

類梯度搜尋演算策略方法

研究生：郭文偉

指導教授：張炳騰 教授

曾宗瑤 教授

東海大學工業工程與經營資訊研究所

摘要

演化策略利用電腦模擬演化的現象，透過演化計算搜尋複雜問題的最佳解，目前被大量應用在搜尋、最佳化、機器學習、排程、製造系統與醫學應用方面，然而在最佳化問題的求解過程中，如何有效率的搜尋到目標函數的全域最佳解並不是一件容易的事，一般的演化策略由於搜尋目標函數的最佳解的效率取決於突變強度的大小，因此，若演化策略作目標函數最佳解的搜尋時若使用較小的突變強度，在細部搜尋時會較有效率，但若母體落在目標函數平坦區域時其演化效率相對較差，更重要的是，當目標函數為有多重區域解時，若突變強度過小會造成無法跳脫區域解的情況。若使用的突變強度較大，雖然可以跳脫目標函數的區域解，但其整體的搜尋效率會變的相對較差，特別是在靠近最佳解區域作細部搜尋時不易快速的逼近最佳解。

本研究提出的類梯度搜尋演算策略方法(Quasi Gradient Search Evolution Strategy Method; QGSES Method)，主要重點在於母體搜尋最佳解時會應先將周圍的資訊先比較過後，根據母體的梯度向量與母體歷史移動向量判斷子代往最佳解區域逼近的最佳方向與距離，加速演化策略整體的演化速度，進一步提升演化策略的搜尋效率。藉由這樣的機制使整體族群在搜尋時能快速且有效的得到最佳點所在位置的資訊並向附近的區域快速的逼近，以達到快速收斂的目的。

經由實驗驗證的結果顯示，本研究所提出之方法在求解多變數函數有相當好的的收斂效率，特別在有多重區域解的函數中優秀的全域搜尋能力，因此證明本研究所提出的架構擁有優秀的多重區域解函數的全域搜尋能力以及穩定的最佳解收斂效率，是一優良的演算法。

關鍵字：演化策略、最佳化、收斂效率、多重區域解函數、全域搜尋

Quasi Gradient Search Evolution Strategy Method

Student : Wen-Wei Kuo

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

Institute of Industrial Engineering and Enterprise Information
Tunghai University

ABSTRACT

Evolution strategies utilize the computer to simulate the phenomenon of evolution, calculate and search the best solution of complicated problem through evolutionary computation. At present, evolution strategies apply to search, optimization, machine learning, scheduling, manufacturing and medicine application etc. How to search the global optimization solution efficiently in object function is a difficult thing in the process of solving optimization questions. In general, the efficiency of evolution strategies depends on the size of mutation strength. Therefore, when evolution strategies use the small mutation strength to search the optimization solution in object function, it will be relatively efficient in searching the details. But it evolves inefficiency when the parent lands on the smooth area of object function. Moreover, small mutation strength will make the parents to escape the local solution hardly when the object function has multi-local solutions. It can escape the local solution when evolution strategies use the large mutation strength, but it will be relatively efficient in searching as a whole. Especially, it will be difficult to close the optimization solution efficiently when parents approach the region of optimization solution and search the details.

This paper brings up the Quasi Gradient Search Evolution Strategy Method (QGSES Method). The main focal point of QGSES is that parents should consider about the environment information around itself before searching the optimization solution. According to gradient vector and historical moving vector of parents judge the best direction and distance of offspring to approach the region of optimization solution. With the method of QGSES, we want to precipitate the whole evolutionary speed and improve the searching efficiency of evolution strategies. In order to achieve the goal of quickly convergence, the whole population can get the information around the optimization solution efficiency and surround the nearby region of the optimization solution quickly by this system.

The result that proves via the experiment shows that the method of this research institute nice convergence efficiency in multi-variable function and excellent global searching capability in multi-local solutions function. It proves that the architecture of this research have excellent global searching capability in multi-local solutions function and steady convergence efficiency of the optimization solution. It is a nice algorithm of searching.

Keywords: Evolution strategies; Optimization; Convergence efficiency; Multi-local solutions function; Global search.

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、洪堯勳老師、白炳豐老師及陳琨太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，感謝學弟妹的鼓勵及支持，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。

郭文偉 謹誌於
東海大學工業工程與經營資訊學系
民國九十四年六月

目錄

摘要.....	I
ABSTRACT.....	II
誌謝.....	IV
目錄.....	V
表目錄.....	VII
圖目錄.....	VIII
第一章 緒論.....	1
1.1 研究背景與動機.....	1
1.2 研究目的.....	1
1.3 研究流程與架構.....	2
1.4 研究工具.....	3
第二章 文獻探討.....	4
2.1 演化理論(EVOLUTIONARY ALGORITHMS)相關文獻.....	4
2.2.1 演化理論簡介.....	4
2.2.2 演化理論模型.....	4
2.2 演化策略(EVOLUTION STRATEGIES; ESS).....	6
2.2.1 演化策略簡介.....	6
2.2.2 天擇(selection).....	7
2.2.3 自我適應能力 (Self-Adaptation).....	8
2.2.4 重構(recombination).....	9
第三章 研究方法.....	11
3.1 QGSES 原理.....	11
3.2 QGSES 系統架構.....	12
3.3 QGSES 演算法.....	15
3.3.1 QGSES 演算法符號說明表.....	16
3.3.2 QGSES 演算法.....	16
3.4 QGSES 與其他演算法的結構比較.....	21
第四章 數值分析與評估.....	23
4.1 多變數函數分析.....	23
4.1.1 函數 1：Colville function.....	23
4.1.2 函數 2：Rosenbrock's function.....	31
4.1.3 函數 3：Shubert function.....	38
4.1.4 函數 4：SinCos function.....	45
4.1.5 函數 5： $x\sin(1/x)$ function with 2 variables.....	52
4.1.6 函數 6： $x\sin(1/x)$ function with 5 variables.....	59
4.1.7 函數 7： $x\sin(1/x)$ function with 10 variables.....	66
4.2 實驗結論.....	68
4.2.1 QGSES 與其他演算法的搜尋效率比較.....	68

4.2.2	QGSES 與其他演算法的搜尋原理分析	68
第五章	結論與建議	71
5.1	研究總結	71
5.2	後續研究建議	71
	參考文獻	73
附錄	μ 與 λ 對 QGSES 搜尋效率的影響數據	77

表目錄

表 3-1 符號說明表	16
表 3-2 QGSES 與其他演算法的結構比較	22
表 4-1 應變數 λ 在 Colville function 中的實驗數據	25
表 4-2 應變數 μ 在 Colville function 中的實驗數據	26
表 4-3 應變數 β 在 Colville function 中的實驗數據	27
表 4-4 應變數 σ 在 Colville function 中的實驗數據	28
表 4-5 不同的演算法對 Colville function 的測試結果	30
表 4-6 應變數 λ 在 Rosenbrock's function 中的實驗數據	32
表 4-7 應變數 μ 在 Rosenbrock's function 中的實驗數據	33
表 4-8 應變數 β 在 Rosenbrock's function 中的實驗數據	34
表 4-9 應變數 σ 在 Rosenbrock's function 中的實驗數據	35
表 4-10 不同的演算法對 Rosenbrock's function 的測試結果	37
表 4-11 應變數 λ 在 Shubert function 中的實驗數據	39
表 4-12 應變數 μ 在 Shubert function 中的實驗數據	40
表 4-13 應變數 β 在 Shubert function 中的實驗數據	41
表 4-14 應變數 σ 在 Shubert function 中的實驗數據	42
表 4-15 不同的演算法對 Shubert function 的測試結果	44
表 4-16 應變數 λ 在 SinCos function 中的實驗數據	46
表 4-17 應變數 μ 在 SinCos function 中的實驗數據	47
表 4-18 應變數 β 在 SinCos function 中的實驗數據	48
表 4-19 應變數 σ 在 SinCos function 中的實驗數據	49
表 4-20 不同的演算法對 SinCos function 的測試結果	51
表 4-21 應變數 λ 在 $x\sin(1/x)$ function with 2 variables 中的實驗數據	53
表 4-22 應變數 μ 在 $x\sin(1/x)$ function with 2 variables 中的實驗數據	54
表 4-23 應變數 β 在 $x\sin(1/x)$ function with 2 variables 中的實驗數據	55
表 4-24 應變數 σ 在 $x\sin(1/x)$ function with 2 variables 中的實驗數據	56
表 4-25 不同的演算法對 $x\sin(1/x)$ function with 2 variables 的測試結果	58
表 4-26 應變數 λ 在 $x\sin(1/x)$ function with 5 variables 中的實驗數據	60
表 4-27 應變數 μ 在 $x\sin(1/x)$ function with 5 variables 中的實驗數據	61
表 4-28 應變數 β 在 $x\sin(1/x)$ function with 5 variables 中的實驗數據	62
表 4-29 應變數 σ 在 $x\sin(1/x)$ function with 5 variables 中的實驗數據	63
表 4-30 不同的演算法對 $x\sin(1/x)$ function with 5 variables 的測試結果	65
表 4-31 不同的演算法對 $x\sin(1/x)$ function with 10 variables 的測試結果	67

圖目錄

圖 1-1 研究方法與步驟.....	3
圖 3-1 QGSES 架構圖.....	14
圖 3-2 Pseudo-code of QGSES.....	15
圖 3-3 QGSES 與演化策略往最佳解區域逼近的比較.....	19
圖 3-4 QGSES 在目標函數平坦區域的移動向量.....	20
圖 3-5 QGSES 在目標函數最佳解區域的移動向量.....	21
圖 4-1 μ 與 λ 對 QGSES 在 Colville function 中搜尋效率的影響.....	25
圖 4-2 λ 對 QGSES 在 Colville function 中搜尋效率的影響.....	26
圖 4-3 μ 對 QGSES 在 Colville function 中搜尋效率的影響.....	27
圖 4-4 β 對 QGSES 在 Colville function 中搜尋效率的影響.....	28
圖 4-5 σ 對 QGSES 在 Colville function 中搜尋效率的影響.....	29
圖 4-6 Colville function 之平均運算次數比較圖.....	30
圖 4-7 μ 與 λ 對 QGSES 在 Rosenbrock's function 中搜尋效率的影響.....	32
圖 4-8 λ 對 QGSES 在 Rosenbrock's function 中搜尋效率的影響.....	33
圖 4-9 μ 對 QGSES 在 Rosenbrock's function 中搜尋效率的影響.....	34
圖 4-10 β 對 QGSES 在 Rosenbrock's function 中搜尋效率的影響.....	35
圖 4-11 σ 對 QGSES 在 Rosenbrock's function 中搜尋效率的影響.....	36
圖 4-12 Rosenbrock's function 之平均次數時間比較圖.....	37
圖 4-13 μ 與 λ 對 QGSES 在 Shubert function 中搜尋效率的影響.....	39
圖 4-14 λ 對 QGSES 在 Shubert function 中搜尋效率的影響.....	40
圖 4-15 μ 對 QGSES 在 Shubert function 中搜尋效率的影響.....	41
圖 4-16 β 對 QGSES 在 Shubert function 中搜尋效率的影響.....	42
圖 4-17 σ 對 QGSES 在 Shubert function 中搜尋效率的影響.....	43
圖 4-18 Shubert function 之平均運算次數比較圖.....	44
圖 4-19 μ 與 λ 對 QGSES 在 SinCos function 中搜尋效率的影響.....	46
圖 4-20 λ 對 QGSES 在 SinCos function 中搜尋效率的影響.....	47
圖 4-21 μ 對 QGSES 在 SinCos function 中搜尋效率的影響.....	48
圖 4-22 β 對 QGSES 在 SinCos function 中搜尋效率的影響.....	49
圖 4-23 σ 對 QGSES 在 SinCos function 中搜尋效率的影響.....	50
圖 4-24 SinCos function 之平均運算次數比較圖.....	51
圖 4-25 μ 與 λ 對 QGSES 在 $x\sin(1/x)$ function with 2 variables 中搜尋效率的影響.....	53
圖 4-26 λ 對 QGSES 在 $x\sin(1/x)$ function with 2 variables 中搜尋效率的影響.....	54
圖 4-27 μ 對 QGSES 在 $x\sin(1/x)$ function with 2 variables 中搜尋效率的影響.....	55
圖 4-28 β 對 QGSES 在 $x\sin(1/x)$ function with 2 variables 中搜尋效率的影響.....	56
圖 4-29 σ 對 QGSES 在 $x\sin(1/x)$ function with 2 variables 中搜尋效率的影響.....	57
圖 4-30 $x\sin(1/x)$ function with 2 variables 之平均運算次數比較圖.....	58
圖 4-31 μ 與 λ 對 QGSES 在 $x\sin(1/x)$ function with 5 variables 中搜尋效率的影響.....	

響.....	60
圖 4-32 λ 對 QGSES 在 $x\sin(1/x)$ function with 5 variables 中搜尋效率的影響	61
圖 4-33 μ 對 QGSES 在 $x\sin(1/x)$ function with 5 variables 中搜尋效率的影響	62
圖 4-34 β 對 QGSES 在 $x\sin(1/x)$ function with 5 variables 中搜尋效率的影響	63
圖 4-35 σ 對 QGSES 在 $x\sin(1/x)$ function with 5 variables 中搜尋效率的影響	64
圖 4-36 $x\sin(1/x)$ function with 5 variables 之平均運算次數比較圖.....	65
圖 4-37 $x\sin(1/x)$ function with 10 variables 之平均運算次數比較圖.....	67

第一章 緒論

1.1 研究背景與動機

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(Genetic Algorithms; GAs)是採用隨機搜尋的方式，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，演化策略(Evolution strategies; ESs)雖然可以穩定的往最佳解的方向逼近且在細部搜尋上有較佳的效率，但若初始母體距離最佳解較遠時，子代的演化距離受限於其突變強度(Mutation strength)的侷限，無法快速的朝最佳解的方向逼近，進而造成搜尋的時間增加，效率不佳。

QGSES(Quasi Gradient Search Evolution Strategy)主要在母體搜尋最佳解時會應先將周圍的資訊先比較過後再決定子代突變的方向與距離，而非演化策略中子代個別隨機突變的運作方式，藉由這樣的機制使整體族群在搜尋時能快速且有效的得到最佳點所在位置的資訊並向附近的區域快速的包圍住，以達到快速收斂的目的。

1.2 研究目的

傳統的演化策略由於使用固定的突變強度(mutation strength)，因此在搜尋目標函數最佳解時，子代的演化距離受限於突變強度的侷限，當母體落在目標函數平坦區域時，無法快速的朝最佳解的方向逼近，此外，固定的突變強度在靠近最佳解區域時作細部搜尋時的效率取決於突變強度的大小，因此，傳統的演化策略作目標函數最佳解的搜尋時若使用較小的突變強度，在細部搜尋時會較有效率，但若母體落在目標函數平坦區域時其演化效率相對較差，更重要的是，當目標函數為有多重區域(local)解時，若突變強度過小會造成無法跳脫區域解的情況，若使用的突變強度較大，雖然可以跳脫目標函數為的區域解，但其整體的搜尋效率會變的相對較差。

為了改善傳統的演化策略的搜尋效率，本研究試圖提出一套新的演化策略搜尋法概念，主張演化時應考量母體週遭環境的概念導入其中，藉由演化策略中隨機突變子代的操作，將隨機突變的個別子代與母體之間的方向與距離當作母體的梯度向量，再以母體梯度向量判斷子代的演化距離與方向的運算機制。藉由這樣的機制使整體族群在搜尋目標函數時能快速且

有效的得到最佳解所在位置的資訊並向最佳解附近的區域快速的逼近，以達到快速收斂的目的。

1.3 研究流程與架構

本研究分為以下幾個步驟，研究的流程與架構如圖1-1：

1. 問題探討與確立目標

探討演化策略對於函數最佳解搜尋方法的問題，提出改善的方法。確立研究範圍與目標。

2. 文獻探討

藉由相關文獻的回顧，了解目前相關研究領域上的成果，作為研究過程的參考。

3. 研究方法

本章以實際建立新的演化策略模型和其演算方法，並詳細描述本研究所提出的演化策略模型之步驟。

4. 模式實作與結果分析

本研究以4個多變數函數範例來進行實驗驗證與實驗設計，實驗結果也將與近年來發表在國際期刊上且有優秀的搜尋效率的演算法一併做比較。

5. 結論與建議

本章為本研究論文之結論，並對本研究之建議及未來之研究方向作一概括描述。

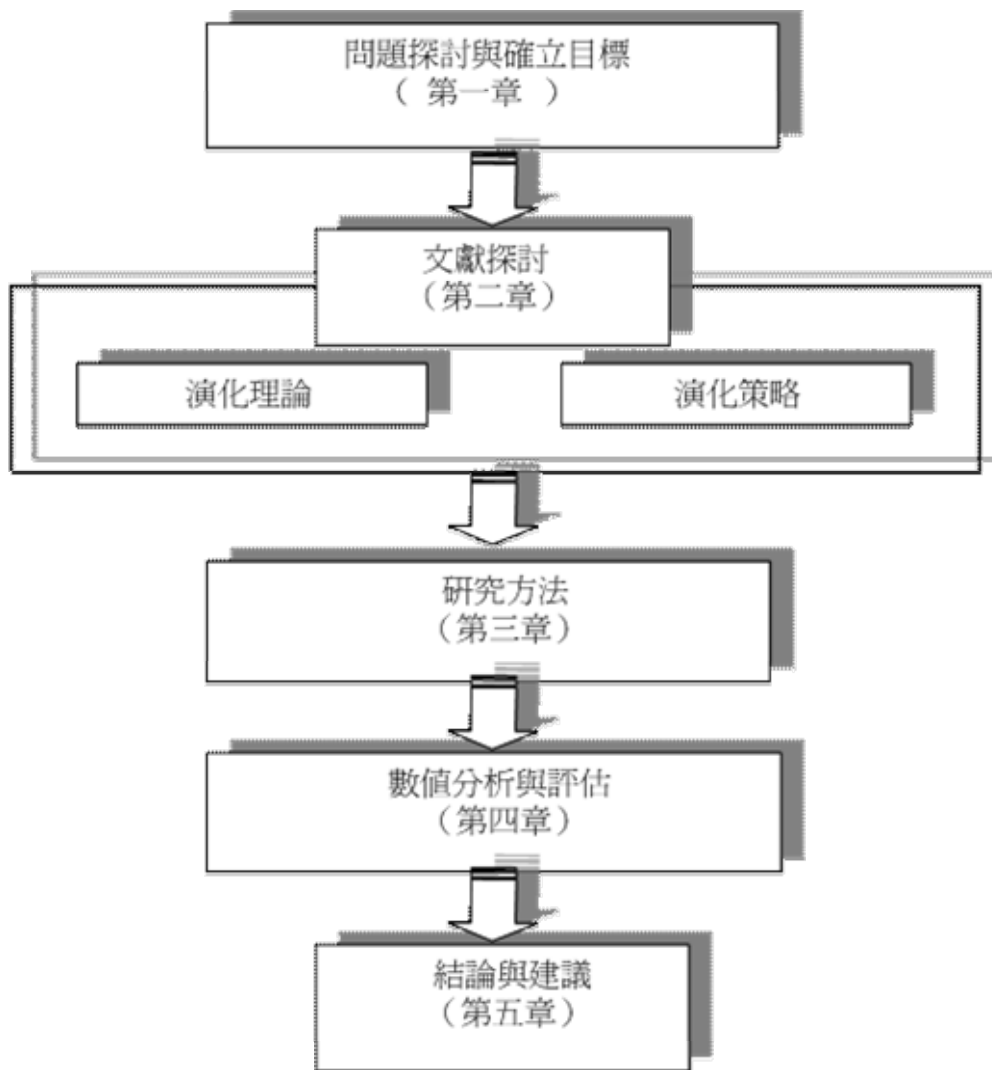


圖 1-1 研究方法與步驟

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。

第二章 文獻探討

2.1 演化理論(Evolutionary Algorithms)相關文獻

2.2.1 演化理論簡介

一般進行最佳化之大致流程，包括依據設計需求建立數學模型、選擇最佳化方法以及實際求解。自 1960 年代起，學者開始研究如何經由模仿物進化的模式，用來求解複雜之工程問題，進而發展出演化理論(Evolutionary algorithms; EAs)，包括由 Holland[33][34][35]提出之基因演算法(Genetic Algorithms; GAs)、Fogel[22][23]提出之演化規劃(Evolutionary programming; EP)、Rechenberg[42][43]與 Schwefel[45][47]提出之演化策略(Evolution strategies; ESs)、Koza[36][37]提出之遺傳規劃(Genetic programming; GP)以及 Dorigo et al.[15][16]與提出之蟻拓尋優法(Ant Colony Optimization; ACO)。

演化理論利用電腦模擬演化的現象，透過演化計算搜尋複雜問題的最佳解，目前被大量應用在搜尋(search)、最佳化(optimization)、排程(scheduling)等方面 [2][7][25][39][41][48]。

2.2.2 演化理論模型

許多學者對於演化理論的分類方法有不同的意見，基本上演化理論的模型可以分為以下幾種：

- 基因演算法(Genetic Algorithms; GAs)

基因演算法是由 John Holland[33][34][35]所提出的一般性最佳化演算法則，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)。

基因演算法的主要操作(operator)為基因交換(crossover)、天擇(selection)與突變(mutation)。天擇主要係依據母代(Parent)中每一染色體之適應值來決定子代(offspring)中，哪些染色體該被淘汰或複製及保留至下一代的一種運算操作，常見的方法有輪盤法(roulette wheel)、競賽法(tournament selection)、比例法(proportionate selection)、排序法(ranking selection)[28]。

基因交換的目的是在整個族群中製造差異，藉以產生優良且具適應能力的下一代，常見的基因交換方式有單點交換(One-point crossover)、尾尾交換(Tail-tail crossover)、首尾交換(Head-tail crossover)、複合式交換(Multiple crossover)、均勻交換(Uniform crossover)[1][51]。突變是為了增加族群個體的差異性而使用的一種基本運算子，藉由突變過程的變動運作，增加更多新的染色體進入搜尋空間中，以避免過早收斂至局部最佳解。

除了傳統的GAs模型外，近來也發展出一些特殊的GAs模型，如：Darrell Whitley提出的Genitor[10][11][12]、Eshelman提出的CHC[20][21]、Goldberg提出的Messy GAs[27][29]、Cantù-Paz提出的Island based GAs[6]等。

- 基因規劃(Genetic Programming; GP)

基因規劃是由 Koza[36][37]所發展出的演化方法，遺傳規劃衍生自基因演算法，發展基因規劃的主要目的是希望電腦程式(computer program)可以自我學習解決問題的能力。基因規劃與基因演算法最大的不同是將基因演算法的演化單位從固定長度的二元字串(binary string)擴充成可以變動大小和形狀的分析樹(parse tree)，用以代表電腦程式、函數或交易策略，且修改了基因交換(crossover)與突變(mutation)的方法以配合程式演化[38]。

- 演化策略(Evolution Strategies; ESs)

演化策略是 Rechenberg[42][43]及 Schwefel[45][47]發展出來的演化模型。演化策略與基因演算法最大的差異在於演化策略著重的是族群個體表現型態(phenotype)的變化，基因演算法著重的是個體基因型態(genotype)的擾動。

