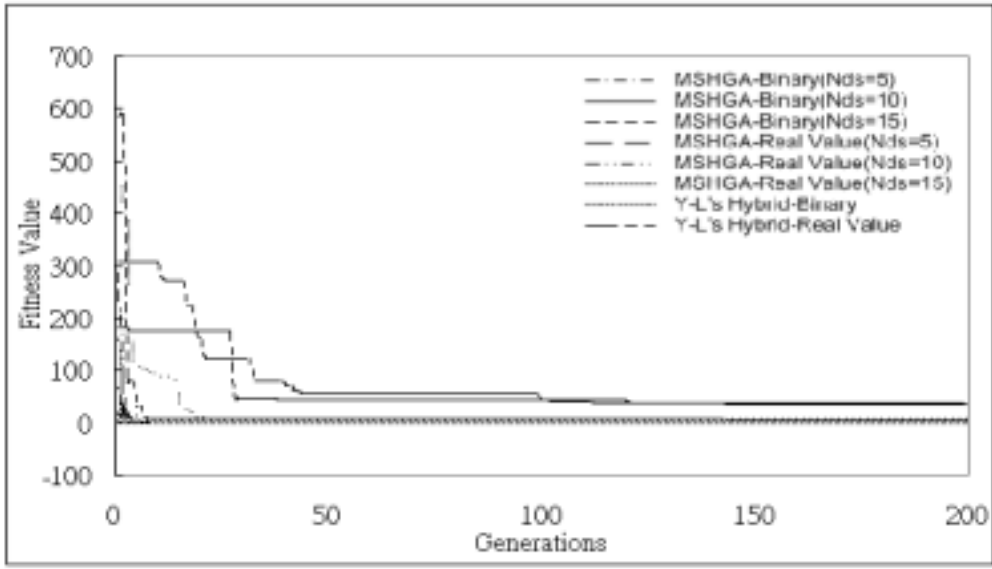


圖 4. 115 代數收斂表示圖(算例十五)

4.2 實驗結論

本研究提出不同的資料型態搭配不同的參數組合來進行驗證，在上列的測試結果表中，雖然本研究所採用的比較數據類別比 Y-L 的方法多，也就是 6：2，不過本研究並非想以多個較佳的數據來與較少不佳的數據比較。在經變異數分析(ANOVA)後，本研究證實了在大部分的範例中即使不同的參數組合對結果也不會有太大的影響，因此對於比較兩方法而言是屬於公平的。

針對本研究所驗證的部分可得到以下的結論：

1. 在上述 15 個範例的實驗結果中可得知，HMSGGA-CS 所測試出來的結果確實比傳統的基因演算法及 Y-L [52]的架構來的好很多。在『收斂百分比』方面，HMSGGA-CS 擁有 80% 以上的收斂百分比的算例佔 73.33%(22/30, 30 為 HMSGGA-CS 方法的 15 算例組合總數)。Y-L 擁有 80% 以上的收斂百分比的算例則佔 30%(9/30)，Y-L 擁有 20% 以下的收斂百分比的算例則佔 30%(9/30)。GA 擁有 80% 以上的收斂百分比的算例則佔 16.67%(5/30)，GA 擁有 20% 以下的收斂百分比的算例則佔 63.33%(19/30)。在『平均收斂代數』及『平均計算時間』方面，15 個算例皆有顯著的差異。
2. 在 HMSGGA-CS 中，單體演化代數越多，並不一定會造成較佳的結果，需視函數的特性來決定。函數的適應值差異越大，則可能需要較高的單體演化代數來讓單體鑽入其中，反之亦然。在 15 個測試的算例中，有 11 個算例滿足此條件。
3. 在 HMSGGA-CS 中，菁英化數目的比例越高，並不一定造成較佳的結果。原因在於有些函數的區域解過多，如果全採用菁英化來保留子代，便有可能使的單體的變化不大，協同性質不夠。在 2 個隨機挑選的算例中，皆滿足此假說。
4. 在測試組合表中，即使部分的例子有較佳的收斂代數或收斂時間，但必須先觀察其『收斂百分比』的比例。因為比例越低有可能使的平均代數、標準差及收斂時間下降，且此項指標也可看出該方法的穩定度。另一方面，由於本研究算例的比較方法其有些『收斂百分比』並未達到 100%，在進行變異數分析(ANOVA)時，是採 30 次