演化策略主要的演化方法為天擇(selection)、突變(mutation)與重構(recombination)。最常見的天擇方法為 (μ, λ) [45]的型態，Schwefel and Rudolph[50]提出 (μ, κ, λ) -ES 架構， κ 代表演化代數，若母代存活代數大於 κ 則保留母體， $1 \leq \kappa \leq \infty$ ，若 $\kappa = 1$ ，則天擇方法會成為 (μ, λ) -ES 的型態；若 $\kappa = \infty$ ，則天擇方法會成為 $(\mu + \lambda)$ -ES 的型態。為了使演化策略的演化更具有彈性，針對突變強度(mutation strength)的調整而發展出自我適應能力的演化策略(self-adaptation evolution strategies)，使用學習參數 τ_0 以及突變強度(mutation strength)的更新規則增強演化效率。此外，為了在整個族群中製造差異，藉以產生優良且具適應能力的下一代，發展出類似基因交換(crossover)

的操作方式，稱為重構。

- 演化規劃(Evolutionary Programming; EP)

演化規劃是 L. Fogel 於 1960 年代為了發展出可以作到自我演化的人工智能所提出的演化模型[24]。Fogel 認為人工智能需要能預測環境的變化並作出適當的改變以接近目標，因此他將環境當作一連串的符號(symbol)，找出預測環境的機制用以產生出下一個相關環境的符號。

演化規劃對於族群中的個體利用有限狀態機(finite state machine)當作演化的單元，每一次對環境的預測都給訂一個 pay off function 用以評斷個體演化的好壞。在演化操作方面，演化規劃並不使用重構的方法，而是每一個母體經由突變產生出單一的子代，主要的突變作法有：改變輸出符號(change of an output symbol)、改變轉換狀態(change of a state transition)、增加狀態(addition of a state)、刪除狀態(deletion of a state)以及改變初始狀態(change of the initial state)；在天擇的方法方面，演化規劃可以視為($\mu + \mu$)形式的天擇，由於每一個母體產生出單一的子代，再從母體與子代中選擇出其中的一半做為存活的個體。

- 蟻拓尋優法(Ant Colony Optimization; ACO)

蟻群系統(Ant system)為 Dorigo et al.[17]提出。蟻群系統乃是根據自然界中螞蟻覓食的行為模式所發展出來的演算法，其最大的特色為人造螞蟻(artificial ant)根據路徑上的費洛蒙(pheromone)多寡來搜尋求解組合最佳化的問題。

蟻群尋優法在 Colorni et al.[8][9]提出，稱之為蟻群系統 (Ant System)。是藉由自然界螞蟻尋找食物的精神發展出一套演算法則，並求解蟻群系統驗證旅行銷售員(traveling sales man ; TSP)問題。Dorigo and Maria[17]提出改良於蟻群系統的蟻拓法(Ant Colony System; ACS)，蟻拓法被廣泛被使用於求解各種最佳化問題。Dorigo et al.[16]。將蟻群系統、蟻拓法與相關應用的法則歸納成一套啟發式演算法 Ant Colony Optimization Meta-Heuristic。以這套演算法為精神的統稱為蟻拓尋優法(Ant Colony Optimization; ACO)。

2.2 演化策略(Evolution Strategies; ESs)

2.2.1 演化策略簡介

演化策略是 Rechenberg[42][43]及 Schwefel[45][47]於 1960 年代所發展

出來的模擬演化模型。演化策略與二位元基因演算法(Binary GAs)之間最大的不同在於演化策略的操作使用實數參數(real parameter value)，以及初期的演化策略只作天擇(selection)的操作而不使用基因交換相關的操作方式。演化策略是從母代(parent)以一固定的突變強度(mutation strength)產生子代(offspring)後再作天擇，而為了使演化策略的演化更有彈性，發展出自我適應能力的演化策略(self-adaptation evolution strategies)，近來演化策略發展出一些類似基因交換的操作方式，稱為重構(recombination)，主要目的在整個族群中製造差異，藉以產生優良且具適應能力的下一代，因此，對於演化策略的討論主要可以分為天擇、自我適應能力與重構三個部分。

2.2.2 天擇(selection)

最初的演化策略的結構為(1+1)-ES[42][44]，由一個母代使用高斯突變操作(Gaussian mutation operator)產生一個子代，在母代與子代中選取較佳的值成為新的母代。

Rechenberg[42]針對(1+1)-ES 模型中 σ 最佳化提出兩個模型：

- 線性方程 $f(x) = c_0 + c_1 x_1, \quad -b/2 \leq x_i \leq b/2, \quad \forall i = 1, 2, \dots, n$

$$\text{最佳化的 } \sigma \text{ 為 } \sigma^* = \sqrt{\frac{\pi b}{2 n}} \quad (1)$$

- 球體模型(sphere model) $f(x) = \sum_{i=1}^n x_i^2$

$$\text{最佳化的 } \sigma \text{ 為 } \sigma^* = 1.224 \frac{r}{n} \quad (2)$$

Rechenberg[42]在 1973 年提出收斂速度(convergent rate)的理論：1/5 success rule，認為突變強度 σ 應隨著演化代數被修正。Schwefel[45]提出關於 σ 的更新規則，在計算 n 次試驗中成功的突變數目 p_s ，建議 $c_d = 0.817$ ，則 σ 的更新規則為：

$$\sigma^{(t+1)} = \begin{cases} c_d \sigma^t, & \text{if } p_s < 1/5 \\ \frac{1}{c_d} \sigma^t, & \text{if } p_s > 1/5 \\ \sigma^t, & \text{if } p_s = 1/5 \end{cases} \quad (3)$$

(μ, λ)-ES 與 ($\mu + \lambda$)-ES[45]是演化策略中基本的兩種天擇(selection)的

形式， μ 代表母代(parent)的大小(size)， λ 代表每一代的演化產生的子代(offspring)數量，且 $\lambda > \mu$ 。 (μ, λ) -ES 與 $(\mu + \lambda)$ -ES 之間最大的差異點在於選取新的母代時， (μ, λ) -ES 是由 λ 個子代中選取最好的 μ 個作為新的母代； $(\mu + \lambda)$ -ES 是由 μ 個母代與 λ 個子代中選取最好的 μ 個作為新的母代。Schwefel 與 Rudolph[50]提出 (μ, κ, λ) -ES 架構， κ 代表演化代數，若母代存活代數大於 κ 則保留母體， $1 \leq \kappa \leq \infty$ 。若 $\kappa = 1$ ，則天擇方法(selection method)會成為 (μ, λ) -ES 的型態；若 $\kappa = \infty$ ，則天擇方法會成為 $(\mu + \lambda)$ -ES 的型態。Schwefel 與 Bäck (1998)[49]提出的建議為 $\kappa \geq 1$ ，用以允許生命的展延。

在重構(recombination)中常使用 $(\mu/\rho, \sigma)$ -ES[3]的結構作天擇方法，其中 ρ 代表作重構操作時，由 ρ 個母代重構出 λ 個子代，再由 μ 個母代與 λ 個子代中選取最好的 μ 個作為新的母代。

2.2.3 自我適應能力 (Self-Adaptation)

主要的自我適應能力的演化策略可以分為以下四類：

1. 等向的自我適應能力(Isotropic Self-Adaptation)

等向的自我適應能力主要使用一個學習參數 τ_0 以及突變強度的更新規則增強演化效率

$$\sigma^{(t+1)} = \sigma^{(t)} \exp(\tau_0 N(0,1)) \quad (4)$$

$$x_i^{(t+1)} = x_i^{(t)} + \sigma^{(t+1)} N_i(0,1) \quad (5)$$

Schwefel[46]在 1987 年提出學習參數 τ_0 最佳設定值應為： $\tau_0 \propto n^{-1/2}$ ， n 為變數的數量。Beyer[4]提出在 sphere model 中，對於 $(1, \lambda)$ -ES 最佳的學習參數 $\tau_0 = c_{1,\lambda} / \sqrt{n}$ ， $c_{1,\lambda}$ 為行進係數。

2. 非等向的自我適應能力(Non-Isotropic Self-Adaptation)

非等向的自我適應能力對於每個不同的變數使用不同的突變強度

$$\sigma_i^{(t+1)} = \sigma_i^{(t)} \exp(\tau' N(0,1) + \tau N_i(0,1)) \quad (6)$$

$$x^{(t+1)} = x^{(t)} + N(0,1) \quad (7)$$

where $\tau' \propto (2n)^{-1/2}$ and $\tau \propto (2n^{1/2})^{-1/2}$

Beyer[5]提出對於球體模型(sphere model)最佳的學習參數設定為 $\tau_0 \propto 1/\sqrt{n}$ ，可得到最大的收斂速率。

3. 關聯式自我適應能力(Correlated Self-Adaptation)

關聯式自我適應能力對於每個不同的變數使用不同的突變強度，因此最多存在 $\binom{n}{2}$ 個共變數，而共有 $\left(\binom{n}{2} + n\right)$ 個外部的策略參數(exogenous strategy parameter)需作更新與調整，關聯式自我適應能力的更新規則為在 n 個變數、 n 個突變強度 σ_i 與 $\binom{n}{2}$ 個 α_i 的旋轉角度的設定下，Schwefel[46]與 Bäck[2]提出參數 β_α 固定為 0.0873(或 5°)。

$$\sigma_i^{(t+1)} = \sigma_i^{(t)} \exp(\tau' N(0,1) + \tau N_i(0,1)) \quad (8)$$

$$\alpha_i^{(t+1)} = \alpha_i^{(t)} + \beta_\alpha N_j(0,1) \quad (9)$$

$$x^{(t+1)} = x^{(t)} + N(0, C(\sigma^{(t+1)}, \alpha^{(t+1)})) \quad (10)$$

4. CMA-ES

CMA(The Covariance Matrix Adaptation) 由 Hansen and Ostermeier[30] 於 1996 年提出，以 Covariance matrix 記錄整個族群數代演進的歷史，計算出目標變數變異數與共變數，而搜尋成果受變異數的影響。

$$x_k^{(g+1)} \sim N(m^{(g)}, \sigma^{(g)^2} C^{(g)}) \quad \text{for } k=1, \dots, \lambda \quad (11)$$

where $x_k^{(g+1)} \in \mathcal{R}^n$, k -th offspring of generation g .

$m^{(g)} \in \mathcal{R}^n$, mean value of the search distribution.

$C^{(g)} \in \mathcal{R}^{n \times n}$, covariance matrix.

Hansen and Ostermeier[31]在接受 Ingo Rechenberg(1998)的建議，從 λ 個子代中以權重重構(weighted recombination)的方法找出最佳的 μ ，將 CMA 改寫為 (μ_w, λ) -CMA_ES。

2.2.4 重構(recombination)

演化策略的重構主要使用 $(\mu/\rho + \lambda)$ [45]的天擇方法，基本的重構方法有：

(1) 中點型重構法(intermediate recombination)： $y = \frac{1}{\rho} \sum_{i=1}^{\rho} x^{(i)}$, $1 \leq \rho \leq \mu$

(2) 離散型重構法(discrete recombination)：從 ρ 個母體中 ($1 \leq \rho \leq \mu$, $\rho \in N$) 隨機選取變數值重構出一新的子代。

近來有些學者使用實參數基因演算法(real-parameter genetic algorithms)的重構方法(recombination operator)，例如 Eshelman 與 Schaffer 提出的 BLX(blend crossover)[21][26]、Deb 與 Agrawal 提出的 SBX(simulated binary crossover)[13]、Ono 與 Kobayashi 提出的 UNDX(unimodal normally distributed crossover)[40]、Higuchi 提出的 SPX(simplex crossover)[32]、Deb 提出的 PCX(parent-centric recombination)[14]。

第三章 研究方法

傳統的演化策略由於使用固定的突變強度(mutation strength) σ ，因此在搜尋目標函數最佳解時，搜尋時的效率取決於 σ 的大小，因此，若演化策略作目標函數最佳解的搜尋時若使用較小的 σ ，在細部搜尋時會較有效率，但若母體落在目標函數平坦區域時其演化效率相對較差，更重要的是，當目標函數為有多重區域(local)解時，若 σ 過小會造成無法跳脫區域解的情況，若使用的 σ 較大，雖然可以跳脫目標函數為的區域解，但其整體的搜尋效率會變的相對較差，特別是在靠近最佳解區域作細部搜尋時不易快速的逼近最佳解。自我適應能力的演化策略的發展主要在強調當母體在最佳解區域的細部搜尋能力， σ 隨著其演化歷程的資訊而不斷調整，為了能快速的搜尋最佳解， σ 會縮小到一定的程度以求快速的收斂，然而當目標函數為有多重區域(local)解時，過小的 σ 會造成無法跳脫區域解的情況，因此，自我適應能力的演化策略雖然對單一區域解的函數有非常優秀的搜尋效率，但無法對有多重區域(local)解的函數作全域(global)搜尋。

因此，本研究提出的類梯度搜尋演算策略方法(Quasi Gradient Search Evolution Strategy Method; QGSES)主要重點在於母體搜尋最佳解時會應先將周圍的資訊先比較過後再決定子代突變的方向與距離，而非演化策略中子代個別隨機突變的運作方式，藉由這樣的機制使整體族群在搜尋時能快速且有效的得到最佳點所在位置的資訊並向附近的區域快速的包圍住，以達到快速收斂的目的。

3.1 QGSES 原理

本研究的主要方向在於改善演化策略在搜尋時演化歷程在突變強度的限制下，子代的演化距離受限於其突變強度的侷限，當母體落在目標函數平坦區域時，無法快速的朝最佳解的方向逼近，以及在固定 σ 的限制下其細部搜尋的無效率。QGSES 主要在母體搜尋最佳解時會應先將周圍的資訊先比較過後再決定子代突變的方向與距離，而非演化策略中子代個別隨機突變的運作方式，藉由這樣的機制使整體族群在搜尋時能快速且有效的得到最佳點所在位置的資訊並向附近的區域快速的包圍住，以達到快速收斂的目的。

因此，本研究在傳統的演化策略中加入新的策略參數，此策略參數為

移動參數 ξ ，根據母體隨機突變的子代判斷母體梯度向量，再計算新的子代應該跳躍的距離與方向，當母體落在目標函數平坦區域時，可以作有方向性的跨越，縮短母體落在目標函數平坦區域的演化歷程，因此，可以避免演化策略在目標函數的平坦區域搜尋時的無效率，可以使母體快速的往最佳解區域逼近，此外，當母體在最佳解區域時由於梯度相對較大，在移動參數 ξ 的調整下，子代的移動除了方向的調整外，其移動距離也會相對小於固定的 σ ，因此在最佳解區域的細部搜尋上不會耗費過多的時間去演化最佳解。

QGSES 主要的核心在於根據母體的梯度向量調整移動參數 ξ ，根據母體的梯度向量與母體歷史移動向量判斷子代往最佳解區域逼近的最佳方向，加速演化策略整體的演化速度，進一步提升演化策略的搜尋效率。

3.2 QGSES 系統架構

本研究所發展的 QGSES 演算法是建立在 Schwefel 與 Bäck 所提出的 Multi-Memberes ES: $(\mu + \lambda)$ -ES 型態的演化策略。在一般 $(\mu + \lambda)$ -ES 型態的演化策略中，其演算法步驟如下：

1. Initialize population

隨機產生 μ 個初始母體。

2. Reproduction

由 μ 個母體複製出 λ 個子代。

3. Evolutionary operation : mutation or recombination

主要目的在整個族群中製造差異，經由突變或重構的操作，以產生優良且具適應能力的下一代。

4. Evaluation

對 μ 個母體與 λ 個子代做適合度評估。

5. Competition and selection

從母體與子代中(共 $\mu + \lambda$ 個個體)，選出 μ 個最好的個體存活到下一代，並捨棄其餘個體。

6. Ending

假如有可接受的解出現或其他停止演化條件發生，則中止演算法，否則回到步驟 2。

圖 3-1 為 QGSES 的架構圖，QGSES 演算法步驟如下：

1. Initialize population
隨機產生 μ 個初始母體。
2. Reproduction
由 μ 個母體複製出 λ 個子代。
3. Evolutionary operation
 - (1) 經由 mutation 的操作，產生 λ 個隨機的子代。
 - (2) 根據母體隨機突變的子代判斷母體梯度向量，再經由母體的梯度資訊與母體歷史的移動向量的綜合判斷，決定出新的子代應該跳躍的方向，再以移動參數 ξ 決定移動的距離，產生出 λ 個移動的子代。
4. Evaluation
對 μ 個母體與 λ 個隨機突變的子代以及 λ 個移動的子代做適合度評估。
5. Competition and selection
從母體與子代中(共 $\mu + \lambda + \lambda$ 個個體)，選出 μ 個最好的個體存活到下一代，並捨棄其餘個體。
6. Ending
假如有可接受的解出現或其他停止演化條件發生，則中止演算法，否則回到步驟 2。

QGSES 與一般 $(\mu + \lambda)$ -ES 型態的演化策略最大的差異在於演化操作 (evolutionary operation) 的部份，QGSES 除了保留演化策略中突變的方法外，提出移動參數 ξ 的概念，根據母體隨機突變的子代判斷母體梯度向量，再經由母體的梯度資訊與母體歷史的移動向量的綜合判斷，決定出子代應該跳躍的方向，再以移動參數 ξ 決定移動的距離，以期可以更有效率的搜尋最佳解與快速的收斂。

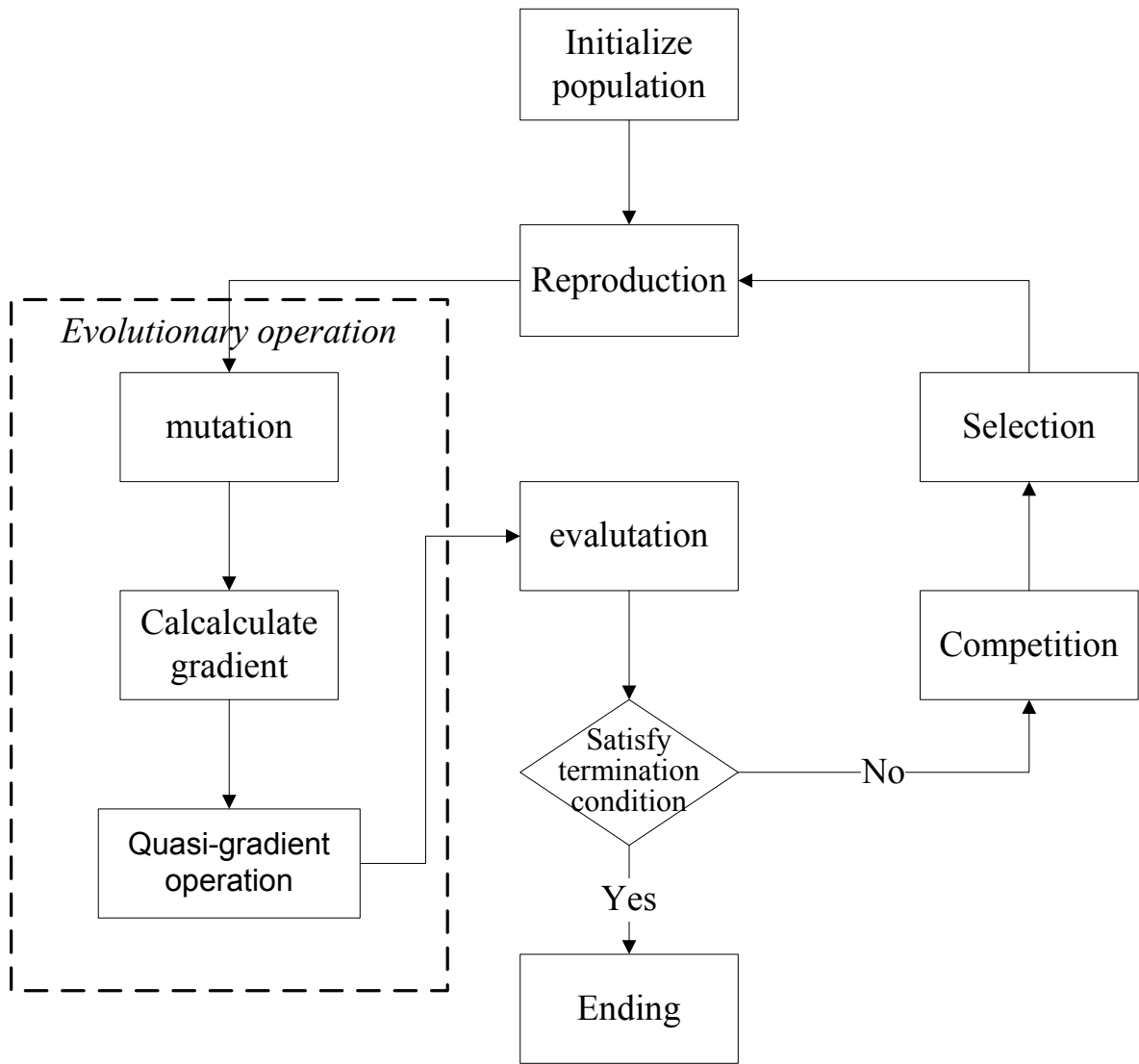


圖 3-1 QGSES 架構圖

圖 3-2 為 QGSES 演算法程序的虛擬碼(pseudo-code), $P_{t;l;\mu}$ 為第 t 代的 μ 個母體, $P'_{t;l;\lambda}$ 為由第 t 代的 μ 個母體經由突變操作產生的 λ 個子代, $P''_{t;l;\lambda}$ 為由第 t 代的 μ 個母體根據母體隨機突變的子代判斷母體梯度向量, 再經由母體的梯度資訊與母體歷史的移動向量的綜合判斷, 決定出新的子代應該跳躍的方向, 再以移動參數 ξ 決定移動的距離, 產生出 λ 個移動的子代, 再評估 $f(P'_{t;l;\lambda})$ 與 $f(P''_{t;l;\lambda})$ 是否滿足收斂條件, 若尚未滿足收斂條件, 則由 $P_{t;l;\mu}$ 、 $P'_{t;l;\lambda}$ 與 $P''_{t;l;\lambda}$ 經過競爭比較後選出最佳的 μ 個成為新的母體 $P_{t+1;l;\mu}$, 重複此步驟直到滿足收斂條件為止。

```

Begin QGSES
   $t := 0$ 
  Initialize population  $P_{t;l;\mu}$ 
  while not terminate do
    Evaluate  $P_{t;l;\mu}$ 
     $P'_{t;l;\lambda} := \text{MutationOperators}[P_{t;l;\mu}]$ 
     $P''_{t;l;\lambda} := \text{QGS-Operators}[P_{t;l;\mu}]$ 
     $P_{t+1;l;\mu} := \text{Competition}[P_{t;l;\mu} \cup P'_{t;l;\lambda} \cup P''_{t;l;\lambda}]$ 
     $t := t + 1$ 
  od
end QGSES

```

圖 3- 2 Pseudo-code of QGSES

3.3 QGSES 演算法

本研究所發展的 QGSES 演算法在天擇(selection)方法的選擇上，是建立在 Schwefel 與 Bäck 提出的建議，希望允許生命的展延，因此使用 Multi-Memberes ES: $(\mu + \lambda)$ -ES，且使用固定的突變強度 (mutation strength) σ 。

QGSES 演算法主要發展的重點在於母體搜尋最佳解時會應先將周圍的資訊先比較過後再決定子代突變的方向與距離，而非演化策略中子代個別隨機突變的運作方式，藉由這樣的機制使整體族群在搜尋時能快速且有效的得到最佳點所在位置的資訊並向附近的區域快速的包圍住，以達到快速收斂的目的。