的結果進行分析，而未達 30 次的方法則以 1000 代來計算，因此分析出來的 F 值才會如此的大。

造成本實驗結果有如此優勢的表現，原因在於 HMSGGA-CS 藉由多群體的演算方式，搭配交配及突變等運算子來進行資訊的交換與溝通，並將每代演算後的組合以群體方式來演化下一代並記憶母代資訊，藉由這樣的協同機制，便能迅速包圍最佳解所在的區域並得知其資訊，接著再利用單體搜尋區域最佳解的能力，就能迅速找出最佳解。

第五章 結論與未來研究方向

5.1 結論

為實現協同式搜尋(Cooperative Search)的概念及證實其有效性，本研究將基因演算法與單體法(Simplex Method)合併使用來作為本研究的主要工具，並將其以 15 個範例來進行實驗驗證與實驗設計。本研究的研究成果可分為下列幾點：

1. 本研究提出一套新的演算法機制，主張協同式搜尋(Cooperative Search)的概念及重要性，以區別於其他文獻所提出的方法，並以實驗來測試本研究所主張的概念之效率，經實驗證實協同式搜尋(Cooperative Search)的機制卻實有必要存在演算法中，並能改善演算法搜尋的效率。
2. 藉由協同的機制使演算法建立起資訊的傳遞與溝通的功能後，並藉此來觸發與激化(Stimulate)所搜尋的群體，使演算法搜尋的效率更佳。
3. 由於將基因演算法與單體法(Simplex Method)合併的文獻並不多，藉由此概念與架構所設計出了運作模式，將單體法(Simplex Method)與基因演算法真正的合併使用，提供予相關研究者一篇參考的文獻，並可用於求解其它模式的問題。
4. 經由第四章實驗驗證的結果顯示，本研究所提出之方法在求解這些算例的效果比 Y-L 所提出的架構或 GA 有更佳的結果。
5. 本研究對於部分相關重要參數作實驗設計後，證實了參數的改變對於計算結果並無太大的影響，因此足以證明本研究所提出的架構並非會因問題的不同而使的結果差異甚大，是一優良的演算法。

5.2 未來研究方向

本研究目的在發展出一套新的演算法演算機制，將協同的概念以基因演算法與單體法合併後來實現並驗證本研究之結果好壞。為確立本研究研究之完整性，本研究歸納出幾項值得探討的未來研究方向，依重要性序列如下：

1.分合機制的建立

本研究在演算法分合機制的建立是以隨機方式產生，但如果考慮其他的分合機制，如自然界中動物之分合等，將其學習模式導入本研究之演算法中，相信對於實驗結果或許會有幫助。

2.群體間訊息傳遞與溝通機制的建立

本研究對於群體間訊息傳遞與溝通之機制採用隨機方式進行，但若考慮其他相關機制，如化學反應等，將其中相關及重要的想法納入其中，必能發展出一套新的演算法機制。

3.演算法之選擇

本研究將基因演算法與單體法合併使用來實現協同演算機制的建立，若能考慮其他演算法的合併，相信又是另一個可研究之方向。

4.群體與群體內數量之選擇

本研究群體與群體內數量之選擇是以單體法本身所需單體之結構所挑選的，若能改變單體的結構，不再以 $N+1$ 個點來形成 $N+1$ 的多邊形的話(N 為變數個數)，進而改變單體內的數量，或許能用較少的群體或點來創造更好的結果。

5.基因演算法運算子之參數探討

本研究之基因演算法相關運算子參數設定，如交配率及突變率等，是以隨機產生一組參數來進行實驗驗證，因此可在此方面多設定幾組參數來進行實驗驗證。

6.單體法運算子之參數探討

本研究在單體法運算子之參數設定方面是以隨機產生一組參數，同樣的若能多測試幾組參數設定，便可使本研究更具完整性。

7.系統之驗證

本研究在實驗驗證方面並未套入實際的模式，因此往後可以此方法套入實際的模式來求得最佳解。

參考文獻

- [1] 蘇木春, *機器學習類神經網路、模糊系統以及基因演算法則*, 全華出版社, 1997。
- [2] 游麗娟, 「基因演算法於幾何形狀最佳化設計之研究」, 國立中央大學機械工程研究所碩士論文, 八十九年六月。
- [3] 張子傑, *統計學講義*, 鼎茂圖書出版有限公司, 1997。
- [4] Adler D., “Genetic algorithms and simulated annealing: a marriage proposal”. *IEEE Int. Conf. On Neural Networks*, pp. 1104-1109, 1993.
- [5] Agapie A., F. Fagarasan, and B. Stanciulescu, “A genetic algorithm for fitting problem”, *Nuclear Instruments and Methods in Physics Research*, Vol. 389, pp.288-292, 1997.
- [6] Agnon Y., A. Golan, and M. Shearer, “Nonparametric, nonlinear, short-term forecasting : theory and evidence for nonlinearities in the commodity markets”, *Economics Letters*, Vol. 65, pp. 293-299, 1999.
- [7] Brown D., et al., “A parallel genetic heuristics for the quadratic assignment problem”, *Proc. 3rd Int. Conf. On Genetic Algorithms*, pp. 406-515, 1989.
- [8] Chen Y. W., S. Narieda, and K. Yamashita, “Blind nonlinear channel identification based on constrained hybrid genetic algorithm”, *IEEE Instrumentation and Measurement Technology Conference Budapest, Hungary*, 2001.
- [9] Costa L. and P. Oliveira, “Evolutionary algorithms approach to the solution of mixed integer non-linear programming problems”, *Computers and Chemical Engineering*, Vol. 25, pp. 257-266, 2001.
- [10] Crauwels H. A. J., C. N. Potts, L. N. Van Wassenhove, “Local search heuristics for single-machine scheduling with batching to minimize the number of the late jobs”, *European Journal of Operational Research*, Vol. 90, pp. 200-213, 1996.
- [11] Dozier G., J. Bowen, and D. Bahler, “Solving small and large scale constraint satisfaction problems using a heuristic-based microgenetic algorithm”, *Proceedings of 1st IEEE Conf. Evolutionary Computation*, Orlando, 1994.

- [12]Frieese T., et al., “Use of evolutionary algorithms for the calculation of group contribution parameters in order to predict thermodynamic properties Part 1: Genetic algorithms”, *Computers Chem. Engng*, Vol. 22, No. 11, pp. 1559-1572, 1998.
- [13]Gelsema E. S., “Special issue on genetic algorithms”, *Pattern Recognition Letters*, Vol. 16, pp. 779-780, 1995.
- [14]Glover F., J. P. Kelly, and M. Laguna, “Genetic algorithms and tabu search : hybrids for optimization”, *Computers Ops Res.*, Vol. 22, No. 1, pp. 111-134, 1995.
- [15]Goldberg D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Publishion, 1989.
- [16]Ishibuchi H., N. Yamamoto, T. Murata, and H. Tanaka, “Genetic algorithms and neighborhood search algorithms for fuzzy flowshop scheduling problems”, *Fuzzy Sets Syst.*, Vol. 67, No. 1, pp. 81-100, 1994.
- [17]Iwamatsu M., “Global geometry optimization of silicon clusters using the space-fixed genetic algorithm”, *Journal of chemical physics*, Vol. 112, Number 24, 2000.
- [18]Jeong I. K., and J. J. Lee, “Adaptive simulated annealing genetic algorithm for system identification”, *Engin Applic. Artif. Intell.*, Vol. 9, pp. 523-532, 1996.
- [19]Katayama K. and H. Sakamoto, “The efficiency of hybrid mutation genetic algorithm for the travelling salesman problem”, *Mathematical and Computer*, Vol. 31, pp. 197-203, 2000.
- [20]Leite J. P. B. and B. H. V. Topping, “Improved genetic operators for structural engineering optimization”, *Advanced in Engineering Software*, Vol. 29, No. 7-9, pp. 529-562, 1998.
- [21]Liaw C. F., “A hybrid genetic algorithm for open shop scheduling problem”, *European Journal of Operational Research*, Vol. 124, pp. 28-42, 2000.
- [22]Lin H. and K. Yamashita, “Hybrid simplex genetic algorithm for blind equalization using RBF networks”, *IEEE*, pp. 411-415, 2000.
- [23]Lis J., “Genetic algorithm with the dynamic probability of mutation in the classification problem”, *Pattern Recognition Letters*, Vol. 16, pp.

1311-1320, 1995.