此外，在有多重區域解的目標函數中，使用固定的突變強度有較好的跳脫能力，可以避免陷入區域解而無法跳出的困境，因此，QGSES 演算法擁有多重區域解函數的全域搜尋能力以及穩定的最佳解收斂效率。

3.3.1 QGSES 演算法符號說明表

表 3-1 符號說明表

符號	說明
$P_{t,i}$	第 t 代第 i 個母體
$V_{t,i,j}$	第 t 代第 i 個母體第 j 維度
$P'_{t,k}$	第 t 代第 k 個突變的子代
$V'_{t,k,j}$	第 t 代第 k 個突變的子代第 j 維度
$\alpha_{k,j}$	第 k 個突變的子代第 j 維度與母體的梯度向量
β	母體經過調整的子代的最大移動距離
$\xi_{k,j}$	母體經過調整的子代的移動距離
$P''_{t,k}$	第 t 代第 k 個移動的子代
$V''_{t,k,j}$	第 t 代第 k 個移動的子代第 j 維度
x_{opt}	最佳點

3.3.2 QGSES 演算法

本研究所發展的 QGSES 演算法是建立 Multi-Memberes ES: $(\mu + \lambda)$ -ES 型態的天擇方法，在固定的突變強度下，根據母體的梯度向量調整移動參數 ξ ，再根據母體的梯度向量與母體歷史移動向量判斷子代往最佳解區域逼近的最佳方向，提升演化策略的搜尋效率。

Step 1 : *Random initialize* $P_{t=0;1:\mu} \in [a_1, b_1] \times \cdots \times [a_n, b_n]$ *uniformly*

$$\forall P_{t,i} \in \mathbb{R}^n, P_{t,i} = [v_{t,i,1}, \cdots, v_{t,i,n}], i = 1, \cdots, \mu \quad (12)$$

$$\forall v_{t,i,j} \in \mathbb{R}^n, v_{t,i,j} = \{v_{t,i,j} \mid v_{t,i,j} \in [s_j, t_j], \forall s_j, t_j \in \mathbb{R}, j = 1, \cdots, n\} \quad (13)$$

$$\forall a_j \in \mathbb{R}, \sum_{j=1}^n a_j v_{t,i,j} = 0 \Leftrightarrow \forall a_j = 0 \quad (14)$$

在 step 1 中，在 $[a_1, b_1] \times \cdots \times [a_n, b_n]$ 的空間中以均勻隨機的方式初始 μ

個母體，母體的各個維度(dimension)互相獨立(independent)。

Step 2 : *Select* $\sigma > 0$

$$s.t. P'_{t;1:\lambda} = P_{t;1:\mu} + N(0, \sigma), \lambda \geq \mu \quad (15)$$

$$i.e. P'_{t;k} = [v'_{t;k;1}, \dots, v'_{t;k;n}], v'_{t;k;j} = \{v'_{t;k;j} \mid v'_{t;k;j} = v_{t;k;j} + N_j(0, \sigma)\} \quad (16)$$

where $k = 1, \dots, \lambda$

在 step 2 中，首先選擇一個固定的突變強度 σ ，母體以加上平均數為 0、標準差為 σ 的隨機常態分配的方式產生 λ 個子代。

Step 3 : $\exists P'_{t;k} \in P'_{t;1:\lambda}, P'_{t;k} = P_{t;i} + N(0, \sigma), 1 \leq k \leq \lambda, 1 \leq i \leq \mu$

$$\alpha_{k;j} = \frac{f(P'_{t;k}) - f(P_{t;i})}{v'_{t;k;j} - v_{t;i;j}} \approx \frac{\partial f(P_{t;i})}{\partial v_{t;i;j}} \quad (17)$$

在 step 3 中，計算隨機突變產生的子代與母體之間位移與適應值的差距，此差距即相當類似母體梯度的向量，並以此代替母體梯度向量。

Step 4 : *Define* $\beta = \tau\sigma, \tau > 1$ (18)

定義母體落在平坦區時，子代最大的移動距離 β 。

Step 5 : $\xi_{k;j} = \frac{1}{(1 + |\alpha_{k;j}|)^n} \beta, n > 0$ (19)

定義移動參數 ξ 的計算方法。在 step 5 中，當母體落在較為平坦區域時，母體梯度趨近於 0，因此移動參數會趨近於最大移動距離 β ，用以作大幅度的跨越；當母體落在較為陡峭區域時，母體梯度趨近於無限大，因此移動參數會根據母體梯度調整而縮小，以利於作細部搜尋。

Step 6 : $P''_{t;1:\lambda} = P_{t;1:\mu} \oplus \xi$ (20)

$$i.e. P''_{t;k} = [v''_{t;k;1}, \dots, v''_{t;k;n}]$$

$$v''_{t;j} = \begin{cases} v_{t-1;i;j} + \xi & \text{if } v_{t-1;i;j} \geq 0 \text{ and } \alpha_{k;j} < 0 \\ v_{t-1;i;j} - \xi & \text{if } v_{t-1;i;j} < 0 \text{ and } \alpha_{k;j} \geq 0 \\ -\xi & \text{if } v_{t-1;i;j} \geq 0 \text{ and } \alpha_{k;j} \geq 0 \\ \xi & \text{if } v_{t-1;i;j} < 0 \text{ and } \alpha_{k;j} < 0 \end{cases} \quad (21)$$

where $t \geq 1, \forall v_{t=0;j} = \bar{0}$

在 step 6 中，母體根據移動參數 ξ 決定子代移動的距離，再根據母體

梯度向量與母體歷史的移動資訊判斷子代的移動方向，若梯度方向與母體的歷史移動方向同向，則子代在該維度的移動方向與距離為移動參數 ξ 加上母體的歷史移動方向與距離；若梯度方向與母體的歷史移動方向反向，則子代在該維度的移動方向與距離為移動參數 ξ ，方向為母體的梯度方向。

$$\text{Step 7: } P_{t+1;l;\mu} = f^{-1}(\min_1^{\mu}(f(P_{t;l;\mu} \cup P'_{t;l;\lambda} \cup P''_{t;l;\lambda}))) \quad (22)$$

在 step 7 中主要從母體與所有子代(共 $\mu + 2\lambda$ 個個體)中選出最佳的 μ 個個體作為新的母體。

Step 8: $\exists \varepsilon > 0$

$$\text{if } \exists f(P_{t+1;i}) \in f(\bigcup_1^{\mu} P_{t+1;i}), f(P_{t+1;i}) \leq f(x_{opt}) + \varepsilon, \text{ then } f(P_{t+1;i}) \approx f(x_{opt})$$

$$\text{if } \forall f(P_{t+1;i}) \in f(\bigcup_1^{\mu} P_{t+1;i}), f(P_{t+1;i}) > f(x_{opt}) + \varepsilon, \text{ then find } P_{t+2;l;\mu}$$

設定母體與最佳值之間的中止條件為母體與最佳值的適應值間的差距小於 ε ，若新的母體滿足中止條件，則停止程式運算；若新的母體尚未滿足中止條件，則回到 step 2 尋找下一代新的母體。

圖 3-3 為 QGSES(Quasi Gradient Search Evolution Strategy)與演化策略(Evolution Strategy)整體演化效率的比較示意圖，演化策略雖然可以穩定的往最佳解的方向逼近，然而子代的演化距離受限於其突變強度(mutation strength)的侷限，只能以連續型的方式逐漸的往最佳解區域前進，無法快速的朝最佳解的方向逼近，進而造成搜尋的時間增加。QGSES 在移動參數 ξ 與母代歷史移動向量的雙重考量下，產生出明確方向與距離的新子代，用以快速的往最佳解區域逼近，特別當母體落在目標函數的平坦區域時，可以作有方向性且大幅度的跨越，縮短母體落在目標函數平坦區域的演化歷程，有效的加強演化策略的演化效能。

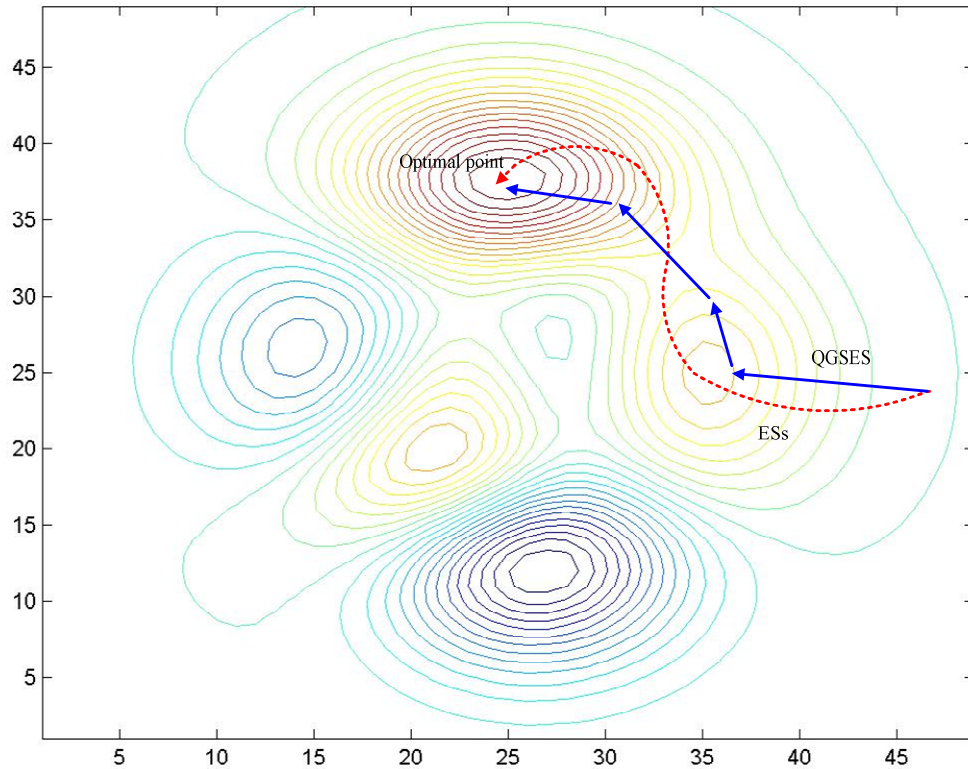


圖 3-3 QGSES 與演化策略往最佳解區域逼近的比較

圖 3-4 為 QGSES 在目標函數平坦區的示意圖，母體 P_{t+1} 先作隨機突變產生子代 P'_{t+1} ($V_{mut} : P_{t+1} \rightarrow P'_{t+1}$)，再計算母體梯度並決定移動參數 ξ ，再考慮母體之前的移動向量 ($V_t : P_t \rightarrow P_{t+1}$)，進而決定新子代的移動方向與距離 ($V_{move} : P_{t+1} \rightarrow P''_{t+1}$)。在移動參數 ξ 的調整下，QGSES 在目標函數平坦區時所產生的移動向量會大於設定的突變強度 (Mutation strength) 所產生的隨機位移向量，此外，移動參數 ξ 經由母體的梯度資訊與母體之前的移動向量的綜合判斷，決定出明確的移動方向，因此，QGSES 在平坦區域得演化效率會遠優於演化策略，可以大幅度且快速的跨越目標函數的平坦區域，節省演化策略在目標函數的平坦區域中許多的演化歷程。

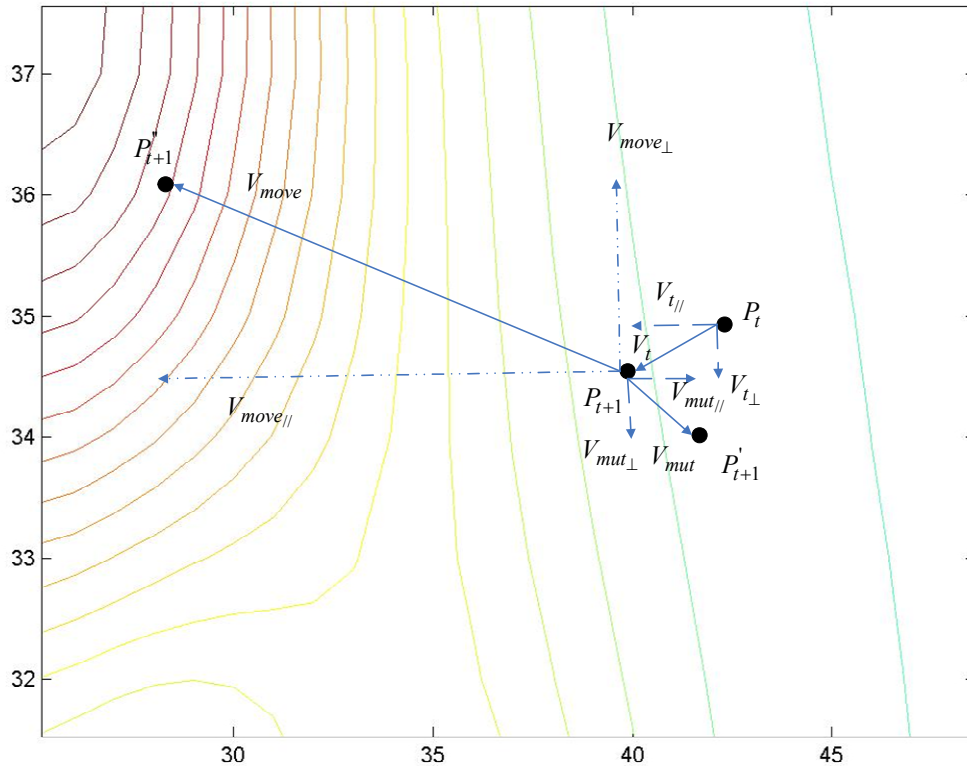


圖 3-4 QGSES 在目標函數平坦區域的移動向量

圖 3-5 為 QGSES 在目標函數最佳解區域的示意圖，QGSES 在移動參數 ξ 的調整下，在目標函數最佳解區域時所產生的移動向量會小於設定的突變強度所產生的隨機位移向量，由於移動參數 ξ 決定出明確的移動方向，因此，QGSES 在目標函數最佳解區域除了隨機突變的搜尋能力之外，加上移動參數 ξ 產生明確方向與距離的移動向量的搜尋，更可以有效率的搜尋最佳解與快速的收斂，這樣的演算機制必能快速且有效率的找到最佳值。

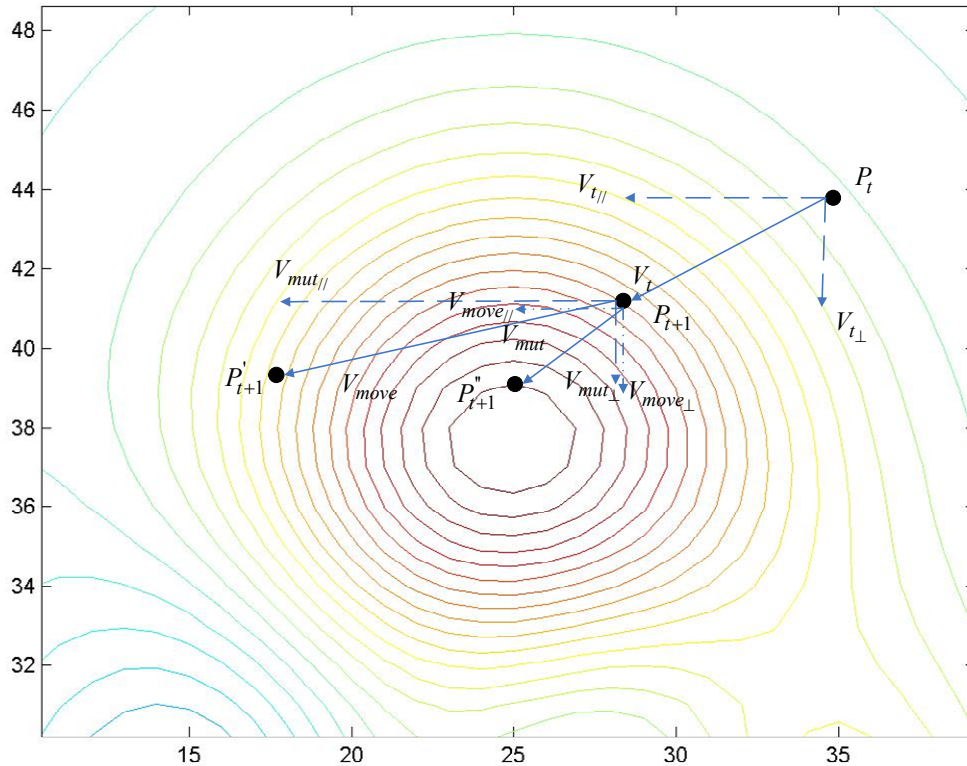


圖 3-5 QGSES 在目標函數最佳解區域的移動向量

3.4 QGSES 與其他演算法的結構比較

演化理論中各個模型之間的主要差異大都分為三個部份：天擇 (selection)、複製(reproduction)、演化操作(evolutionary operation)。表 3-2 為 QGSES 與其他具代表性的演算法的架構作綜合的比較與探討，包括 $(\mu + \lambda)$ 型態的演化策略、Y-L 所提出的架構、CMA-ES 及 G3PCX。

表 3-2 QGSES 與其他演算法的結構比較

方法	複製	演化操作	天擇
QGSES	$(\mu + \lambda)$ 型態：由 μ 個母體複製出 λ 個子代。	<ol style="list-style-type: none"> 1. 突變：經由突變的操作，以產生優良且具適應能力的下一代。 2. 根據母體隨機突變的子代判斷母體梯度向量，再經由母體的梯度資訊與母體歷史的移動向量的綜合判斷，決定出新的子代應該跳躍的方向，再以移動參數 ξ 決定移動的距離，產生出 λ 個移動的子代。 	$(\mu + \lambda)$ 型態：從母體與子代中 (共 $\mu + \lambda + \lambda$ 個個體)，選出 μ 個最好的個體存活到下一代，並捨棄其餘個體。
$(\mu + \lambda)$ 型態的演化策略	$(\mu + \lambda)$ 型態：由 μ 個母體複製出 λ 個子代。	突變：經由突變的操作，以產生優良且具適應能力的下一代。	$(\mu + \lambda)$ 型態：從母體與子代中 (共 $\mu + \lambda$ 個個體)，選出 μ 個最好的個體存活到下一代，並捨棄其餘個體。
Y-L 所提出的架構	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。	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。	菁英化政策：保留部分適應值較佳的個體到下一代。
CMA-ES	Weight multi-recombination：以 Covariance matrix 計算出目標變數變異數與共變數，以權重的方式重構出新的母體。	突變：CMA 以 Covariance matrix 記錄整個族群數代演進的歷史，計算出目標變數變異數與共變數，藉此縮小突變強度，達到快速收斂的目的。	—
G3PCX	Generalized Generation Gap (G3) Model：從母體中選擇最佳母體，在從其他的母體中隨機選擇兩個母體。	PCX (Parent-Centric Recombination)：利用 G3 Model 選出的向量與最佳母體重構出新的子代。	Generalized Generation Gap (G3) Model：從隨機選擇兩個母體與產生的子代中選擇最好的兩個值。

資料來源：本研究整理

第四章 數值分析與評估

4.1 多變數函數分析

本章節將依第三章所提出的研究架構，以 4 個多變數函數範例來測試實驗結果，實驗結果也將與近年來發表在國際期刊上且有優秀的搜尋效率的演算法一併做比較，包括 Y-L[52][53]所提出的架構、Hansen and Ostermeier[31]提出的 (μ_w, λ) -CMA-ES 及 K. Deb, A. Anand and D. Joshi[14]提出的 G3PCX。另外，本研究將就 QGSES 演算法中 4 個重要的參數作實驗設計，包括：母體數目 μ 、子代數目 λ 、突變強度 σ 以及子代最大移動距離 β ，探討這些參數對 QGSES 演算法求解效率的影響程度。

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) 達收斂時的平均運算次數：在 50 次的測試中，將達收斂條件的所有運算次數的平均值。
- (2) 達收斂時的平均次數之標準差：在(1)中計算標準差。
- (3) 達收斂時之平均計算時間：在 50 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。
- (4) 達收斂時之平均計算時間之標準差：在(3)中計算標準差。
- (5) 達收斂時之最佳解平均值：將每次測試後的最好的值取平均值。
- (6) 達收斂時之最佳解標準差：將(5)計算標準差。
- (7) 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%。

4.1.1 函數 1：Colville function

(1) 目標函數

$$\text{Min. } F(X) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1)$$

where $x_i \in [0, 10]$

此函數目的在求此函數的最小值，變數數量為 4 個變數，此函數之理想(Ideal)最佳值為 0。

(2) 測試環境與參數設定

QGSES 演算法在 Colville function 之相關變數設定如下：

- a. 母體 μ 的數量設定：本研究架構的實驗中，由於使用 $(\mu + \lambda)$ 的架構，因此母體的大小直接影響每一代產生的子代數目。在本函數中母體 μ 的數量分別為 1、2、4、6、8、10。
- b. 子代 λ 的數量設定：本研究架構的實驗中，子代 λ 的數量設定直接影響每一代產生的子代數目，以及每一代移動的子代數量。在本函數中子代 λ 的數量為 1~10。
- c. 突變強度 σ ：本研究使用固定的突變強度。在本函數中初始的突變強度 σ 為 0.01，突變強度 σ 分別為 0.001、0.005、0.01、0.025、0.05、0.075、0.1。
- d. β 值：母體經過調整子代的最大移動距離，在本研究中設定 $\beta = \tau\sigma$, $\tau > 1$ 。在本函數中初始的 β 值為 1.5σ ， β 值的分別為 1.5σ 、 2σ 、 2.5σ 、 3σ 、 3.5σ 、 4σ 。
- e. 收斂條件：在本範例中，當與最佳值的誤差小於 10^{-3} ，則視為收斂。
- f. 測試次數：50 次。
- g. 最大演化代數：10000 代。
- h. 實驗設計：為了探討 μ 、 λ 、 β 與 σ 四個參數對 QGSES 演算法搜尋效率的影響，本研究的實驗設計是先固定 β 與 σ 兩個自變數，並同時改變 μ 與 λ 兩個應變數，探討 QGSES 在函數中不同的 μ 與 λ 對搜尋效率的影響，找出最佳的 μ 與 λ ，再將最佳的 μ 與 λ 以及 σ 設為自變數找出最佳的 β 值，最後再找出對應的最佳 σ 值。

(3) QGSES 實驗結果

圖 4-1 為 Colville function 在固定 $\beta = 1.5\sigma$ 與 $\sigma = 0.01$ 下，不同的 μ 與 λ 對 QGSES 演算法的搜尋效率的影響，詳細數據請參考附錄，由圖 4-1 可知在固定的 β 與 σ 設定下，最佳的 μ 與 λ 的參數設定值為 $\mu = 1$ 、 $\lambda = 2$ ，此時 QGSES 演算法有最佳的搜尋效率，收斂百分比為 100%，達收斂時的平均運算次數為 8230.44 次，達收斂時的平均次數之標準差為 3210.79727。

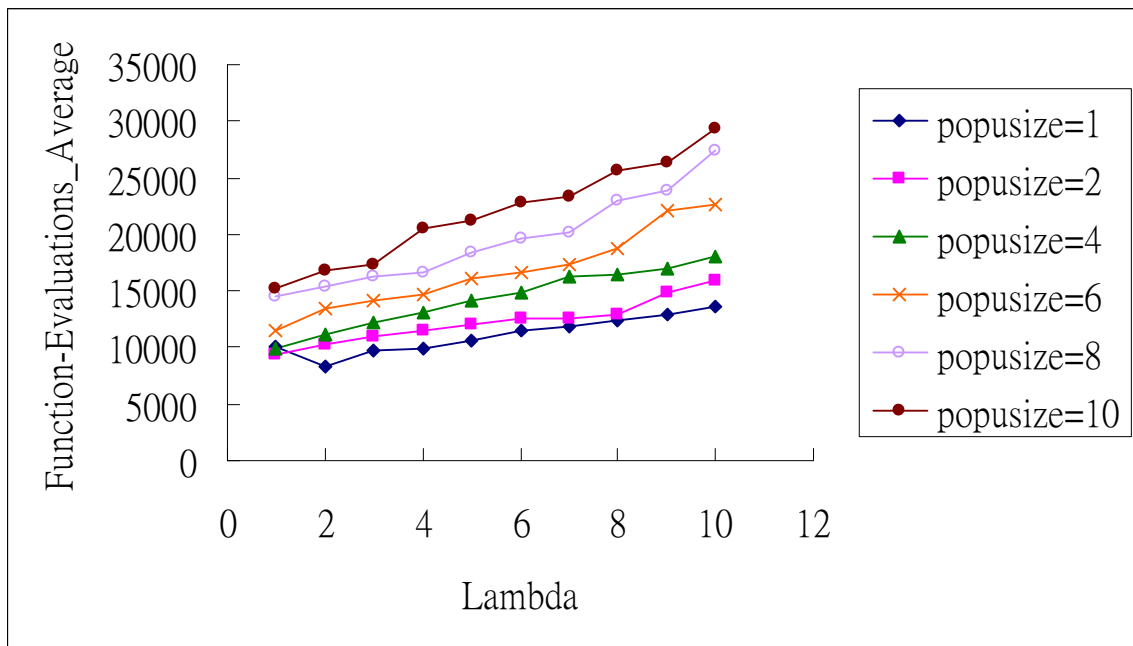


圖 4-1 μ 與 λ 對 QGSES 在 Colville function 中搜尋效率的影響

表 4-1 與圖 4-2 為 Colville function 在固定 $\mu=1$ 、 $\beta=1.5\sigma$ 與 $\sigma=0.01$ 下，不同的 λ 對 QGSES 演算法的搜尋效率的影響，在固定的 μ 、 β 與 σ 設定下，最佳的 λ 的參數設定值為 $\lambda=2$ ，此時 QGSES 演算法有最佳的搜尋效率。

表 4-1 應變數 λ 在 Colville function 中的實驗數據

Lambda	Best-Value		Time (sec.)		Function-Evaluations	
	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
1	0.000947788	9.39711E-05	1.79968	0.380784016	10017	2269.80499
2	0.000771765	0.00022469	1.15786	0.42761102	8230.44	3210.79727
3	0.000796931	0.000144846	1.2126	0.346543987	9751.72	2941.603804
4	0.000831962	0.000141314	1.17058	0.322659174	9925.64	2884.837248
5	0.000736777	0.000171571	1.20968	0.350735579	10670.2	3249.91544
6	0.000711813	0.000203543	1.25786	0.343841089	11415.4	3274.72727
7	0.000755458	0.000184707	1.28216	0.350357157	11868.24	3407.196631
8	0.0006983	0.000212049	1.32584	0.309135421	12393	3038.439557
9	0.000754865	0.00022064	1.35626	0.394146653	12863.08	3913.498293
10	0.000717553	0.000185482	1.42092	0.337792524	13529.8	3362.846608

備註：自變數 $\mu=1$ 、 $\sigma=0.01$ 、 $\beta=1.5\sigma$

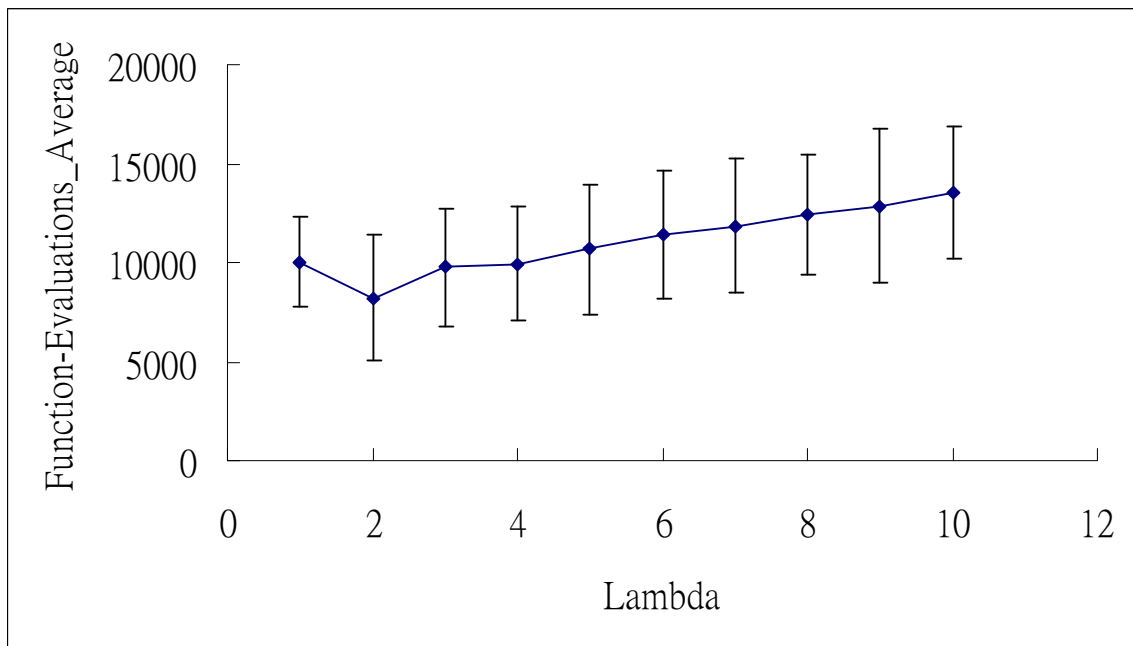


圖 4-2 λ 對 QGSES 在 Colville function 中搜尋效率的影響

表 4-2 與圖 4-3 為 Colville function 在固定 $\lambda=2$ 、 $\beta=1.5\sigma$ 與 $\sigma=0.01$ 下，不同的 μ 對 QGSES 演算法的搜尋效率的影響，在固定的 λ 、 β 與 σ 設定下，最佳的 μ 的參數設定值為 $\mu=1$ ，此時 QGSES 演算法有最佳的搜尋效率。

表 4-2 應變數 μ 在 Colville function 中的實驗數據

Population size	Best-Value		Time (sec.)		Function-Evaluations	
	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
1	0.000771765	0.00022469	1.15786	0.42761102	8230.44	3210.79727
2	0.000710533	0.000203005	1.16304	0.250293946	10317.52	2327.672695
4	0.000723829	0.000260327	1.0957	0.38263369	11102.24	4112.986157
6	0.000737783	0.000230142	1.25692	0.429032221	13427.76	4816.950219
8	0.000677111	0.000215647	1.39916	0.458575806	15339.2	5251.197632
10	0.000674641	0.000208466	1.49758	0.38130298	16706	4436.457495

備註：自變數 $\lambda=2$ 、 $\sigma=0.01$ 、 $\beta=1.5\sigma$

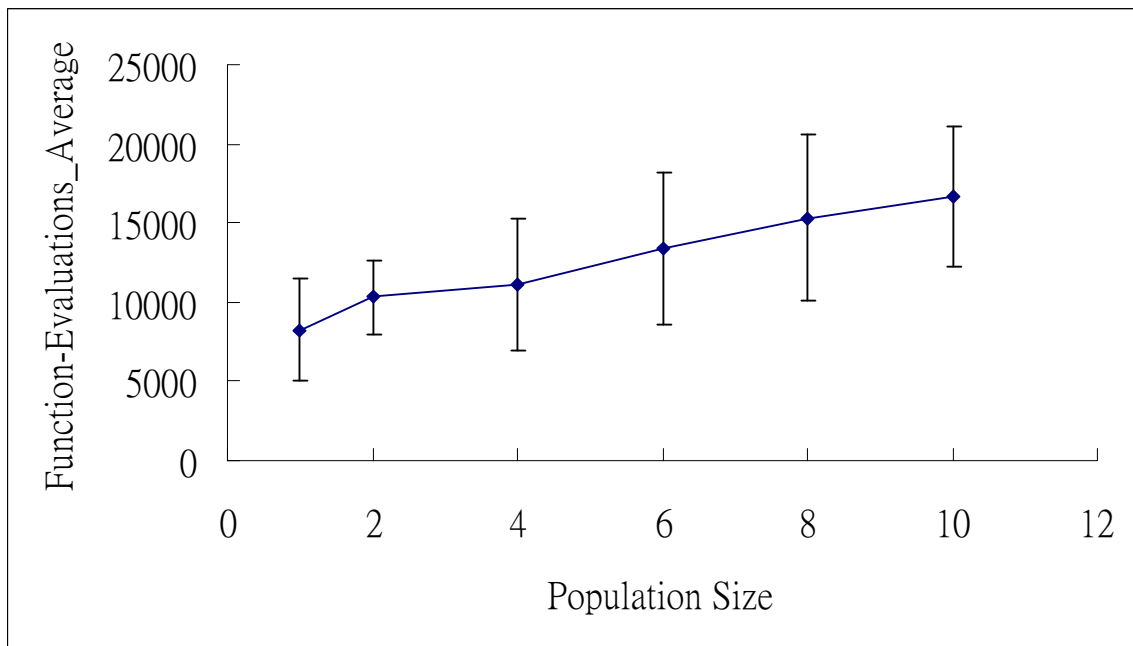


圖 4-3 μ 對 QGSES 在 Colville function 中搜尋效率的影響

表 4-3 與圖 4-4 為 Colville function 在最佳的 μ 與 λ 的設定下 ($\mu=1$, $\lambda=2$), 固定 $\sigma=0.01$, 測試不同的 β 值對 QGSES 演算法的搜尋效率的影響, 實驗結果顯示最佳的 β 的參數設定值為 $\beta=1.5\sigma$, 此時 QGSES 演算法有最佳的搜尋效率。

表 4-3 應變數 β 在 Colville function 中的實驗數據

Beta	Best-Value		Time (sec.)		Function-Evaluations	
	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
1.5σ	0.000771765	0.00022469	1.15786	0.42761102	8230.44	3210.79727
2σ	0.000855091	0.000131149	1.32622	0.473367238	9524.04	3607.417715
2.5σ	0.000811337	0.000166903	1.3947	0.473354863	10131.48	3574.12624
3σ	0.000804364	0.000208116	1.5512	0.467569006	11150.6	3553.824224
3.5σ	0.000840827	0.000123278	1.61564	0.461635724	11618.52	3442.920819
4σ	0.000839039	0.000163882	1.70036	0.561189399	12225.48	4173.596434

備註：自變數 $\mu=1$ 、 $\lambda=2$ 、 $\sigma=0.01$

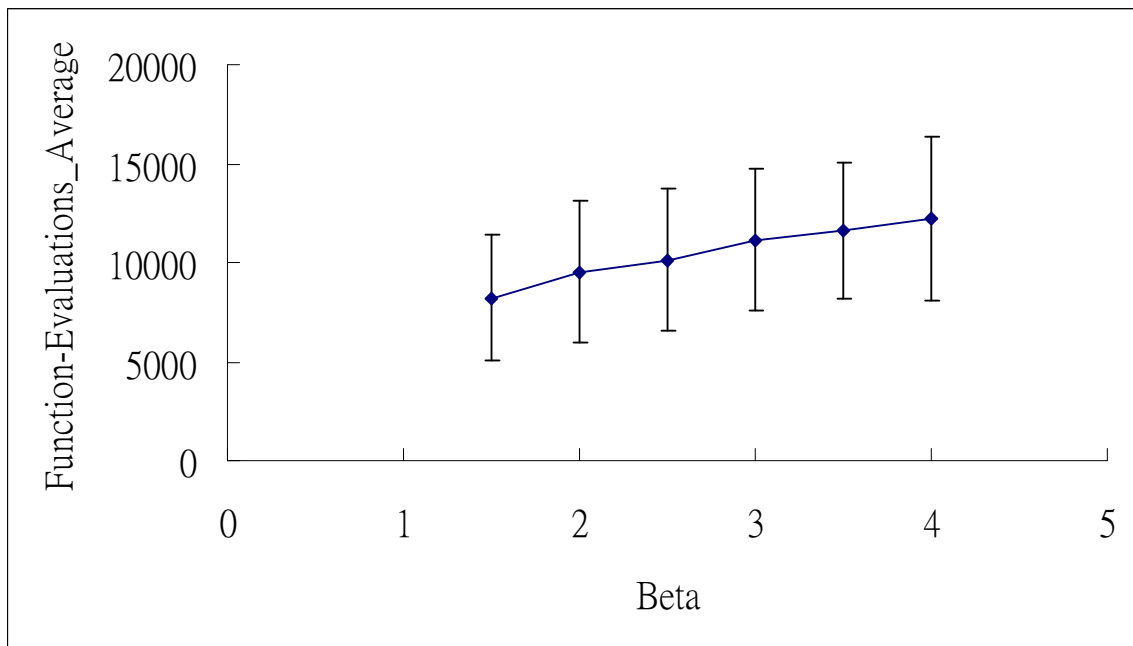


圖 4-4 β 對 QGSES 在 Colville function 中搜尋效率的影響

表 4-4 與圖 4-5 為 Colville function 在最佳的 μ 、 λ 與 β 的設定下 ($\mu=1$, $\lambda=2$, $\beta=1.5\sigma$)，測試不同的 σ 值對 QGSES 演算法的搜尋效率的影響，實驗結果顯示最佳的 σ 的參數設定值為 $\sigma=0.005$ ，此時 QGSES 演算法有最佳的搜尋效率。

表 4-4 應變數 σ 在 Colville function 中的實驗數據

Sigma	Best-Value		Time (sec.)		Function-Evaluations	
	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
0.001	0.23076153	1.574486845	3.2378	1.482601932	24333.48	11424.91015
0.005	0.00086257	0.000154465	1.12654	0.325836957	8087.32	2504.194045
0.01	0.000802062	0.000177052	1.17506	0.366400941	8464.36	2827.792488
0.025	0.00085955	0.000193248	1.59556	0.583520182	11689.56	4490.732335
0.05	0.00085618	0.000223149	1.74254	0.624376692	12829.96	4821.334056
0.075	0.000896677	0.000192778	1.8148	0.499612135	13332.36	3816.218567
0.1	0.000949853	0.000132111	1.9626	0.413478771	14544.52	3193.379646

備註：自變數 $\mu=1$ 、 $\lambda=2$ 、 $\beta=1.5\sigma$

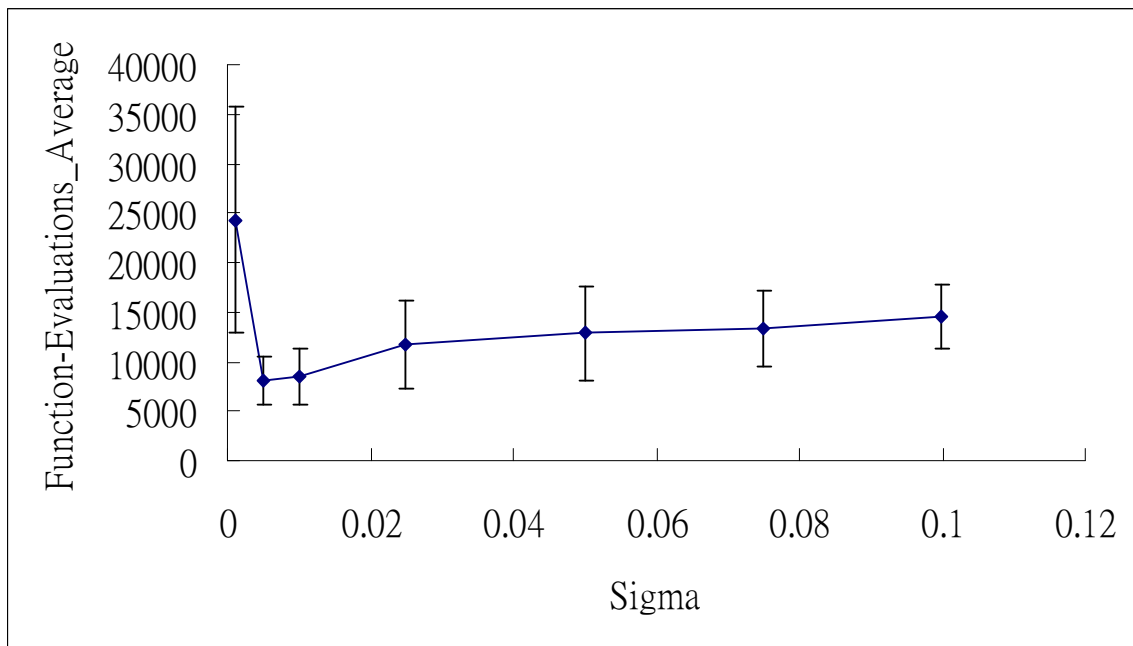


圖 4-5 σ 對 QGSES 在 Colville function 中搜尋效率的影響

從以上的數據顯示 QGSES 演算法在 Colville function 中最佳的參數設計為 $\mu=1$ 、 $\lambda=2$ 、 $\sigma=0.005$ 、 $\beta=1.5\sigma$ ，在最佳的參數設計下，此時 QGSES 演算法有最佳的搜尋效率，收斂百分比為 100%，達收斂時的平均運算次數為 8230.44 次，達收斂時的平均次數之標準差為 3210.79727。

(4) QGSES 與其他搜尋方法的比較

表 4-5 為 Colville function 在 QGSES、CMA 與 Y-L 三種方法的實驗結果，測試次數皆為 50 次的重複試驗。圖 4-6 為 Colville function 在 QGSES、CMA 與 Y-L 三種方法之平均運算次數比較圖。由表 4-5 中的實驗數據顯示，在單一最佳解的函數中，QGSES 的收斂效率雖然遠優於 Y-L 所提出的架構，然而 CMA 有最好的收斂效率，此外，根據最新的研究，Deb 提出的 G3PCX 在單一最佳解的函數中的收斂效率平均優於 CMA 約 15%。

表 4-5 不同的演算法對 Colville function 的測試結果

Method	Convergence Percentage	Best-Value		Time (sec.)		Function-Evaluations	
		Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
QGSES	100%	0.00086257	0.000154465	1.12654	0.325836957	8087.32	2504.194045
CMA	100%	0.000882641	0.000336576	0.36652	0.150949917	3513.92	1821.15132
Y-L	100%	0.000775943	0.000305029	208.90782	240.7234839	125612.16	71550.69673

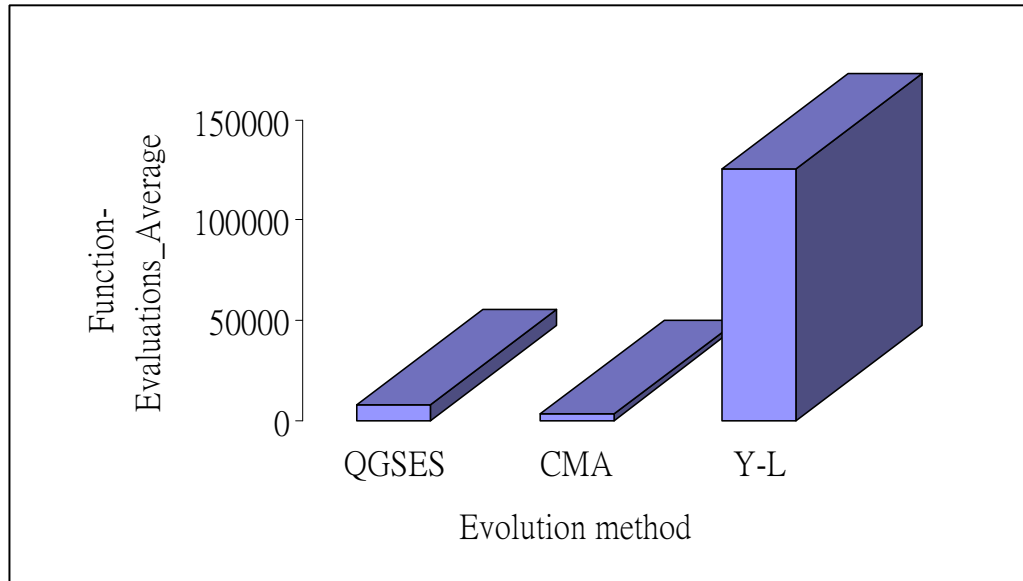


圖 4-6 Colville function 之平均運算次數比較圖

4.1.2 函數 2：Rosenbrock's function

(1) 目標函數

$$\text{Min. } F(X) = \sum_{i=1}^{n-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$$

where $x_i \in [0, 2]$

此函數目的在求此函數的最小值，變數數量為 10 個變數，此函數之理想(Ideal)最佳值為 0。

(2) 測試環境與參數設定

QGSES 演算法在 Rosenbrock's function 之相關變數設定如下：

- a. 母體 μ 的數量設定：在本函數中母體 μ 的數量分別為 1、2、4、6、8、10。
- b. 子代 λ 的數量設定：在本函數中子代 λ 的數量為 1~10。
- c. 突變強度 σ ：在本函數中初始的突變強度 σ 為 0.00005，突變強度 σ 分別為 0.000005、0.000001、0.00005、0.00001、0.0005、0.001、0.005。
- d. β 值：在本函數中初始的 β 值為 50σ ， β 值的分別為 20σ 、 30σ 、 40σ 、 50σ 、 60σ 、 70σ 、 80σ 。
- e. 收斂條件：在本範例中，當與最佳值的誤差小於 10^{-3} ，則視為收斂。
- f. 測試次數：50 次。
- g. 最大演化代數：10000 代。

(3) QGSES 實驗結果

圖 4-7 為 Rosenbrock's function 在固定 $\beta = 50\sigma$ 與 $\sigma = 0.00005$ 下，不同的 μ 與 λ 對 QGSES 演算法的搜尋效率的影響，詳細數據請參考附錄，由圖 4-7 可知在固定的 β 與 σ 設定下，最佳的 μ 與 λ 的參數設定值為 $\mu = 1$ 、 $\lambda = 1$ ，此時 QGSES 演算法有最佳的搜尋效率，收斂百分比為 100%，達收斂時的平均運算次數為 6501.64 次，達收斂時的平均次數之標準差為 2810.238117。