- [24]Mannino M. V. and M. V. Koushik, “The cost-minimizing inverse classification problem : a genetic algorithm approach”, *Decision Support System*, Vol. 29, pp. 283-300, 2000.
- [25]Mathias K. E., L. D. Whitley, C. Stork, and T. Kusuma, “Staged hybrid genetic search for seismic data imaging”, *Proc. 1st IEEE Conf. Evolutionary Computation*, Orlando, pp. 356-361, 1994.
- [26]Mayer D. G., D. Schoorl, D. G. Butler, and A. M. Kelly, “Efficiency and fractal behaviour of optimization methods on multiple-optima surfaces.”, *Agric. Systems*, Vol. 36, pp. 315-328, 1991.
- [27]Mayer D. G., J. A. Belward, and K. Burrage, “Use of advanced techniques to optimize a multi-demensional dairy model”, *Agricultural Systems*, Vol. 50, pp. 239-253, 1996.
- [28]Michalewicz Z., et al., “Evolutionary operations for continuous convex parameter spaces”, *Proceedings of the Third Annual Conference on Evolutionary Programming*, pp. 84-97, 1994.
- [29]Michalewicz Z., *Genetic Algorithms + Data Structures = Evolution Programs*, Springer Verlag, New York, NY, 1992.
- [30]Mitchell M., ”L.D. Davis, Handbook of genetic algorithms”, *Artificial Intelligence*, Vol. 100, pp. 325-330, 1998.
- [31]Murata T. and H. Ishibuchi, “Performance evaluation of genetic algorithms for flowshop scheduling problems”, *Proceedings of 1st IEEE Conf. Evolutionary Computation*, Orlando, 1994.
- [32]Murata T., H. Ishibuchi, and H.. Tanaka, “Genetic algorithms for flowshop scheduling problems”, *Computers ind. Engng*, Vol. 30, No. 4, pp. 1061-1071, 1996.
- [33]Nelder J. A. and R. Mead, “A simplex method for function minimization”, *Computer journal*, Vol. 7, pp. 307-308, 1965.
- [34]Ohata T., Y. Nakamura, T. Katayama, and E. Nakamachi, “Development of optimum process design system by numerical simulation”, *Journal of Materials Proceeding Technology*, Vol. 60, pp. 543-548, 1996.
- [35]Okamoto M. and T. Nonaka, “Nonlinear numerical optimization with use of a hybrid genetic algorithm incorporating the modified powell

- method”, *Applied Mathematics and Computation*, Vol. 91, pp. 63-72, 1998.
- [36]Pirlot M., “General local search methods”, *European Journal of Operational Research*, Vol. 92, pp. 493-511, 1992.
- [37]RAO S. S., *Engineering Optimization third ed.*, A Wiley-Interscience Publication, 1996.
- [38]Reid D. J., “Enhanced genetic operators for the resolution of discrete constrained optimization problems”, *Computers Ops. Res.*, Vol. 24, No. 5, pp. 399-411, 1997.
- [39]Renders J. and H. Bersini, “Hybridizing genetic algorithms with hill-climbing methods for global optimization: Two possible ways”, *Proceedings of the 1st IEEE Conference on Evolutionary Computation*, Orlando, pp. 312-317, 1994.
- [40]Renders J. and S. Flasse, “Hybrid methods using genetic algorithms for global optimization”, *IEEE Trans. Syst., Man, Cybern B*, Vol. 26, pp. 243-258, 1996.
- [41]Rogers D., “G/SPLINES: A hybrid of Friedman’s multivariate adaptive regression splines (MARS) algorithm with Holland’s genetic algorithm”, *Proceedings of 4th Int. Conf. Genetic Algorithms*, pp. 384-391, San Diego, 1991.
- [42]Shaffer R. E. and G. W. Small, “Comparison of optimization algorithms for piecewise linear discriminant analysis: application to Fourier transform infrared remote sensing measurements”, *Analytica Chimica Acta 331*, pp. 157-175, 1996.
- [43]Sirag D. and P. Weisser, “Toward a unified thermodynamic”. *Proc. 2nd Int. Conf. On Genetic Algorithms*, pp. 116-122, 1987.
- [44]De Souza P. S. and S. N. Talukdar, “Genetic Algorithm in asynchronous teams”, *Proceedings of 4th Int. Conf. Genetic Algorithms*, San Diego, pp. 392-397, 1991.
- [45]Thomas G. M., R. Gerth, T. Velasco and L.C. Rabelo, “Using real-coded genetic algorithms for weibull parameter estimation”, *Computers ind. Engng*, Vol. 29, No. 1-4, pp. 377-381, 1995.
- [46]Spendley W., G. R. Hext, and F. R. Himsworth, “Sequential application of simplex designs in optimization and evolutionary