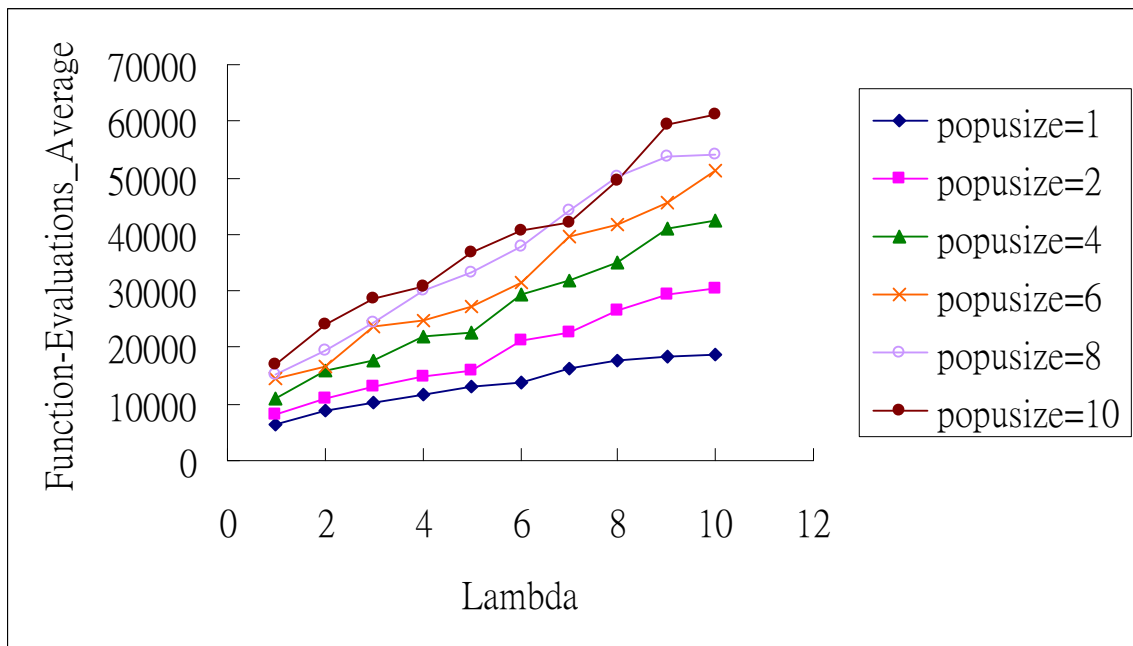


圖 4-7 μ 與 λ 對 QGSES 在 Rosenbrock's function 中搜尋效率的影響

表 4-6 與圖 4-8 為 Rosenbrock's function 在固定 $\mu=1$ 、 $\beta=50\sigma$ 與 $\sigma=0.00005$ 下，不同的 λ 對 QGSES 演算法的搜尋效率的影響，在固定的 μ 、 β 與 σ 設定下，最佳的 λ 的參數設定值為 $\lambda=1$ ，此時 QGSES 演算法有最佳的搜尋效率。

表 4-6 應變數 λ 在 Rosenbrock's function 中的實驗數據

Lambda	Best-Value		Time (sec.)		Function-Evaluations	
	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
1	0.000984487	2.27294E-05	1.83184	0.849032	6501.64	2810.238117
2	0.00097795	3.56274E-05	1.99402	0.868856	8738.92	3924.593081
3	0.000971176	6.13024E-05	2.17722	0.808838	10256.08	3935.228214
4	0.000984394	3.04606E-05	2.3934	1.038784	11696.84	5207.156517
5	0.000985504	2.81916E-05	2.58732	1.037571	13049.6	5362.442335
6	0.000981516	3.79001E-05	2.72534	1.283734	13846.6	6679.280861
7	0.000982889	2.86227E-05	3.17596	1.232174	16375.12	6460.656988
8	0.00097257	5.50339E-05	3.40404	1.304067	17749.48	6929.227993
9	0.000991212	1.35605E-05	3.50586	1.541168	18395.2	8228.238141
10	0.000988664	2.28873E-05	3.53818	1.707161	18665.8	9152.61226

備註：自變數 $\mu=1$ 、 $\sigma=0.00005$ 、 $\beta=50\sigma$

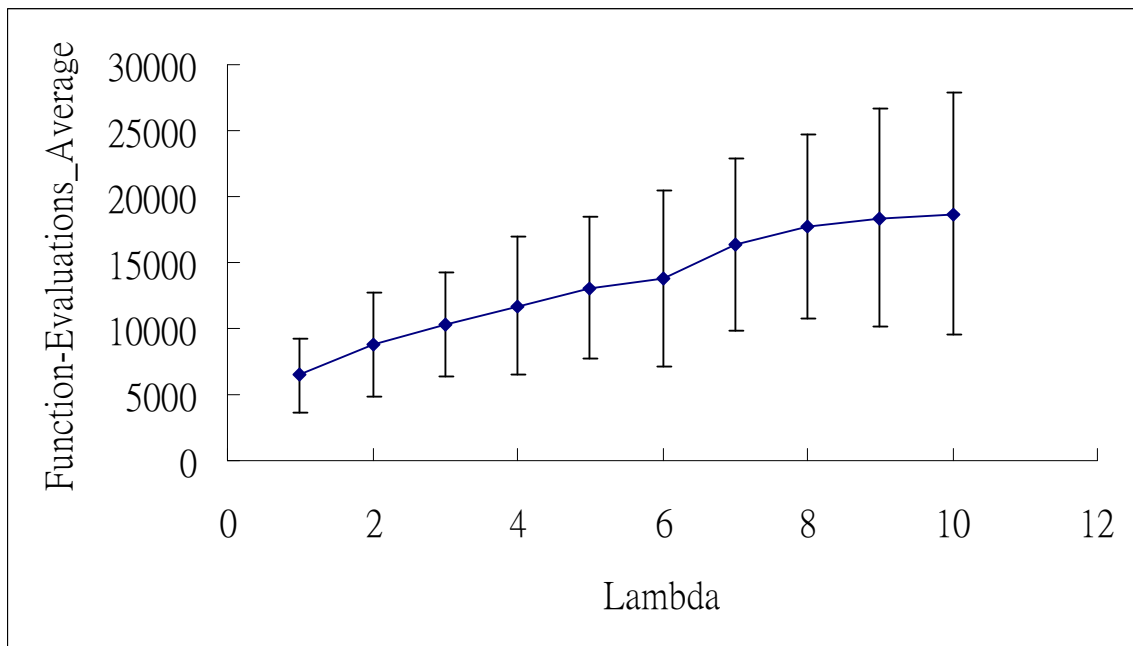


圖 4-8 λ 對 QGSES 在 Rosenbrock's function 中搜尋效率的影響

表 4-7 與圖 4-9 為 Rosenbrock's function 在固定 $\lambda=1$ 、 $\beta=50\sigma$ 與 $\sigma=0.00005$ 下，不同的 μ 對 QGSES 演算法的搜尋效率的影響，在固定的 λ 、 β 與 σ 設定下，最佳的 μ 的參數設定值為 $\mu=1$ ，此時 QGSES 演算法有最佳的搜尋效率。

表 4-7 應變數 μ 在 Rosenbrock's function 中的實驗數據

Population size	Best-Value		Time (sec.)		Function-Evaluations	
	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
1	0.000984487	2.27294E-05	1.83184	0.849032327	6501.64	2810.238117
2	0.000975804	4.57363E-05	1.78006	0.712544003	7960.24	3305.699139
4	0.000967509	4.57365E-05	2.1641	0.80506948	10971.36	4204.616577
6	0.000956462	5.83622E-05	2.70468	1.053989398	14563.44	5805.168127
8	0.000964476	4.33304E-05	2.80716	0.915832093	15306.56	5110.619325
10	0.000952569	5.55581E-05	3.03844	1.142614549	16898.8	6466.996477

備註：自變數 $\lambda=1$ 、 $\sigma=0.00005$ 、 $\beta=50\sigma$

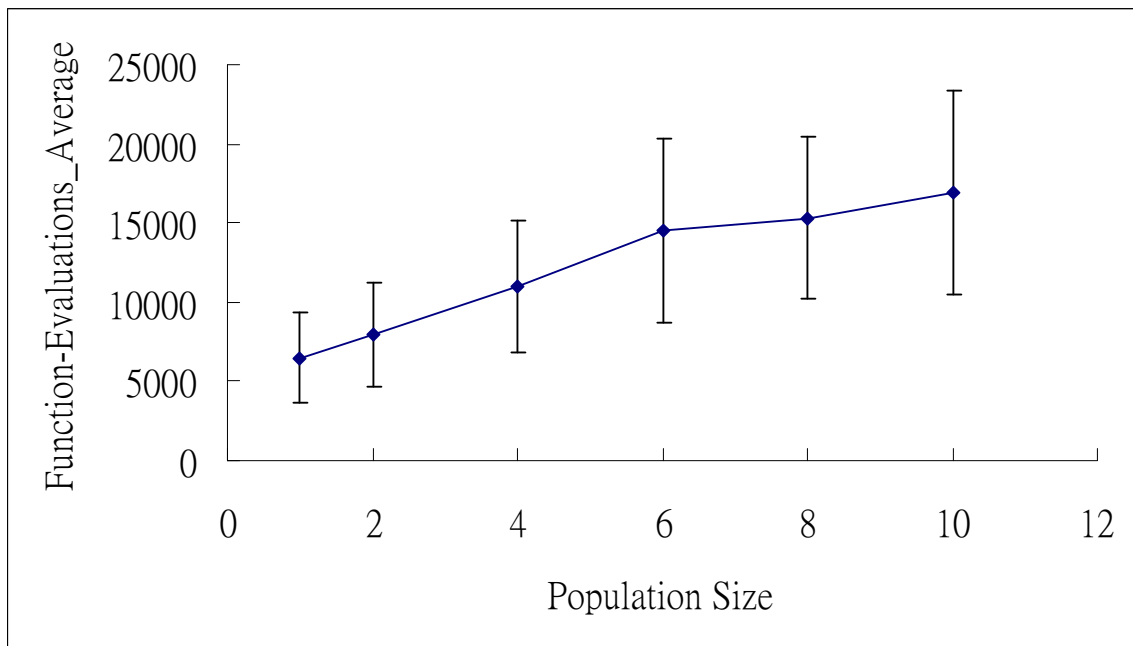


圖 4-9 μ 對 QGES 在 Rosenbrock's function 中搜尋效率的影響

表 4-8 與圖 4-10 為 Rosenbrock's function 在最佳的 μ 與 λ 的設定下 ($\mu=1, \lambda=1$), 固定 $\sigma=0.00005$, 測試不同的 β 值對 QGES 演算法的搜尋效率的影響, 實驗結果顯示最佳的 β 的參數設定值為 $\beta=50\sigma$, 此時 QGES 演算法有最佳的搜尋效率。

表 4-8 應變數 β 在 Rosenbrock's function 中的實驗數據

Beta	Best-Value		Time (sec.)		Function-Evaluations	
	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
20σ	0.000989784	1.98235E-05	7.7859	7.182608938	29649.32	27553.84754
30σ	0.000973287	3.45401E-05	3.18752	3.334468016	12008.28	12822.21291
40σ	0.000979656	2.54784E-05	2.26874	1.672960761	8481.72	6430.687242
50σ	0.000977601	4.66671E-05	1.81242	0.66652327	6734.12	2561.046826
60σ	0.000955701	7.53576E-05	1.93854	0.837587698	7219.84	3221.41959
70σ	0.00097268	5.25118E-05	1.99612	0.799546189	7456.12	3092.192094
80σ	0.000982152	2.76162E-05	2.1464	0.757370047	8003.16	2901.488565

備註：自變數 $\mu=1$ 、 $\lambda=1$ 、 $\sigma=0.00005$

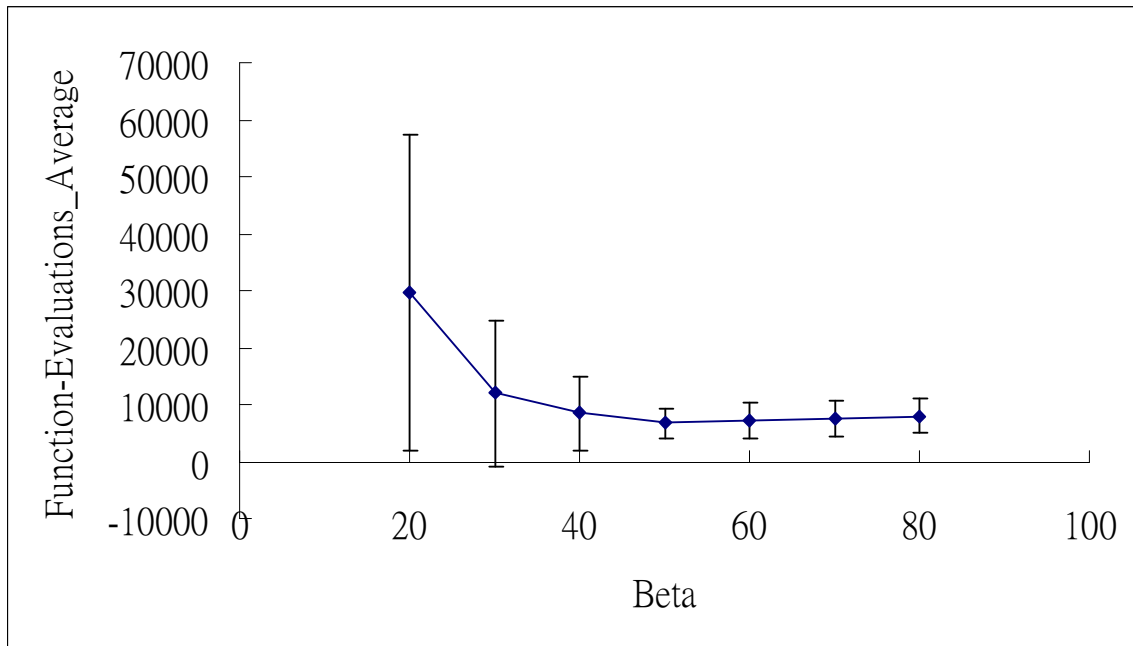


圖 4-10 β 對 QGES 在 Rosenbrock's function 中搜尋效率的影響

表 4-9 與圖 4-11 為 Rosenbrock's function 在最佳的 μ 、 λ 與 β 的設定下 ($\mu=1$ ， $\lambda=1$ ， $\beta=50\sigma$)，測試不同的 σ 值對 QGES 演算法的搜尋效率的影響，實驗結果顯示最佳的 σ 的參數設定值為 $\sigma=0.01$ ，此時 QGES 演算法有最佳的搜尋效率。

表 4-9 應變數 σ 在 Rosenbrock's function 中的實驗數據

Sigma	Best-Value		Time (sec.)		Function-Evaluations	
	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
0.000005	378.4801789	523.8450222	45.0616	12.82998958	180315.92	51490.43406
0.00001	250.4618132	486.4231709	41.63278002	16.29717906	165171.12	64748.87963
0.00005	0.000952291	7.30582E-05	1.91836	0.842333924	6907.84	3026.788831
0.0001	0.000970655	4.93558E-05	2.23904	0.850285425	8694.28	3409.475793
0.0005	0.000987225	2.95877E-05	2.9634	1.482538336	11600.76	5961.918437
0.001	0.000986763	1.60719E-05	3.09506	1.501274104	12120.04	6032.087414
0.005	0.000949017	5.58227E-05	3.16062	1.185494091	12377.52	4771.932562

備註：自變數 $\mu=1$ 、 $\lambda=1$ 、 $\beta=50\sigma$

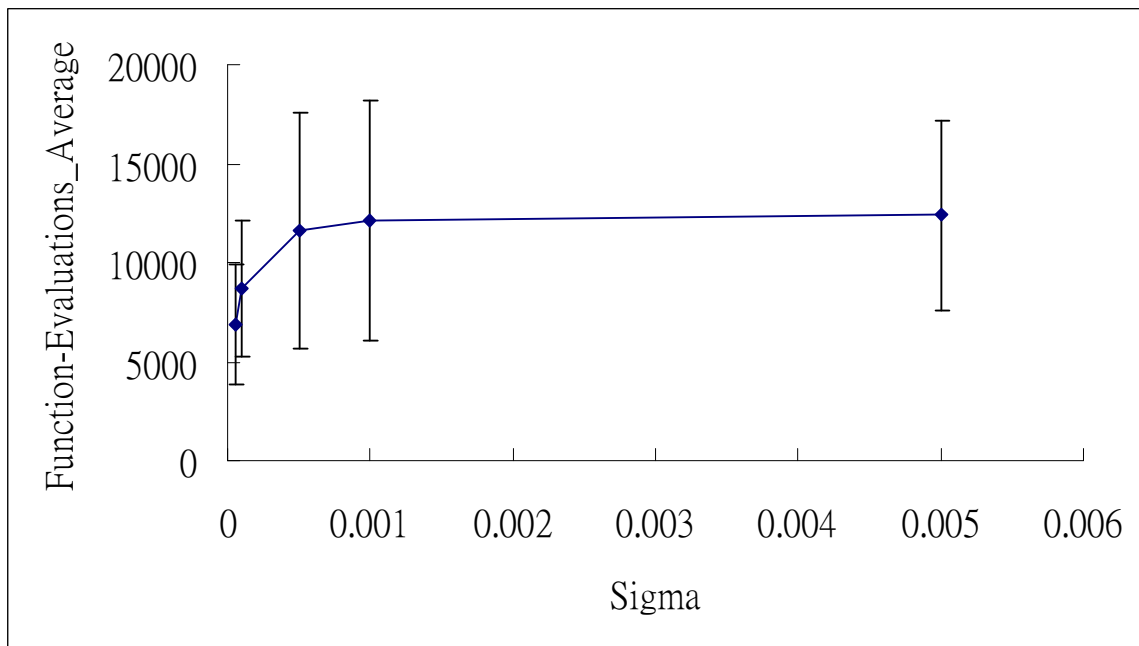


圖 4-11 σ 對 QGSES 在 Rosenbrock's function 中搜尋效率的影響

從以上的數據顯示 QGSES 演算法在 Rosenbrock's function 中最佳的參數設計為 $\mu=1$ 、 $\lambda=1$ 、 $\sigma=0.00005$ 、 $\beta=50\sigma$ ，在最佳的參數設計下，此時 QGSES 演算法有最佳的搜尋效率，收斂百分比為 100%，達收斂時的平均運算次數為 6501.64 次，達收斂時的平均次數之標準差為 2810.238117。

(4) QGSES 與其他搜尋方法的比較

表 4-10 為 Rosenbrock's function 在 QGSES、CMA 與 Y-L 三種方法的實驗結果，測試次數皆為 50 次的重複試驗。圖 4-12 為 Rosenbrock's function 在 QGSES、CMA 與 Y-L 三種方法之平均運算次數比較圖。由表 4-10 中的實驗數據顯示，在單一最佳解的函數中，如同函數 1，QGSES 的收斂效率優於 Y-L 所提出的架構，然而 CMA 有最好的收斂效率，但是與函數 1 的數據相比，當變數數量增加，QGSES 與 CMA 的收斂效率的差距也快速的縮小。此外，根據最新的研究，Deb 提出的 G3PCX 在單一最佳解的函數中的收斂效率平均優於 CMA 約 15%。

表 4-10 不同的演算法對 Rosenbrock's function 的測試結果

Method	Convergence Percentage	Best-Value		Time (sec.)		Function-Evaluations	
		Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
QGSES	100%	0.000984487	2.27294E-05	1.83184	0.849032327	6501.64	2810.238117
CMA	100%	0.000936813	5.68755E-05	0.6215	0.108787051	5453.2	1067.268954
Y-L	100%	0.000992075	0.000279166	98.11093333	82.41250569	73785.53333	31507.87416

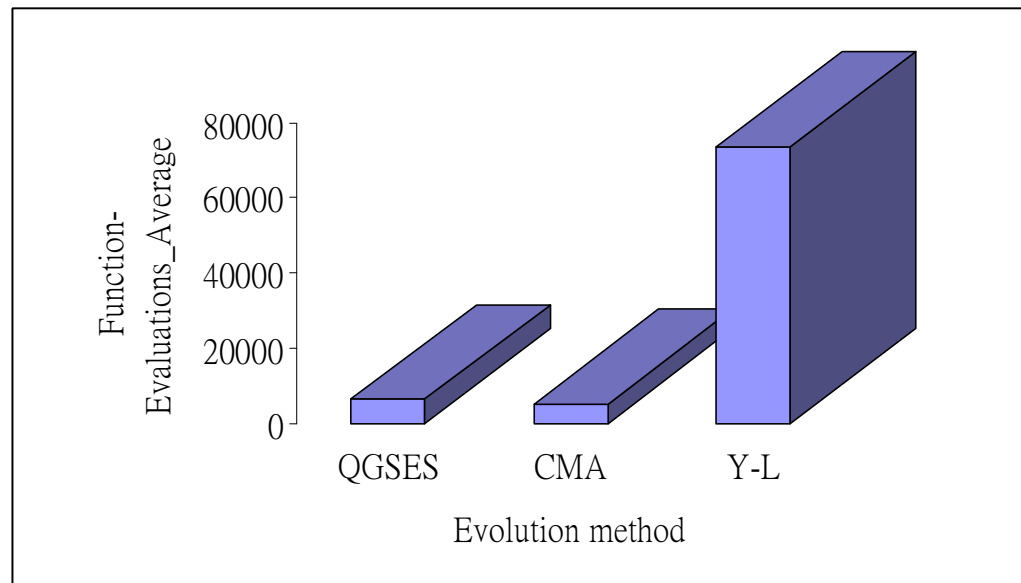


圖 4-12 Rosenbrock's function 之平均次數時間比較圖

4.1.3 函數 3：Shubert function

(1) 目標函數

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

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

此函數目的在求此函數的最小值，變數數量為 2 個變數，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。

(2) 測試環境與參數設定

QGSES 演算法在 Shubert function 之相關變數設定如下：

- a. 母體 μ 的數量設定：在本函數中母體 μ 的數量分別為 1、2、4、6、8、10。
- b. 子代 λ 的數量設定：在本函數中子代 λ 的數量為 1~10。
- c. 突變強度 σ ：在本函數中初始的突變強度 σ 為 0.5，突變強度 σ 分別為 0.01、0.05、0.1、0.25、0.5、0.75、1。
- d. β 值：在本函數中初始的 β 值為 10σ ， β 值的分別為 5σ 、 10σ 、 20σ 、 30σ 、 40σ 、 50σ 、 60σ 。
- e. 收斂條件：在本範例中，當與最佳值的誤差小於 10^{-3} ，則視為收斂。
- f. 測試次數：50 次。
- g. 最大演化代數：10000 代。

(3) QGSES 實驗結果

圖 4-13 為 Shubert function 在固定 $\beta = 10\sigma$ 與 $\sigma = 0.5$ 下，不同的 μ 與 λ 對 QGSES 演算法的搜尋效率的影響，詳細數據請參考附錄，由圖 4-13 可知在固定的 β 與 σ 設定下，最佳的 μ 與 λ 的參數設定值為 $\mu = 1$ 、 $\lambda = 3$ ，此時 QGSES 演算法有最佳的搜尋效率，收斂百分比為 100%，達收斂時的平均運算次數為 381.64 次，達收斂時的平均次數之標準差為 30.62196091。

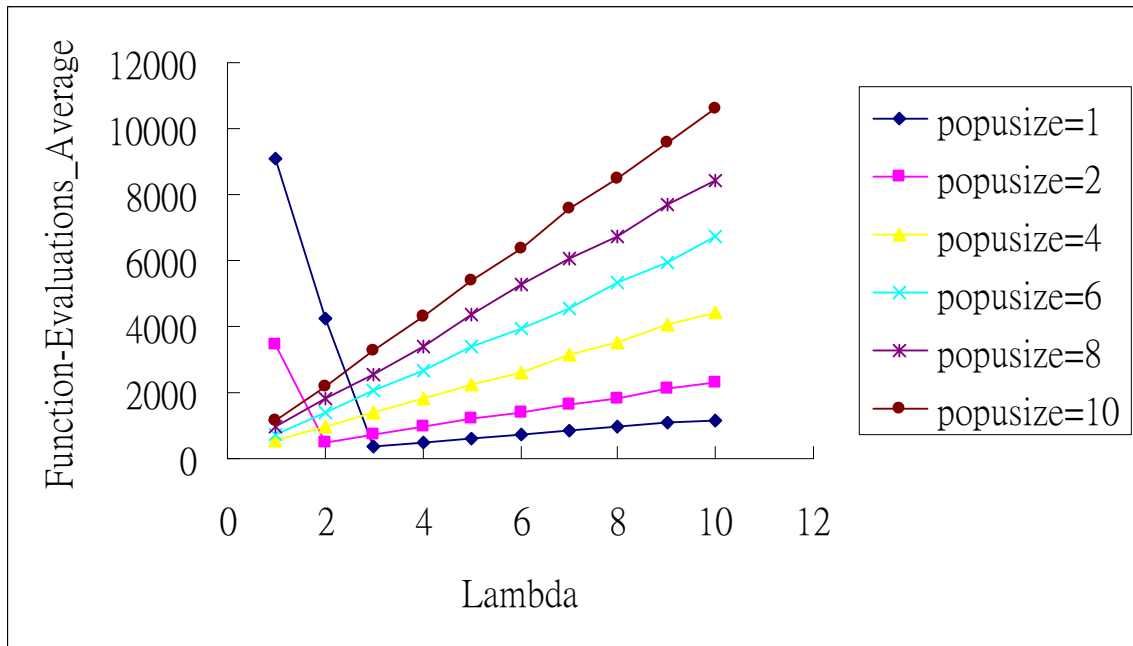


圖 4-13 μ 與 λ 對 QGSES 在 Shubert function 中搜尋效率的影響

表 4-11 與圖 4-14 為 Shubert function 在固定 $\mu=1$ 、 $\beta=10\sigma$ 與 $\sigma=0.5$ 下，不同的 λ 對 QGSES 演算法的搜尋效率的影響，在固定的 μ 、 β 與 σ 設定下，最佳的 λ 的參數設定值為 $\lambda=3$ ，此時 QGSES 演算法有最佳的搜尋效率。

表 4-11 應變數 λ 在 Shubert function 中的實驗數據

Lambda	Best-Value		Time (sec.)		Function-Evaluations	
	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
1	-186.7059092	0.037718162	1.48444	1.544605255	9085.84	9875.117228
2	-186.7219603	0.022510104	0.56592	1.423488742	4248.12	12038.693
3	-186.7283138	0.001496897	0.10348	0.007677797	381.64	30.62196091
4	-186.7284337	0.001592326	0.11192	0.007283865	494.6	41.49846321
5	-186.7286497	0.001504867	0.12188	0.008925176	612.4	58.02919744
6	-186.7286609	0.001684781	0.13124	0.008299398	739	42.14842578
7	-186.7289112	0.001309551	0.13918	0.009635585	855.84	85.43582387
8	-186.7287566	0.001530204	0.1494	0.009549014	968.36	70.31757698
9	-186.7286275	0.001366697	0.159	0.009839529	1088.2	81.64032617
10	-186.7285581	0.001403641	0.16528	0.009521276	1179.4	97.96625825

備註：自變數 $\mu=1$ 、 $\sigma=0.5$ 、 $\beta=10\sigma$

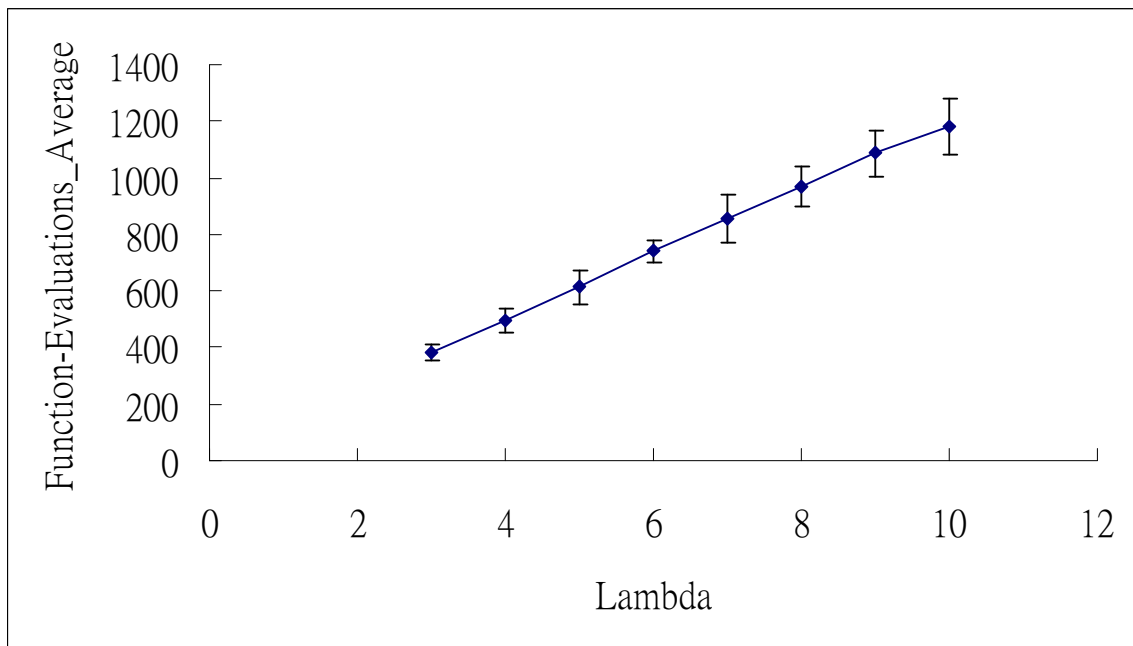


圖 4-14 λ 對 QGSES 在 Shubert function 中搜尋效率的影響

表 4-12 與圖 4-15 為 Shubert function 在固定 $\lambda=3$ 、 $\beta=10\sigma$ 與 $\sigma=0.5$ 下，不同的 μ 對 QGSES 演算法的搜尋效率的影響，在固定的 λ 、 β 與 σ 設定下，最佳的 μ 的參數設定值為 $\mu=1$ ，此時 QGSES 演算法有最佳的搜尋效率。

表 4-12 應變數 μ 在 Shubert function 中的實驗數據

Population size	Best-Value		Time (sec.)		Function-Evaluations	
	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
1	-186.7283138	0.001496897	0.10348	0.007677797	381.64	30.62196091
2	-186.7282089	0.001136662	0.12536	0.007403143	720.8	60.95967218
4	-186.7283871	0.001254832	0.1669	0.011157957	1366.72	121.3433382
6	-186.7286734	0.001435273	0.2118	0.017038971	2032.8	193.1483516
8	-186.7285288	0.001447992	0.24686	0.024111146	2571.2	320.6035125
10	-186.7289572	0.001320682	0.29304	0.023272968	3247.6	288.8863911

備註：自變數 $\lambda=3$ 、 $\sigma=0.5$ 、 $\beta=10\sigma$

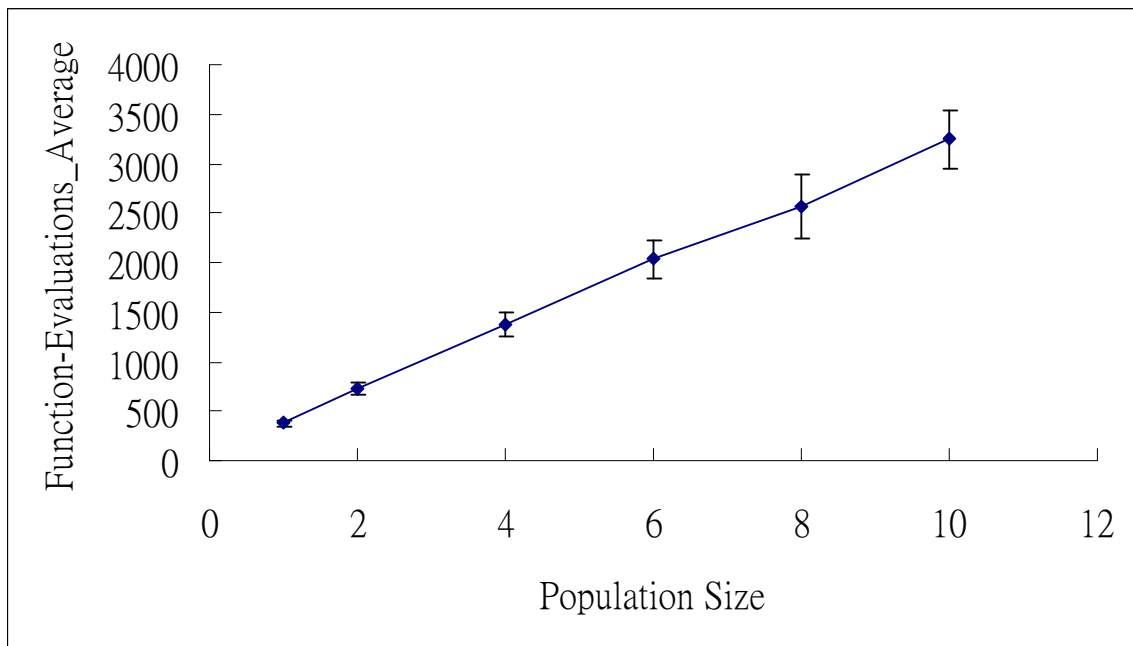


圖 4-15 μ 對 QGSES 在 Shubert function 中搜尋效率的影響

表 4-13 與圖 4-16 為 Shubert function 在最佳的 μ 與 λ 的設定下 ($\mu=1$, $\lambda=3$), 固定 $\sigma=0.5$, 測試不同的 β 值對 QGSES 演算法的搜尋效率的影響, 實驗結果顯示最佳的 β 的參數設定值為 $\beta=10\sigma$, 此時 QGSES 演算法有最佳的搜尋效率。

表 4-13 應變數 β 在 Shubert function 中的實驗數據

Beta	Best-Value		Time (sec.)		Function-Evaluations	
	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
5σ	-186.735906	0.003778096	0.56444	1.526393965	3722.08	13424.3826
10σ	-186.7286674	0.00129546	0.10626	0.00636255	383.92	37.07952843
20σ	-186.7287578	0.00146376	0.11066	0.006965249	418.36	37.73687635
30σ	-186.7283072	0.001292413	0.11156	0.007057085	436.72	31.60730993
40σ	-186.7286732	0.001374995	0.13584	0.007833367	724.8	56.99588241
50σ	-186.7288556	0.001441042	0.14906	0.00789319	895.96	50.45303332
60σ	-186.728661	0.001428355	0.16252	0.008393973	1067.8	52.38320341

備註：自變數 $\mu=1$ 、 $\lambda=3$ 、 $\sigma=0.5$

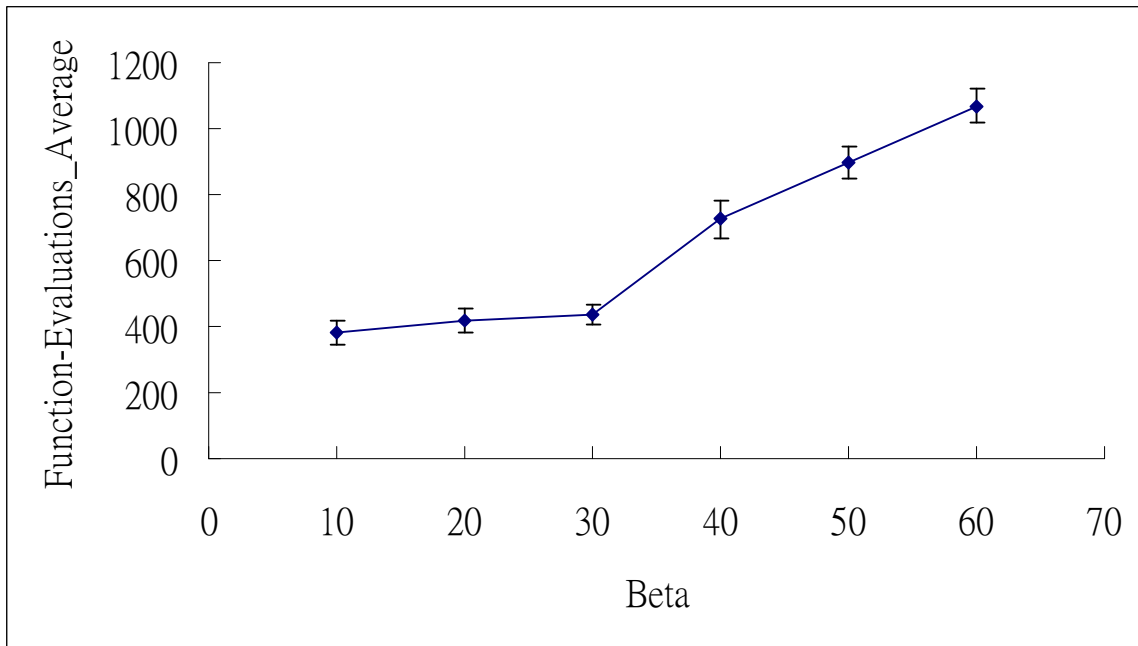


圖 4-16 β 對 QGSES 在 Shubert function 中搜尋效率的影響

表 4-14 與圖 4-17 為 Shubert function 在最佳的 μ 、 λ 與 β 的設定下 ($\mu=1$ ， $\lambda=3$ ， $\beta=50\sigma$)，測試不同的 σ 值對 QGSES 演算法的搜尋效率的影響，實驗結果顯示最佳的 σ 的參數設定值為 $\sigma=0.5$ ，此時 QGSES 演算法有最佳的搜尋效率。

表 4-14 應變數 σ 在 Shubert function 中的實驗數據

Sigma	Best-Value		Time (sec.)		Function-Evaluations	
	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
0.01	-36.92224182	42.11416329	6.2797	1.277384801	57609.76	11833.58597
0.05	-76.26531028	66.326727	5.00002	2.782074432	45678.04	25746.66816
0.1	-113.753795	66.8371394	3.85746	3.227616339	34939	29750.35237
0.25	-186.7264798	0.006231963	1.456	2.426691701	12870.52	22432.07487
0.5	-186.7286674	0.00129546	0.10626	0.00636255	383.92	37.07952843
0.75	-186.7286096	0.001429529	0.13932	0.008786446	806.92	57.89922279
1	-186.7284348	0.001452734	0.15968	0.00900893	1020.76	78.24883386

備註：自變數 $\mu=1$ 、 $\lambda=3$ 、 $\beta=10\sigma$

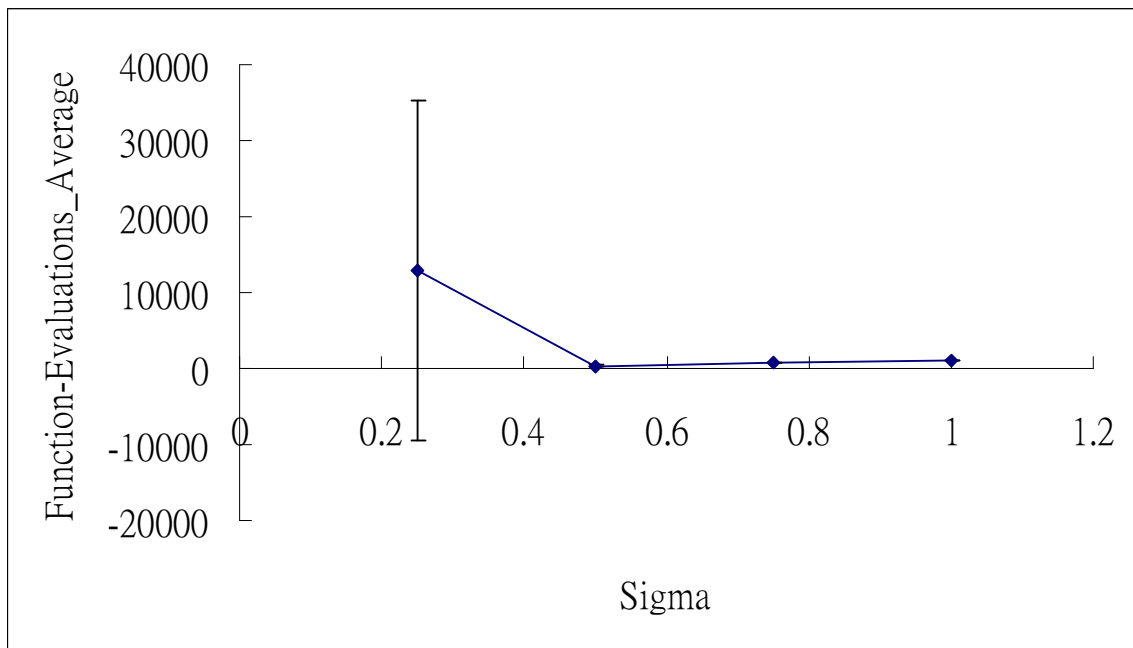


圖 4-17 σ 對 QGSES 在 Shubert function 中搜尋效率的影響

從以上的數據顯示 QGSES 演算法在 Shubert function 中最佳的參數設計為 $\mu=1$ 、 $\lambda=3$ 、 $\sigma=0.5$ 、 $\beta=10\sigma$ ，在最佳的參數設計下，此時 QGSES 演算法有最佳的搜尋效率，收斂百分比為 100%，達收斂時的平均運算次數為 381.64 次，達收斂時的平均次數之標準差為 30.62196091。

(4) QGSES 與其他搜尋方法的比較

表 4-15 為 Shubert function 在 QGSES、CMA、Y-L 與 G3PCX 與四種方法的實驗結果，測試次數皆為 50 次的重複試驗。圖 4-18 為 Shubert function 在 QGSES、CMA、Y-L 與 G3PCX 與四種方法之平均運算次數比較圖。由表 4-15 中的實驗數據顯示，在有多重區域解的函數中，QGSES 有最好的收斂效率，而且有 100% 的收斂比率，而 CMA 與 Y-L 的收斂比率分別只有 28% 與 82%，G3PCX 雖然有 100% 的收斂比率，然而其收斂效率與 QGSES 相比相對較差。

表 4-15 不同的演算法對 Shubert function 的測試結果

Method	Convergence Percentage	Best-Value		Time (sec.)		Function-Evaluations	
		Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
QGSES	100%	-186.7283138	0.001496897	0.10348	0.007677797	381.64	30.62196091
CMA	28%	-102.911605	63.55770874	3.88314	2.380558295	43329.6	27004.33571
Y-L	82%	-179.727	1.802808	132.4651	178.3495425	108387.4667	104628.039
G3PCX	100%	-186.7284981	0.00137794	2.339	4.194943171	2173	3869.727518

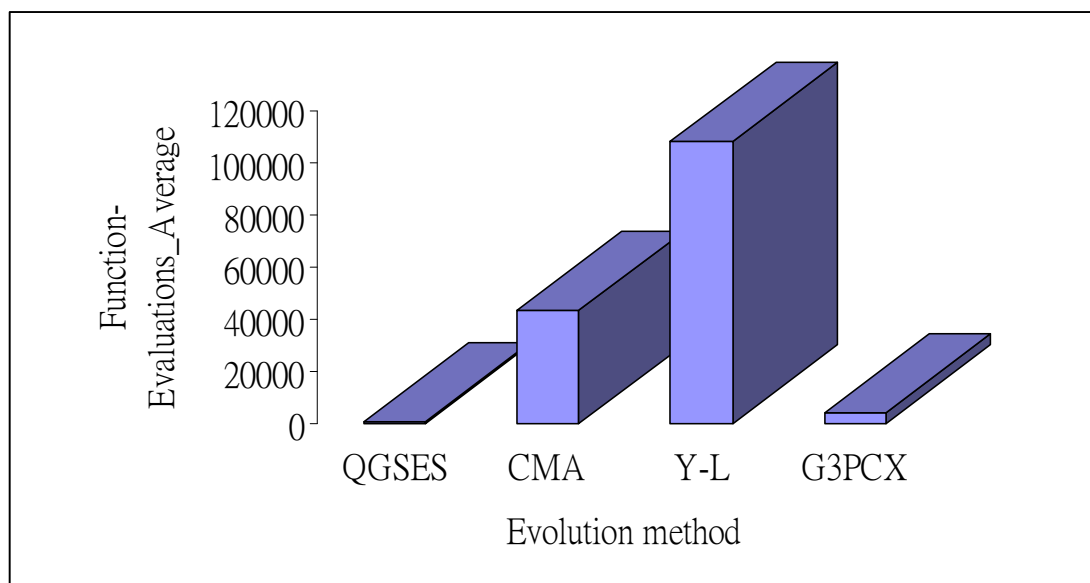


圖 4-18 Shubert function 之平均運算次數比較圖

4.1.4 函數 4：SinCos function

(1) 目標函數

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

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

此函數目的在求函數的最小值，變數數量為 2 個變數，函數之理想(Ideal)最佳值為 0。此函數有 20,480 個區域最佳解。

(2) 測試環境與參數設定

QGSES 演算法在 SinCos function 之相關變數設定如下：

- a. 母體 μ 的數量設定：在本函數中母體 μ 的數量分別為 4、6、8、10、20、30。
- b. 子代 λ 的數量設定：本研究架構的實驗中，子代 λ 的數量設定直接影響每一代產生的子代數目，以及每一代移動的子代數量。在本函數中子代 λ 的數量為 1~10。
- c. 突變強度 σ ：在本函數中初始的突變強度 σ 為 2，突變強度 σ 分別為 1、1.5、2、2.5、3、3.5、4。
- d. β 值：在本函數中初始的 β 值為 15σ ， β 值的分別為 5σ 、 10σ 、 15σ 、 20σ 、 25σ 、 30σ 。
- e. 收斂條件：在本範例中，當與最佳值的誤差小於 10^{-3} ，則視為收斂。
- f. 測試次數：50 次。
- g. 最大演化代數：10000 代。

(3) QGSES 實驗結果

圖 4-19 為 SinCos function 在固定 $\beta = 15\sigma$ 與 $\sigma = 2$ 下，不同的 μ 與 λ 對 QGSES 演算法的搜尋效率的影響，詳細數據請參考附錄，由圖 4-19 可知在固定的 β 與 σ 設定下，最佳的 μ 與 λ 的參數設定值為 $\mu = 4$ 、 $\lambda = 4$ ，此時 QGSES 演算法有最佳的搜尋效率，收斂百分比為 100%，達收斂時的平均運算次數為 3286.56 次，達收斂時的平均次數之標準差為 1654.555174。