- operation”, *Technometrics*, Vol. 4, pp. 400-441, 1962.
- [47]Wang D. and S.-C. Fang, “A semi-infinite programming model for earliness/tardiness production planning with a genetic algorithm”, *Computers Math. Applic.*, Vol. 31, No. 8, pp. 95-106, 1996.
- [48]Wang Q. J., “Using genetic algorithms to optimize model parameters”, *Environmental Modeling & Software*, Vol. 12, No. 1, pp. 27-34, 1997.
- [49]Wu A. S. and R. K. Lindsay, “A comparison of fixed and floating building block representation in the genetic algorithm”, *Evolutionary Computation*, Vol. 4, pp. 169-193, 1997.
- [50]Wu S. J. and P. T. Chow, “Steady-state genetic algorithms for discrete optimization of trusses”, *Computers & Structures*, Vol. 56, No. 6, pp. 979-991, 1995.
- [51]Yen J. and B. Lee, ”A simplex genetic algorithm hybrid”, *IEEE*, pp. 175-180, 1997.
- [52]Yen J., J. C. Liao, B. Lee, and D. Randolph, “A hybrid approach to modeling metabolic systems using a genetic algorithm and simplex method”, *IEEE Transactions On Systems. Man, and Cybernetics*, pp. 173-191, 1998.

附錄一 變異數分析(ANOVA)

本研究將針對各方法不同結構下的組合進行變異數分析，以驗證結果的差異性是否顯著。

1.作法[3]：

	A_1 A_2 A_c
B_1	X_{ijk}
B_2	
\vdots	
B_r	

$$SST = SSC + SSR + SSI + SSE$$

$$SST = \sum \sum \sum (X_{ijk} - \bar{\bar{x}})^2$$

$$SSR = c \sum (\bar{x}_{i.} - \bar{\bar{x}})^2$$

$$SSC = r \sum (\bar{x}_{.j} - \bar{\bar{x}})^2$$

$$SSI = \sum \sum \sum (\bar{x}_{ij.} - \bar{x}_{i.} - \bar{x}_{.j} + \bar{\bar{x}})^2$$

$$SSE = \sum \sum \sum (X_{ijk} - \bar{x}_{ij.})^2$$

ANOVA TABLE

變異來源	SS	df	MS	決策法則
A 因素	SSC	c-1	$MSC = \frac{SSC}{r-1}$	$F_c = \frac{MSC}{MSE}$ if $F_c \leq F_{[1-\alpha; c-1; rc(n-1)]}$ then accept H_0
B 因素	SSR	r-1	$MSR = \frac{SSR}{c-1}$	$F_r = \frac{MSR}{MSE}$ if $F_r \leq F_{[1-\alpha; c-1; rc(n-1)]}$ then accept H_0'
交互因素	SSI	$(r-1)*(c-1)$	$MSI = \frac{SSI}{(r-1)(c-1)}$	$F_l = \frac{MSI}{MSE}$ if $F_l \leq F_{[1-\alpha; (r-1)(c-1); rc(n-1)]}$ then accept H_0''
誤差	SSE	rc(n-1)	$MSE = \frac{SSE}{rc(n-1)}$	

Total	SST	N-1		
-------	-----	-----	--	--

2.結果：

(1)算例一、De Jong F5 function

a.變異數分析表

i.代數

表 A1. 1

變異來源	SS	自由度	MS	F	P-值
方法	53848917	9	5983213	262.44	0.0000000*
組內	6611534	290	22798.39		
總和	60460451	299			

表 A1. 2

變異來源	SS	自由度	MS	F	P-值
方法	2386	2	1193	2.58	0.0813
組內	40189	87	462		
總和	42576	89			

ii.時間

表 A1. 3

變異來源	SS	自由度	MS	F	P-值
方法	509996	9	56666	96.667	0.0000000*
組內	169996	290	586.194		
總和	679992	299			

表 A1. 4

變異來源	SS	自由度	MS	F	P-值
方法	46.37	5	9.27	1.106	0.3586
組內	1458.08	174	8.3797		
總和	1504.45	179			

b.結果分析

表 A1. 1 及表 A1. 3 為我們的分析結果。在顯著水準 $\alpha = 0.05$ 的條件下，我們可得知：

- i. 方法的不同對收斂代數有顯著的影響；
- ii. 方法的不同對收斂時間有顯著的影響；

(2)算例二、Sin function

a.變異數分析表

i.代數

表 A1. 5

變異來源	SS	自由度	MS	F	P-值
方法	4.69E+09	9	5.21E+08	5.85	0.000000216*
組內	8.02E+09	90	89070589		
總和	1.27E+10	99			

ii.時間

表 A1. 6

變異來源	SS	自由度	MS	F	P-值
方法	3.56E+09	9	3.95E+08	329	0.000000000*
組內	1.08E+08	90	1200045		
總和	3.67E+09	99			

b.結果分析

表 A1. 5 及表 A1. 6 為我們的分析結果。在顯著水準 $\alpha = 0.05$ 的條件下，我們可得知：

- i. 方法的不同對收斂代數有顯著的影響；
- ii. 方法的不同對收斂時間有顯著的影響；

(3)算例三、De Jong F1 function

a.變異數分析表

i.代數

表 A1. 7

變異來源	SS	自由度	MS	F	P-值
方法	6786279	9	754030	50.589	0.0000000*
組內	4322385	290	14904		
總和	11108660	299			

ii.時間

表 A1. 8

變異來源	SS	自由度	MS	F	P-值
方法	12850	9	1427	16.32	0.00000000*
組內	25370	290	87		
總和	38220	299			

表 A1. 9

變異來源	SS	自由度	MS	F	P-值
方法	0.3173	2	0.1586	1.312	0.2745
組內	10.52	87	0.1209		
總和	10.838	89			

b. 結果分析

表 A1. 7 及表 A1. 8 為我們的分析結果。在顯著水準 $\alpha = 0.05$ 的條件下，我們可得知：

- i. 方法的不同對收斂代數有顯著的影響；
- ii. 方法的不同對收斂時間有顯著的影響；

(4) 算例四、De Jong F2 function

a. 變異數分析表

i. 代數

表 A1. 10

變異來源	SS	自由度	MS	F	P-值
方法	8992457	7	1284637	12.05871	4.11E-13*
組內	24715387	232	106531.8		
總和	33707843	239			

ii. 時間

表 A1. 11

變異來源	SS	自由度	MS	F	P-值
方法	45134.39	7	6447.771	2.541335	0.015427*
組內	588620.8	232	2537.159		
總和	633755.2	239			

b. 結果分析

表 A1. 10 及表 A1. 11 為我們的分析結果。在顯著水準 $\alpha = 0.05$ 的條件下，我們可得知：

- i. 方法的不同對收斂代數有顯著的影響；
- ii. 方法的不同對收斂時間有顯著的影響；

(5)算例五、Schaffer F6 function

a.變異數分析表

i.代數

表 A1. 12

變異來源	SS	自由度	MS	F	P-值
方法	14597110	9	1621901	9.188	0.00000000*
組內	51188218	290	176511		
總和	65785328	299			

表 A1. 13

變異來源	SS	自由度	MS	F	P-值
方法	1118114	5	223622	0.9044	0.479
組內	43018680	174	247223		
總和	65785328	179			

ii.時間

表 A1. 14

變異來源	SS	自由度	MS	F	P-值
方法	3994063	9	443784	36.532	0.00000000*
組內	3522823	290	12147		
總和	7516887	299			

b.結果分析

表 A1. 12 及表 A1. 14 為我們的分析結果。在顯著水準 $\alpha = 0.05$ 的條件下，我們可得知：

- i. 方法的不同對收斂代數有顯著的影響；
- ii. 方法的不同對收斂時間有顯著的影響；

(6)算例六、Schaffer F7 function

a.變異數分析表

i.代數

表 A1. 15

變異來源	SS	自由度	MS	F	P-值
方法	48989258	9	5436584	228.316	0.0000000*
組內	6905367	290	23811		
總和	55834625	299			

表 A1. 16

變異來源	SS	自由度	MS	F	P-值
方法	35725	5	7145	1.317	2.266
組內	943917	174	5424		
總和	979643	179			

ii.時間

表 A1. 17

變異來源	SS	自由度	MS	F	P-值
方法	129308	9	14367	45.975	0.00000000*
組內	90627	290	312		
總和	219936	299			