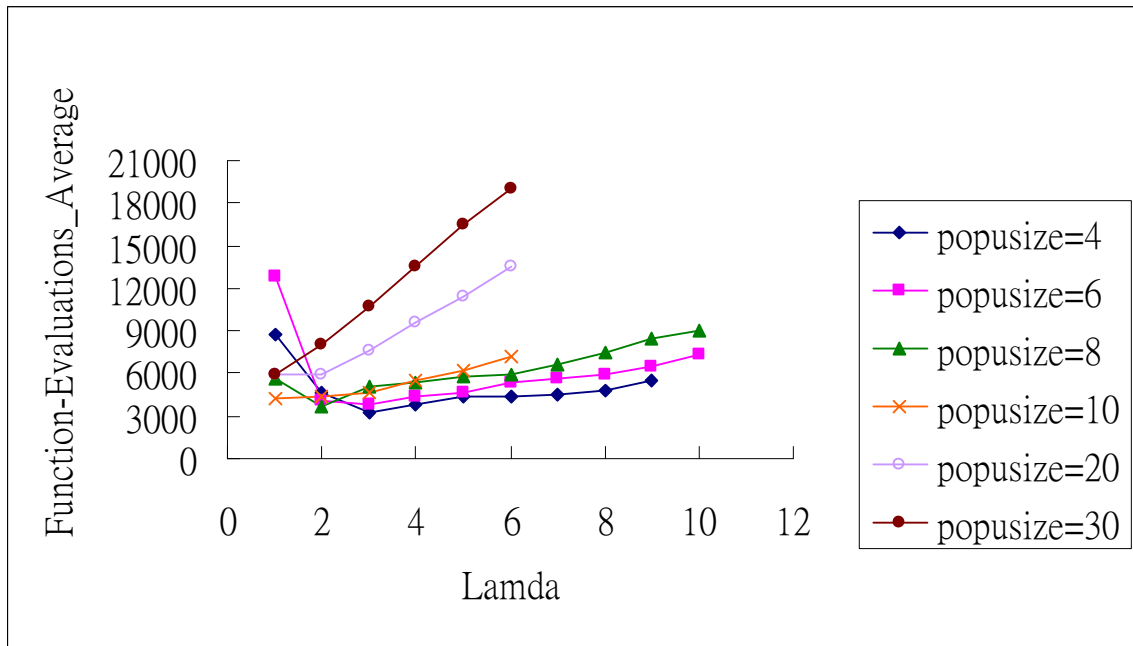


圖 4-19 μ 與 λ 對 QGSES 在 SinCos function 中搜尋效率的影響

表 4-16 與圖 4-20 為 SinCos function 在固定 $\mu=4$ 、 $\beta=15\sigma$ 與 $\sigma=2$ 下，不同的 λ 對 QGSES 演算法的搜尋效率的影響，在固定的 μ 、 β 與 σ 設定下，最佳的 λ 的參數設定值為 $\lambda=4$ ，此時 QGSES 演算法有最佳的搜尋效率。

表 4-16 應變數 λ 在 SinCos function 中的實驗數據

Lambda	Best-Value		Time (sec.)		Function-Evaluations	
	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
1	0.00200059	0.002774057	2.82994	3.103286891	28386.72	31929.88332
2	0.000626185	0.000304946	0.83412	1.904830772	8731.36	22222.47913
3	0.000632366	0.000294788	0.46564	0.300539363	4685.92	3635.91502
4	0.000589368	0.000240899	0.34852	0.135867468	3286.56	1654.555174
5	0.000645906	0.0002887	0.38808	0.173155214	3857.6	2155.590511
6	0.000568255	0.000265485	0.4218	0.232283694	4321.12	2945.498625
7	0.00059452	0.000253621	0.42254	0.165389239	4332.8	2084.679352
8	0.000612947	0.000237812	0.43184	0.08947588	4469.92	1132.555352
9	0.000580325	0.000248577	0.45416	0.113571763	4771.84	1435.753821
10	0.000545721	0.000281159	0.50722	0.158833155	5460	2030.10807

備註：自變數 $\mu=4$ 、 $\sigma=2$ 、 $\beta=15\sigma$

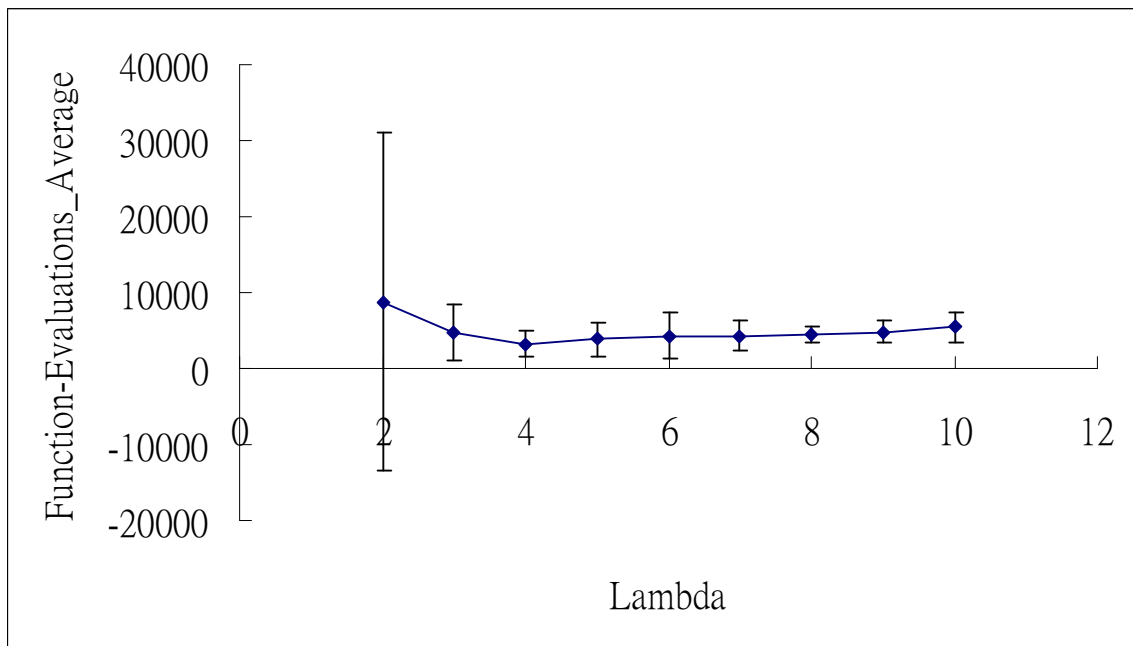


圖 4-20 λ 對 QGSES 在 SinCos function 中搜尋效率的影響

表 4-17 與圖 4-21 為 SinCos function 在固定 $\lambda=4$ 、 $\beta=15\sigma$ 與 $\sigma=2$ 下，不同的 μ 對 QGSES 演算法的搜尋效率的影響，在固定的 λ 、 β 與 σ 設定下，最佳的 μ 的參數設定值為 $\mu=4$ ，此時 QGSES 演算法有最佳的搜尋效率。

表 4-17 應變數 μ 在 SinCos function 中的實驗數據

Population size	Best-Value		Time (sec.)		Function-Evaluations	
	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
4	0.000589368	0.000240899	0.34852	0.135867468	3286.56	1654.555174
6	0.000571975	0.000248216	0.42588	0.187537603	4435.44	2427.371234
8	0.000578476	0.000252624	0.49216	0.179036236	5346.88	2318.506351
10	0.000639013	0.000230306	0.5038	0.154096842	5536.4	1952.168611
20	0.000548917	0.000270675	0.81752	0.065912818	9520.8	827.220179
30	0.000577389	0.000241216	1.13504	0.084187827	13537.2	1062.348265

備註：自變數 $\lambda=4$ 、 $\sigma=2$ 、 $\beta=15\sigma$

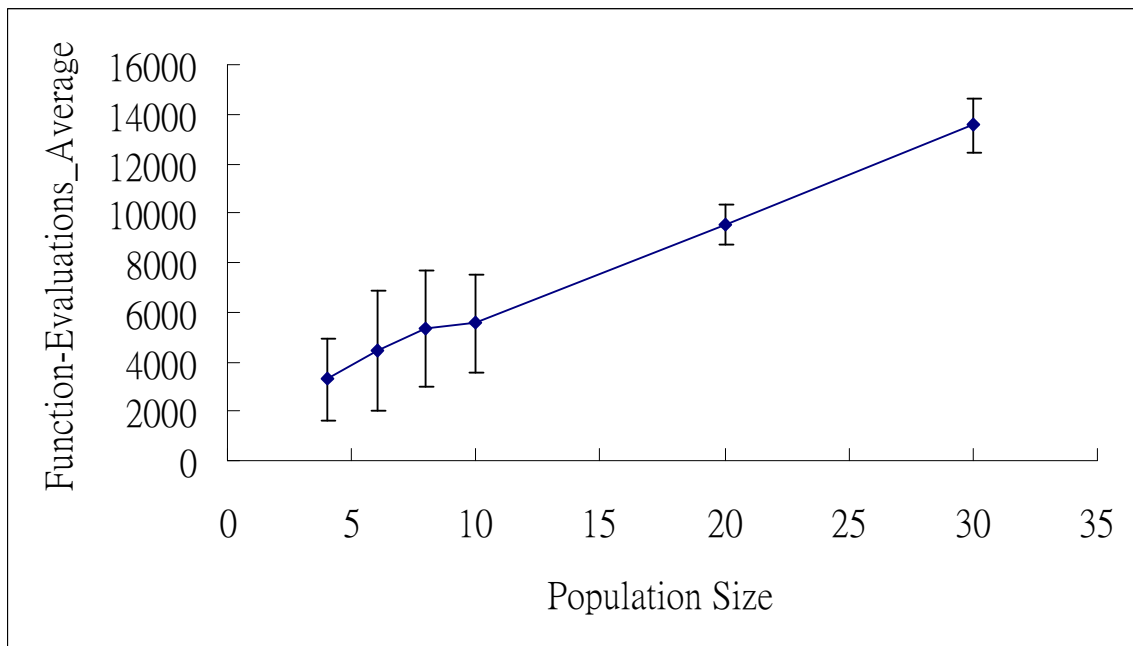


圖 4-21 μ 對 QGSES 在 SinCos function 中搜尋效率的影響

表 4-18 與圖 4-22 為 SinCos function 在最佳的 μ 與 λ 的設定下 ($\mu=4$, $\lambda=4$), 固定 $\sigma=2$, 測試不同的 β 值對 QGSES 演算法的搜尋效率的影響, 實驗結果顯示最佳的 β 的參數設定值為 $\beta=15\sigma$, 此時 QGSES 演算法有最佳的搜尋效率。

表 4-18 應變數 β 在 SinCos function 中的實驗數據

Beta	Best-Value		Time (sec.)		Function-Evaluations	
	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
5σ	0.000679724	0.000234974	0.85528	0.555497352	9577.76	6879.563164
10σ	0.000645423	0.000234587	0.4397	0.299391781	4463.52	3711.204756
15σ	0.000589368	0.000240899	0.34852	0.135867468	3286.56	1654.555174
20σ	0.000611804	0.000243767	0.35904	0.176283279	3456.8	2172.099746
25σ	0.00053737	0.000265085	0.37214	0.175634594	3626.4	2183.880089
30σ	0.000596927	0.000273609	0.39718	0.17871328	3923.36	2209.284359

備註：自變數 $\mu=4$ 、 $\lambda=4$ 、 $\sigma=2$

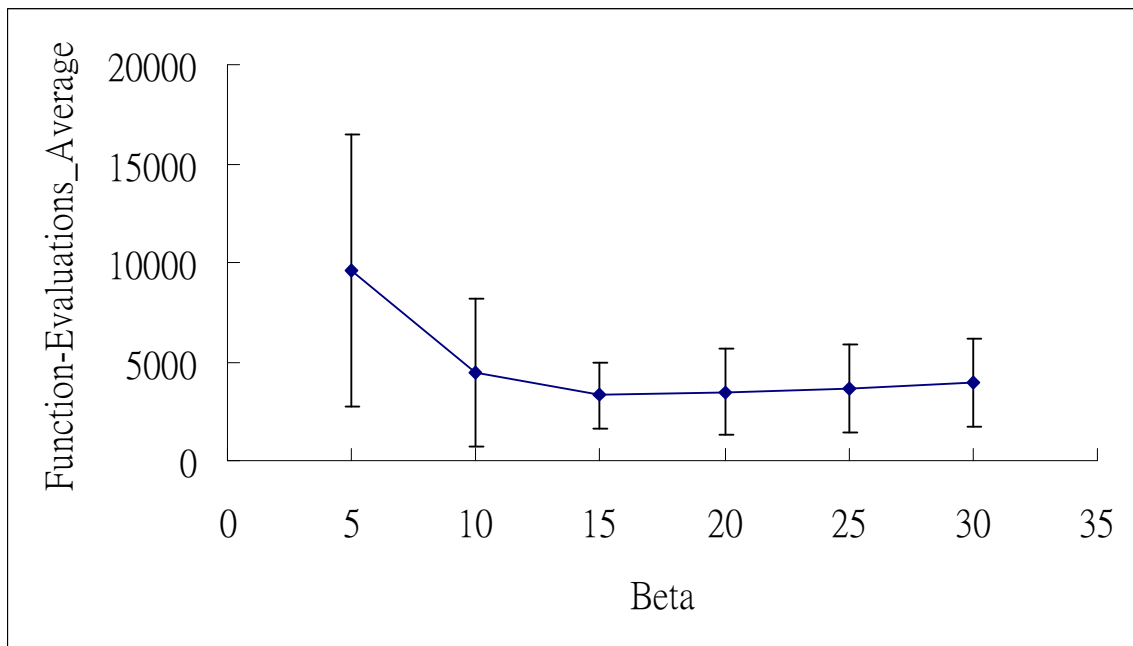


圖 4-22 β 對 QGSES 在 SinCos function 中搜尋效率的影響

表 4-19 與圖 4-23 為 SinCos function 在最佳的 μ 、 λ 與 β 的設定下 ($\mu = 4$, $\lambda = 4$, $\beta = 15\sigma$)，測試不同的 σ 值對 QGSES 演算法的搜尋效率的影響，實驗結果顯示最佳的 σ 的參數設定值為 $\sigma = 2$ ，此時 QGSES 演算法有最佳的搜尋效率。

表 4-19 應變數 σ 在 SinCos function 中的實驗數據

Sigma	Best-Value		Time (sec.)		Function-Evaluations	
	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
1	0.000716833	0.000266745	4.3091	4.407724496	52513.44	54703.75998
1.5	0.000576474	0.000273266	0.57068	0.409824723	6093.6	5118.882531
2	0.000564078	0.000251482	0.35378	0.147591174	3399.2	1814.301462
2.5	0.000548584	0.000301599	0.36942	0.175314888	3588	2148.902102
3	0.000610035	0.000240931	0.43472	0.198935436	4347.04	2469.765129
3.5	0.000599583	0.000263565	0.55214	0.338750164	5855.52	4192.206931
4	0.000535	0.000276	0.61254	0.45642	6467.36	5564.57

備註：自變數 $\mu = 4$ 、 $\lambda = 4$ 、 $\beta = 15\sigma$

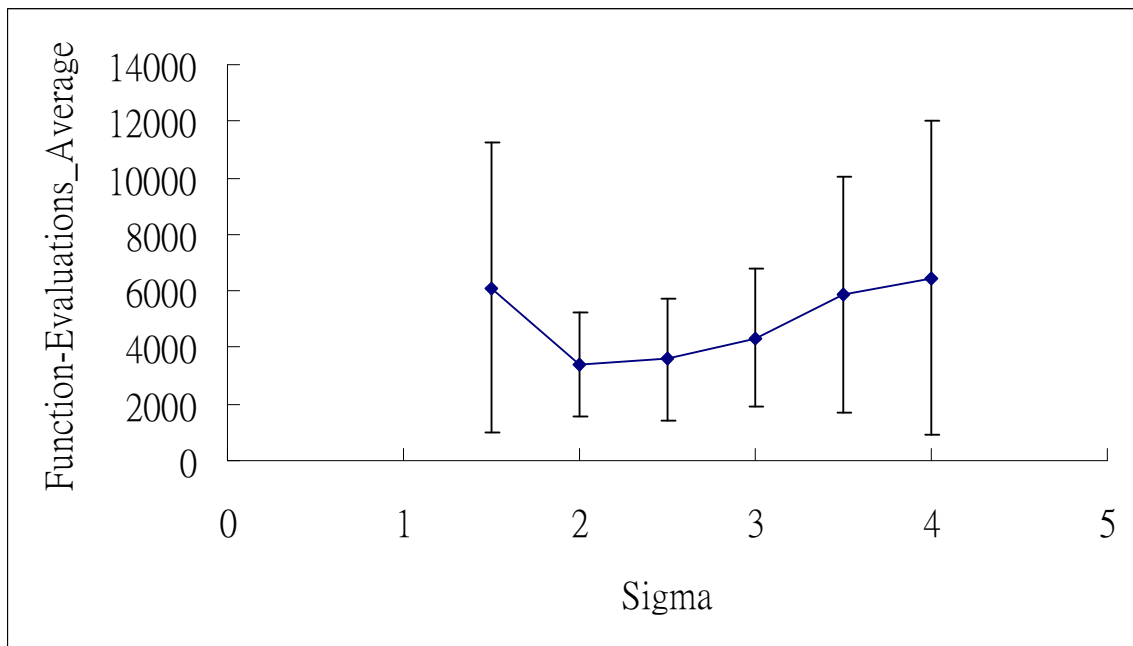


圖 4-23 σ 對 QGSES 在 SinCos function 中搜尋效率的影響

從以上的數據顯示 QGSES 演算法在 SinCos function 中最佳的參數設計為 $\mu=4$ 、 $\lambda=4$ 、 $\sigma=2$ 、 $\beta=15\sigma$ ，在最佳的參數設計下，此時 QGSES 演算法有最佳的搜尋效率，收斂百分比為 100%，達收斂時的平均運算次數為 3286.56 次，達收斂時的平均次數之標準差為 1654.555174。

(4) QGSES 與其他搜尋方法的比較

表 4-20 為 SinCos function 在 QGSES、CMA、Y-L 與 G3PCX 與四種方法的實驗結果，測試次數皆為 50 次的重複試驗。圖 4-24 為 SinCos function 在 QGSES、CMA、Y-L 與 G3PCX 與四種方法之平均運算次數比較圖。由表 4-20 中的實驗數據顯示，在有多重區域解的函數中，QGSES 有最好的收斂效率，而且有 100% 的收斂比率，而 CMA 的收斂比率在 SinCos function 為 0%，表示在 SinCos function 中 CMA 一定會陷入區域解無法跳脫。Y-L 雖然有 100% 的收斂比率，然而其收斂效率與 QGSES 相比相對較差，G3PCX 的收斂比率為 72%，有時會陷入區域解，若不考慮 G3PCX 陷入區域解的情況，其平均運算次數為 8398.5，標準差為 4020.231，與 QGSES 相比 G3PCX 的收斂效率依然相對較差。

表 4-20 不同的演算法對 SinCos function 的測試結果

Method	Convergence Percentage	Best-Value		Time (sec.)		Function-Evaluations	
		Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
QGSES	100%	0.000589368	0.000240899	0.34852	0.135867468	3286.56	1654.555174
CMA	0%	18.1041064	8.451106451	6.84506	0.144871993	80000	0
Y-L	100%	0.000686666	0.000548008	168.70704	189.0608665	108643.88	67726.58921
G3PCX	72%	0.067693824	0.108670226	12.58028	9.00893196	14474.92	10412.81234

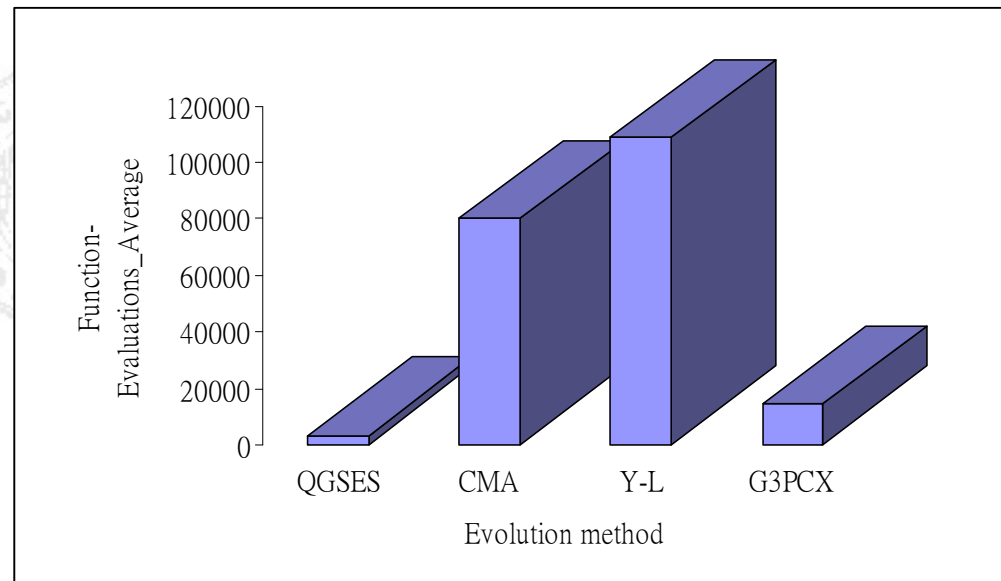


圖 4-24 SinCos function 之平均運算次數比較圖

4.1.5 函數 5：xsin(1/x) function with 2 variables

(1) 目標函數

$$\text{Min. } F(X) = \sum_{i=1}^2 x_i \sin\left(\frac{1}{x_i}\right)$$

where $x_i \in [0, 0.25]$

此函數目的在求函數的最小值，變數數量為 2 個變數，函數之理想(Ideal)最佳值為-0.4344。此函數有無限多個區域最佳解。

(2) 測試環境與參數設定

QGSES 演算法在 xsin(1/x) function with 2 variables 之相關變數設定如下：

- a. 母體 μ 的數量設定：在本函數中母體 μ 的數量分別為 1、2、4、6、8、10。
- b. 子代 λ 的數量設定：本研究架構的實驗中，子代 λ 的數量設定直接影響每一代產生的子代數目，以及每一代移動的子代數量。在本函數中子代 λ 的數量為 1~10。
- c. 突變強度 σ ：在本函數中初始的突變強度 σ 為 0.03，突變強度 σ 分別為 0.005、0.01、0.02、0.03、0.04、0.05。
- d. β 值：在本函數中初始的 β 值為 10σ ， β 值的分別為 5σ 、 10σ 、 20σ 、 30σ 、 40σ 、 50σ 。
- e. 收斂條件：在本範例中，當與最佳值的誤差小於 10^{-3} ，則視為收斂。
- f. 測試次數：50 次。
- g. 最大演化代數：10000 代。

(3) QGSES 實驗結果

圖 4-25 為 xsin(1/x) function 在固定 $\beta = 10\sigma$ 與 $\sigma = 0.03$ 下，不同的 μ 與 λ 對 QGSES 演算法的搜尋效率的影響，詳細數據請參考附錄，由圖 4-25 可知在固定的 β 與 σ 設定下，最佳的 μ 與 λ 的參數設定值為 $\mu = 1$ 、 $\lambda = 3$ ，此時 QGSES 演算法有最佳的搜尋效率，收斂百分比為 100%，達收斂時的平均運算次數為 134.68 次，達收斂時的平均次數之標準差為 116.350883809558。