表 A1. 18

變異來源	SS	自由度	MS	F	P-值
方法	182.55	5	36.51	1.31	0.2619
組內	4850.22	174	27.87		
總和	5032.77	179			

b.結果分析

表 A1. 15 及表 A1. 17 為我們的分析結果。在顯著水準 $\alpha = 0.05$ 的條件下，我們可得知：

- i. 方法的不同對收斂代數有顯著的影響；
- ii. 方法的不同對收斂時間有顯著的影響；

(7)算例七、Shubert function

a.變異數分析表

i.代數

表 A1. 19

變異來源	SS	自由度	MS	F	P-值
方法	32281541	9	3586838	48.5914	0.0000000*
組內	21406731	290	73816		
總和	53688272	299			

表 A1. 20

變異來源	SS	自由度	MS	F	P-值
方法	4799.267	2	2399.633	2.164	0.121
組內	96475.23	87	1108.915		
總和	101274.9	89			

ii.時間

表 A1. 21

變異來源	SS	自由度	MS	F	P-值
方法	323423	9	35935	12.438	0.00000000*
組內	837849	290	2889		
總和	1161273	299			

b.結果分析

表 A1. 19 及表 A1. 21 為我們的分析結果。在顯著水準 $\alpha = 0.05$ 的條件下，我們可得知：

- i. 方法的不同對收斂代數有顯著的影響；
- ii. 方法的不同對收斂時間有顯著的影響；

(8)算例八、SinCos function

a.變異數分析表

i.代數

表 A1. 22

變異來源	SS	自由度	MS	F	P-值
方法	17201350	7	2457336	21.36486	3.89E-22*
組內	26684092	232	115017.6		
總和	43885442	239			

ii.時間

表 A1. 23

變異來源	SS	自由度	MS	F	P-值
方法	185865.3	7	26552.19	6.938257	1.66E-07*
組內	887846.6	232	3826.925		
總和	1073712	239			

b.結果分析

表 A1. 22 及表 A1. 23 為我們的分析結果。在顯著水準 $\alpha = 0.05$ 的條件下，我們可得知：

- i. 方法的不同對收斂代數有顯著的影響；
- ii. 方法的不同對收斂時間有顯著的影響；

(9)算例九、Goldstein-Price function

a.變異數分析表

i.代數

表 A1. 24

變異來源	SS	自由度	MS	F	P-值
方法	28703669	9	3189300	108.88	0.000000000*
組內	8494851	290	29292.6		
總和	37198550	299			

表 A1. 25

變異來源	SS	自由度	MS	F	P-值
方法	1118.04	5	223.6	2.255	0.050991
組內	17251.5	174	99.15		
總和	18369.5	179			

ii.時間

表 A1. 26

變異來源	SS	自由度	MS	F	P-值
方法	28868	9	3207.66	13.945	0.000000000*
組內	66704	290	230.015		

總和	95573	299			
----	-------	-----	--	--	--

表 A1. 27

變異來源	SS	自由度	MS	F	P-值
方法	17.399	5	3.48	1.066	0.381
組內	568.06	174	3.265		
總和	585.46	179			

b. 結果分析

表 A1. 24 及表 A1. 26 為我們的分析結果。在顯著水準 $\alpha = 0.05$ 的條件下，我們可得知：

- i. 方法的不同對收斂代數有顯著的影響；
- ii. 方法的不同對收斂時間有顯著的影響；

(10)算例十、Bohachevsky #1 function

a. 變異數分析表

i. 代數

表 A1. 28

變異來源	SS	自由度	MS	F	P-值
方法	31894504	9	3543834	132.75	0.000000000*
組內	7741490	290	26694.8		
總和	39635994	299			

表 A1. 29

變異來源	SS	自由度	MS	F	P-值
方法	32252.78	5	6450.557	1.16	0.331
組內	967587	174	5560.845		
總和	999839.8	179			

ii. 時間

表 A1. 30

變異來源	SS	自由度	MS	F	P-值
方法	47413.83	9	5268.203	51.98	0.00000000*
組內	29394.1	290	101.36		

總和	76807.93	299			
----	----------	-----	--	--	--

表 A1. 31

變異來源	SS	自由度	MS	F	P-值
方法	143.541	5	28.71	0.768	0.574
組內	6506.451	174	37.4		
總和	6650	179			

b. 結果分析

表 A1. 28 及表 A1. 30 為我們的分析結果。在顯著水準 $\alpha = 0.05$ 的條件下，我們可得知：

- i. 方法的不同對收斂代數有顯著的影響；
- ii. 方法的不同對收斂時間有顯著的影響；

(11) 算例十一、Easom function

a. 變異數分析表

i. 代數

表 A1. 32

變異來源	SS	自由度	MS	F	P-值
方法	55198875	9	6133208	498.1367	0.00000000*
組內	3570567	290	12312.3		
總和	58769442	299			

ii. 時間

表 A1. 33

變異來源	SS	自由度	MS	F	P-值
方法	228288	9	25365	538.501	0.00000000*
組內	13612	290	47.103		
總和	241901	299			

b. 結果分析

表 A1. 32 及表 A1. 33 為我們的分析結果。在顯著水準 $\alpha = 0.05$ 的條件下，我們可得知：

- i. 方法的不同對收斂代數有顯著的影響；

ii. 方法的不同對收斂時間有顯著的影響；

(12)算例十二、SQRN5 function

a.變異數分析表

i.代數

表 A1. 34

變異來源	SS	自由度	MS	F	P-值
方法	33942280	9	3771431	28.893	0.00000000*
組內	37854021	290	130531		
總和	71796902	299			

表 A1. 35

變異來源	SS	自由度	MS	F	P-值
方法	95497	2	47748.5	0.36	0.699
組內	11554411	87	132809		
總和	11649908	89			

ii.時間

表 A1. 36

變異來源	SS	自由度	MS	F	P-值
方法	6699448.5	9	74438	9.55	0.00000000*
組內	2259469	290	7791		
總和	2929418	299			

表 A1. 37

變異來源	SS	自由度	MS	F	P-值
方法	894.274	2	447.14	0.169	0.845
組內	230531.4	87	2649.8		
總和	231425.7	89			

b.結果分析

表 A1. 34 及表 A1. 36 為我們的分析結果。在顯著水準 $\alpha = 0.05$ 的條件下，我們可得知：

i. 方法的不同對收斂代數有顯著的影響；

ii. 方法的不同對收斂時間有顯著的影響；

(13)算例十三、SQRN7 function

a.變異數分析表

i.代數

表 A1. 38

變異來源	SS	自由度	MS	F	P-值
方法	53612915	9	5956991	162.68	0.0000000*
組內	10618872	290	36616.8		
總和	64231787	299			

表 A1. 39

變異來源	SS	自由度	MS	F	P-值
方法	113204	5	22640	2.14	0.063
組內	1841061	174	10580		
總和	1954265	179			

ii.時間

表 A1. 40

變異來源	SS	自由度	MS	F	P-值
方法	719201.4	9	79911	17.345	0.000000000*
組內	197488.1	290	680.99		
總和	916689.4	299			

表 A1. 41

變異來源	SS	自由度	MS	F	P-值
方法	937.41	5	187.5	1.886	0.099
組內	17300.21	174	99.43		
總和	18237	179			

b.結果分析

表 A1. 38 及表 A1. 40 為我們的分析結果。在顯著水準 $\alpha = 0.05$ 的條件下，我們可得知：

i. 方法的不同對收斂代數有顯著的影響；

ii. 方法的不同對收斂時間有顯著的影響；

(14)算例十四、SQRN10 function

a.變異數分析表

i.代數

表 A1. 42

變異來源	SS	自由度	MS	F	P-值
方法	54363980	9	6040442	323.103	0.0000000*
組內	5421574	290	18695		
總和	59785554	299			

ii.時間

表 A1. 43

變異來源	SS	自由度	MS	F	P-值
方法	697975	9	77552	92.3	0.00000000*
組內	243724	290	840		
總和	941699	299			

b.結果分析

表 A1. 42 及表 A1. 43 為我們的分析結果。在顯著水準 $\alpha = 0.05$ 的條件下，我們可得知：

i. 方法的不同對收斂代數有顯著的影響；

ii. 方法的不同對收斂時間有顯著的影響；

(15)算例十五、Colville function

a.變異數分析表

i.代數

表 A1. 44

變異來源	SS	自由度	MS	F	P-值
方法	69487507	9	7720834	4667	0.00000000*
組內	479690.4	290	1654.01		
總和	69967197	299			

ii.時間

表 A1. 45

變異來源	SS	自由度	MS	F	P-值
方法	1228098	9	136455	4750	0.000000000*
組內	8330	290	28.72		
總和	1236428	299			

b. 結果分析

表 A1. 44 及表 A1. 45 為我們的分析結果。在顯著水準 $\alpha = 0.05$ 的條件下，我們可得知：

- i. 方法的不同對收斂代數有顯著的影響；
- ii. 方法的不同對收斂時間有顯著的影響；