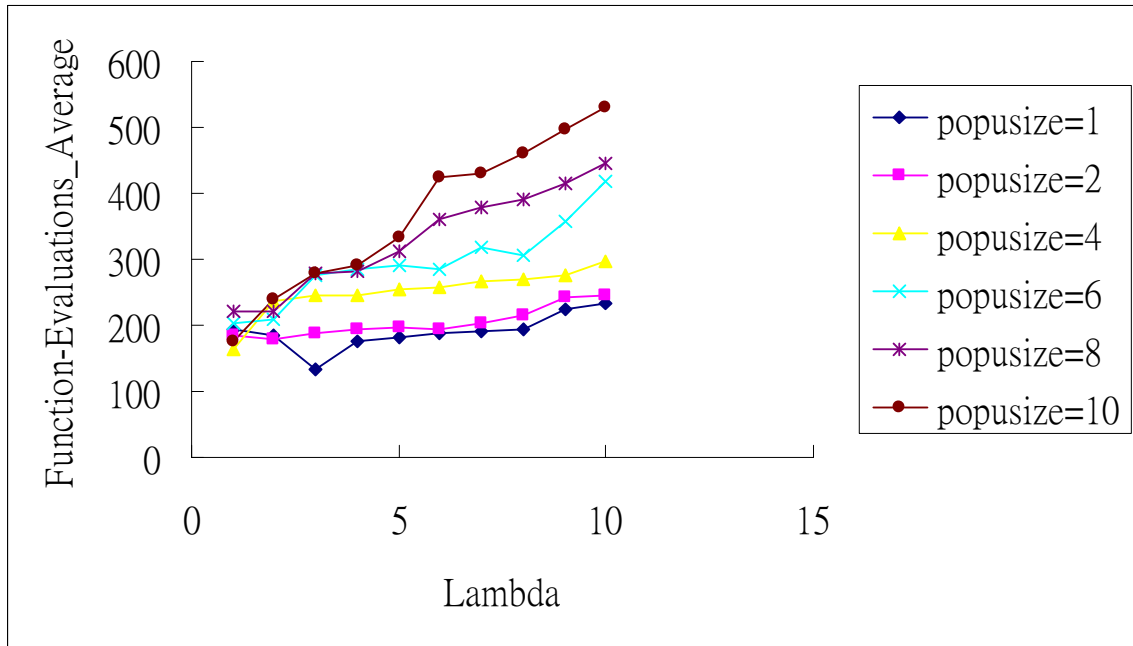


圖 4-25 μ 與 λ 對 QGSES 在 $x\sin(1/x)$ function with 2 variables 中搜尋效率的影響

表 4-21 與圖 4-26 為 $x\sin(1/x)$ function 在固定 $\mu=1$ 、 $\beta=10\sigma$ 與 $\sigma=0.03$ 下，不同的 λ 對 QGSES 演算法的搜尋效率的影響，在固定的 μ 、 β 與 σ 設定下，最佳的 λ 的參數設定值為 $\lambda=3$ ，此時 QGSES 演算法有最佳的搜尋效率。

表 4-21 應變數 λ 在 $x\sin(1/x)$ function with 2 variables 中的實驗數據

Lambda	Best-Value		Time (sec.)		Function-Evaluations	
	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
1	-0.433926229	-0.433926229	0.096	0.025995	194.6	149.4741804
2	-0.433894811	-0.433894811	0.08628	0.018701	186.12	148.3889869
3	-0.433913363	-0.433913363	0.0791	0.014242	134.68	116.3508838
4	-0.433917467	-0.433917467	0.08368	0.016718	175.56	161.6057384
5	-0.433984624	-0.433984624	0.08312	0.022582	183.2	221.4911432
6	-0.433919922	-0.433919922	0.08242	0.014876	188.68	131.9167084
7	-0.434018569	-0.434018569	0.07968	0.020511	189.72	226.1900086
8	-0.433971435	-0.433971435	0.08056	0.014907	194.28	145.0605841
9	-0.433904842	-0.433904842	0.08726	0.016943	225.64	198.4249984
10	-0.433933026	-0.433933026	0.083	0.015576	232.6	147.4734151

備註：自變數 $\mu=1$ 、 $\sigma=0.03$ 、 $\beta=10\sigma$

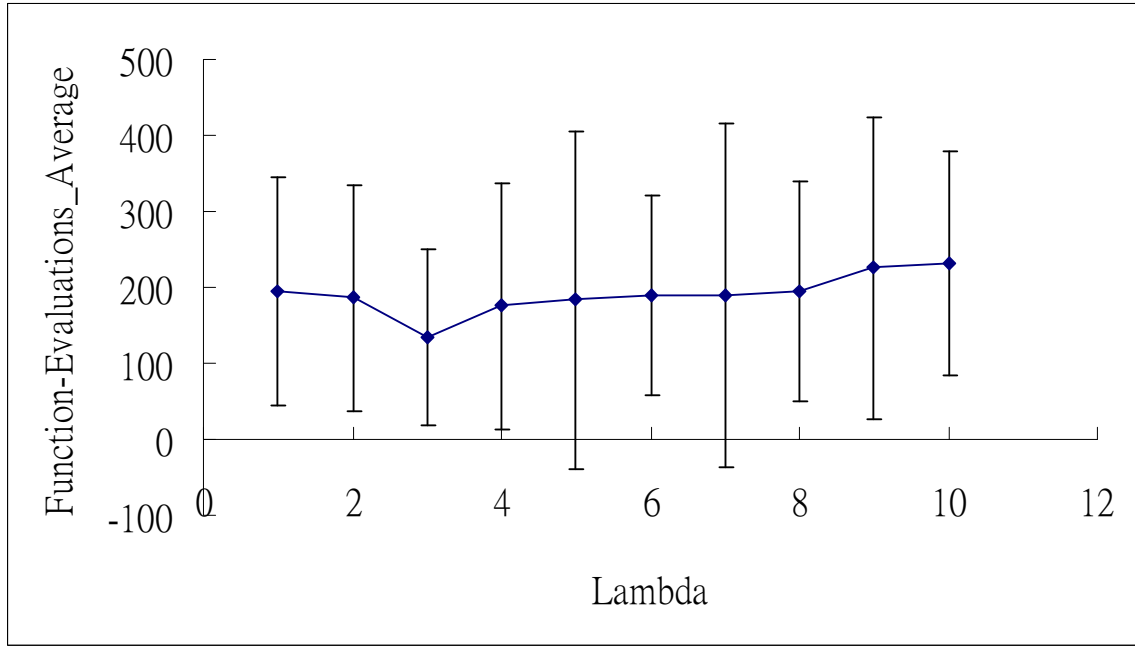


圖 4-26 λ 對 QGSES 在 $x\sin(1/x)$ function with 2 variables 中搜尋效率的影響

表 4-22 與圖 4-27 為 $x\sin(1/x)$ function 在固定 $\lambda=3$ 、 $\beta=10\sigma$ 與 $\sigma=0.03$ 下，不同的 μ 對 QGSES 演算法的搜尋效率的影響，在固定的 λ 、 β 與 σ 設定下，最佳的 μ 的參數設定值為 $\mu=1$ ，此時 QGSES 演算法有最佳的搜尋效率。

表 4-22 應變數 μ 在 $x\sin(1/x)$ function with 2 variables 中的實驗數據

Population size	Best-Value		Time (sec.)		Function-Evaluations	
	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
1	-0.433913363	0.000323263	0.0791	0.014241718	134.68	116.3508838
2	-0.433946673	0.000283724	0.079	0.015484027	186.56	159.4910477
4	-0.433898799	0.000328578	0.08278	0.016138798	245.92	202.3054712
6	-0.433925839	0.000303443	0.08256	0.013875775	276	185.3213425
8	-0.43401562	0.000314046	0.08156	0.01391543	279.68	172.0063407
10	-0.433995182	0.000332951	0.08212	0.012902523	280	182.7343338

備註：自變數 $\lambda=3$ 、 $\sigma=0.03$ 、 $\beta=10\sigma$

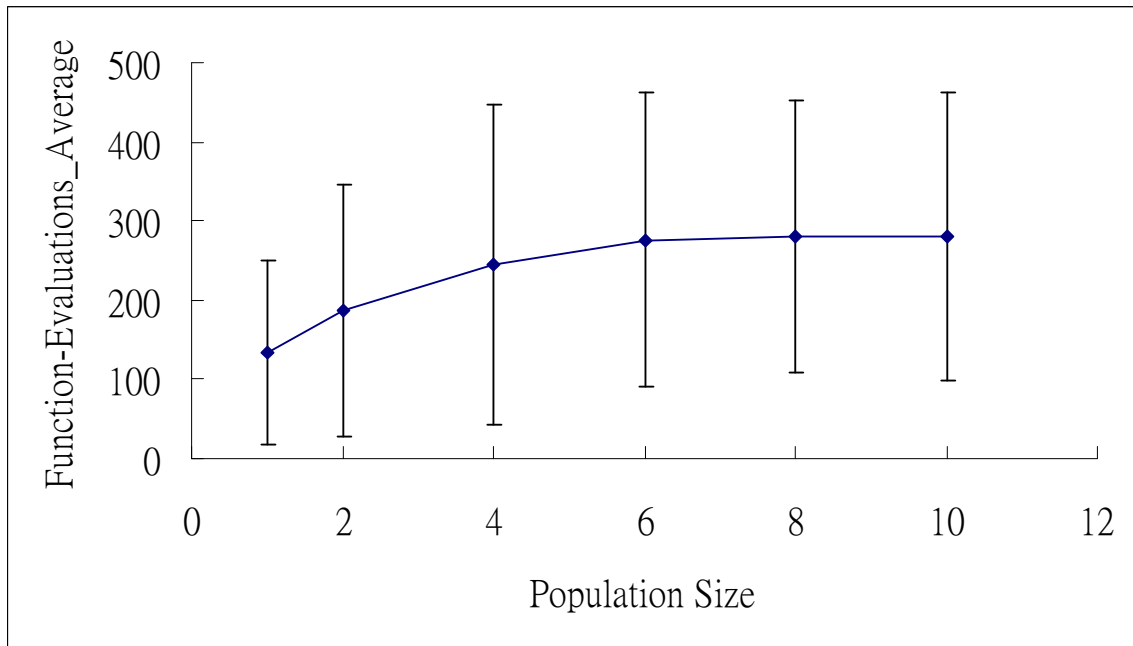


圖 4-27 μ 對 QGSES 在 $x\sin(1/x)$ function with 2 variables 中搜尋效率的影響

表 4-23 與圖 4-28 為 $x\sin(1/x)$ function 在最佳的 μ 與 λ 的設定下 ($\mu=1$, $\lambda=3$), 固定 $\sigma=0.03$, 測試不同的 β 值對 QGSES 演算法的搜尋效率的影響, 實驗結果顯示最佳的 β 的參數設定值為 $\beta=10\sigma$, 此時 QGSES 演算法有最佳的搜尋效率。

表 4-23 應變數 β 在 $x\sin(1/x)$ function with 2 variables 中的實驗數據

Beta	Best-Value		Time (sec.)		Function-Evaluations	
	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
5σ	-0.433956806	0.000346284	0.08616	0.021423199	176.2	128.6742446
10σ	-0.433898324	0.000304693	0.07804	0.013039907	136.6	122.9932801
20σ	-0.43381151	0.000318705	0.08026	0.01726718	153.64	143.9038454
30σ	-0.433918237	0.000324673	0.08062	0.01422256	154.96	129.2303686
40σ	-0.433882165	0.000298053	0.08284	0.016161633	183.64	146.9151859
50σ	-0.434002514	0.000315948	0.08472	0.018969406	186.16	181.3136149

備註：自變數 $\mu=1$ 、 $\lambda=3$ 、 $\sigma=0.03$

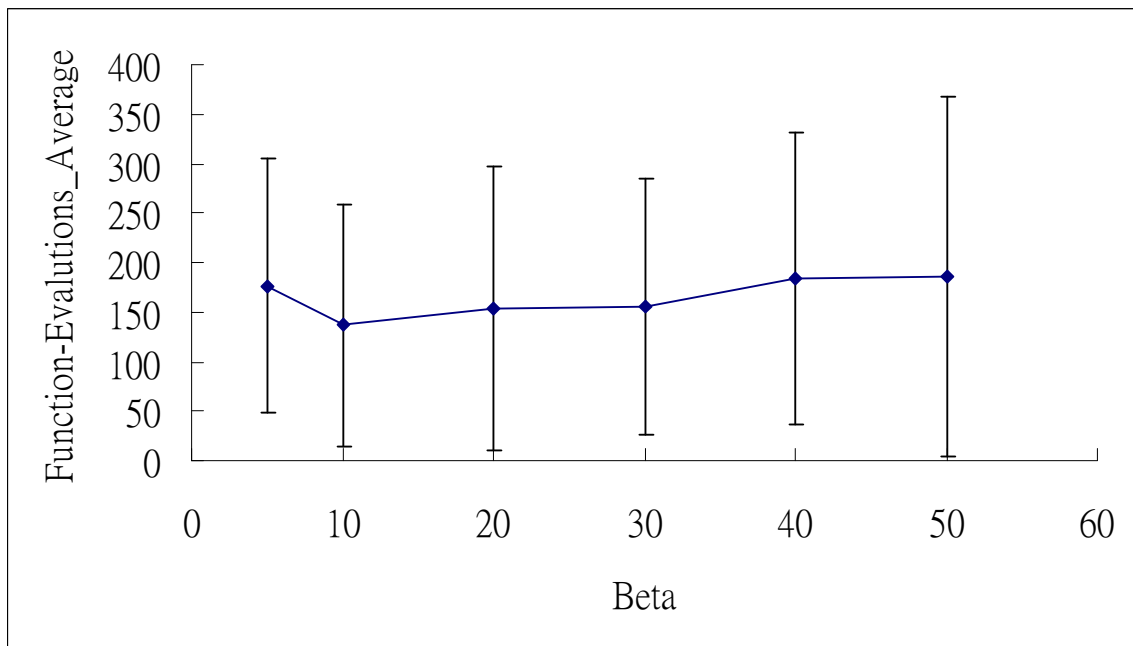


圖 4-28 β 對 QGSES 在 $x\sin(1/x)$ function with 2 variables 中搜尋效率的影響

表 4-24 與圖 4-29 為 $x\sin(1/x)$ function 在最佳的 μ 、 λ 與 β 的設定下 ($\mu=1$ ， $\lambda=3$ ， $\beta=10\sigma$)，測試不同的 σ 值對 QGSES 演算法的搜尋效率的影響，實驗結果顯示最佳的 σ 的參數設定值為 $\sigma=0.02$ ，此時 QGSES 演算法有最佳的搜尋效率。

表 4-24 應變數 σ 在 $x\sin(1/x)$ function with 2 variables 中的實驗數據

Sigma	Best-Value		Time (sec.)		Function-Evaluations	
	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
0.005	-0.348663228	0.104306665	3.0616	3.256852261	27704.92	30112.53731
0.01	-0.434035051	0.000328731	0.0875	0.048531601	214	443.4540217
0.02	-0.433995153	0.000312071	0.07374	0.009964857	80.92	73.24876261
0.03	-0.433913525	0.000317251	0.08116	0.013710357	160.6	130.8596382
0.04	-0.433909322	0.000302246	0.09166	0.025862754	250	234.3223839
0.05	-0.433878382	0.000307399	0.11244	0.051308654	444.28	468.8887009

備註：自變數 $\mu=1$ 、 $\lambda=3$ 、 $\beta=10\sigma$

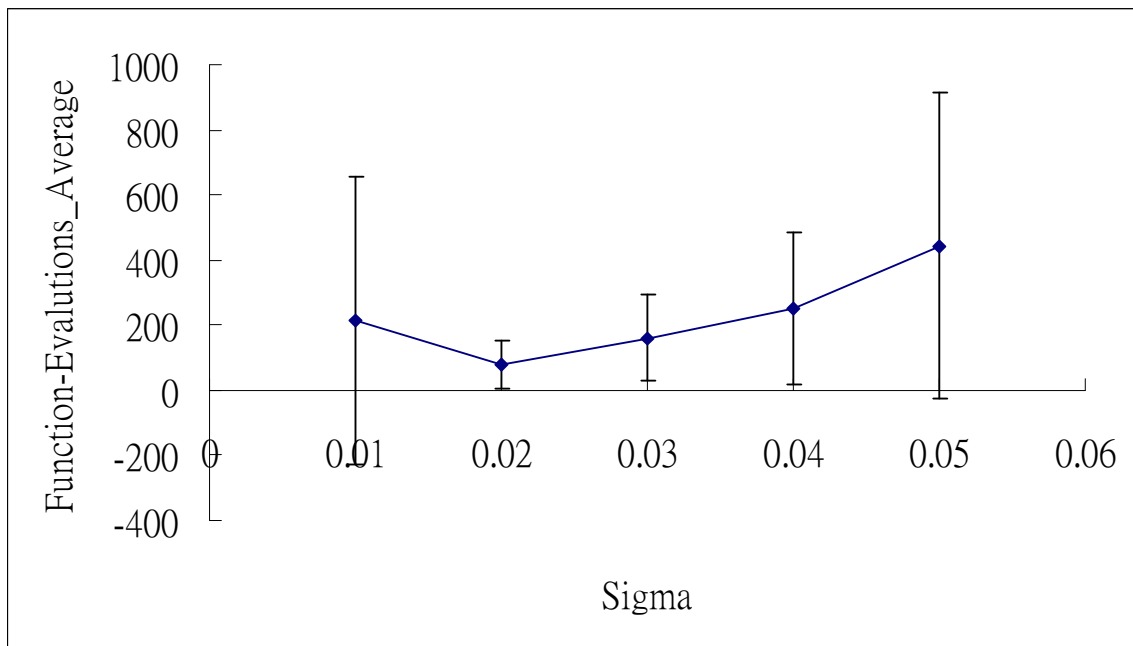


圖 4-29 σ 對 QGSES 在 $x\sin(1/x)$ function with 2 variables 中搜尋效率的影響

從以上的數據顯示 QGSES 演算法在 $x\sin(1/x)$ function 中最佳的參數設計為 $\mu=1$ 、 $\lambda=3$ 、 $\sigma=0.02$ 、 $\beta=10\sigma$ ，在最佳的參數設計下，此時 QGSES 演算法有最佳的搜尋效率，收斂百分比為 100%，達收斂時的平均運算次數為 80.92 次，達收斂時的平均次數之標準差為 73.24876261。

(4) QGSES 與其他搜尋方法的比較

表 4-25 為 $x\sin(1/x)$ function with 2 variables 在 QGSES、CMA、Y-L 與 G3PCX 與四種方法的實驗結果，測試次數皆為 50 次的重複試驗。圖 4-30 為 $x\sin(1/x)$ function with 2 variables 在 QGSES、CMA、Y-L 與 G3PCX 與四種方法之平均運算次數比較圖。由表 4-25 中的實驗數據顯示，在有多重區域解的函數中，QGSES 有最好的收斂效率，而且有 100% 的收斂比率，而 CMA 的收斂比率在 $x\sin(1/x)$ function with 2 variables 為 66%，G3PCX 的收斂比率為 60%，表示在 $x\sin(1/x)$ function with 2 variables 中 CMA 與 G3PCX 會有陷入區域解的情況。Y-L 雖然有 100% 的收斂比率，然而其收斂效率與 QGSES 相比相對較差。

表 4- 25 不同的演算法對 $x\sin(1/x)$ function with 2 variables 的測試結果

Method	Convergence Percentage	Best-Value		Time (sec.)		Function-Evaluations	
		Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
QGSES	100%	-0.433995153	0.000312071	0.07374	0.009964857	80.92	73.24876261
CMA	66%	-0.381288515	0.082227198	1.83314	2.492765352	20457.36	28669.51733
Y-L	100%	-0.434217831	0.000209121	79.782	142.4367016	5663.46	6977.077709
G3PCX	60%	-0.424943951	0.023486203	3.65092	4.410065976	8058.36	9890.541354

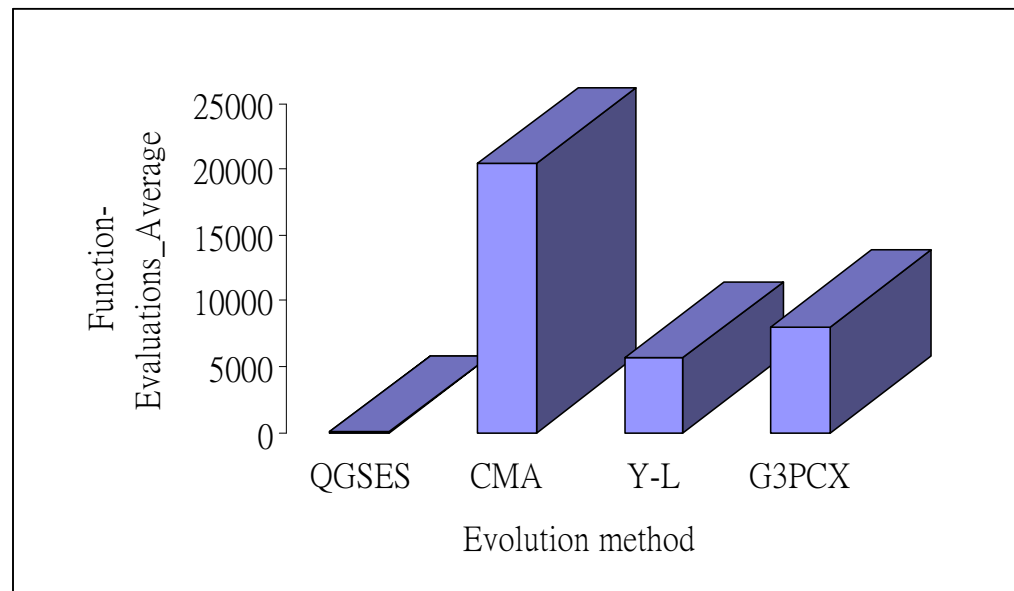


圖 4- 30 $x\sin(1/x)$ function with 2 variables 之平均運算次數比較圖

4.1.6 函數 6：xsin(1/x) function with 5 variables

(1) 目標函數

$$\text{Min. } F(X) = \sum_{i=1}^5 x_i \sin\left(\frac{1}{x_i}\right)$$

where $x_i \in [0, 0.25]$

此函數目的在求函數的最小值，變數數量為 5 個變數，函數之理想(Ideal)最佳值為-1.08。此函數有無限多個區域最佳解。

(2) 測試環境與參數設定

QGSES 演算法在 xsin(1/x) function with 5 variables 之相關變數設定如下：

- a. 母體 μ 的數量設定：在本函數中母體 μ 的數量分別為 1、2、4、6、8、10。
- b. 子代 λ 的數量設定：本研究架構的實驗中，子代 λ 的數量設定直接影響每一代產生的子代數目，以及每一代移動的子代數量。在本函數中子代 λ 的數量為 1~10。
- c. 突變強度 σ ：在本函數中初始的突變強度 σ 為 0.03，突變強度 σ 分別為 0.005、0.01、0.02、0.03、0.04、0.05。
- d. β 值：在本函數中初始的 β 值為 10σ ， β 值的分別為 5σ 、 10σ 、 20σ 、 30σ 、 40σ 、 50σ 。
- e. 收斂條件：在本範例中，當與最佳值的誤差小於 10^{-3} ，則視為收斂。
- f. 測試次數：50 次。
- g. 最大演化代數：100000 代。

(3) QGSES 實驗結果

圖 4-31 為 xsin(1/x) function 在固定 $\beta = 10\sigma$ 與 $\sigma = 0.03$ 下，不同的 μ 與 λ 對 QGSES 演算法的搜尋效率的影響，詳細數據請參考附錄，由圖 4-25 可知在固定的 β 與 σ 設定下，最佳的 μ 與 λ 的參數設定值為 $\mu = 1$ 、 $\lambda = 3$ ，此時 QGSES 演算法有最佳的搜尋效率，收斂百分比為 100%，達收斂時的平均運算次數為 2909.56 次，達收斂時的平均次數之標準差為 2299.6676615262。

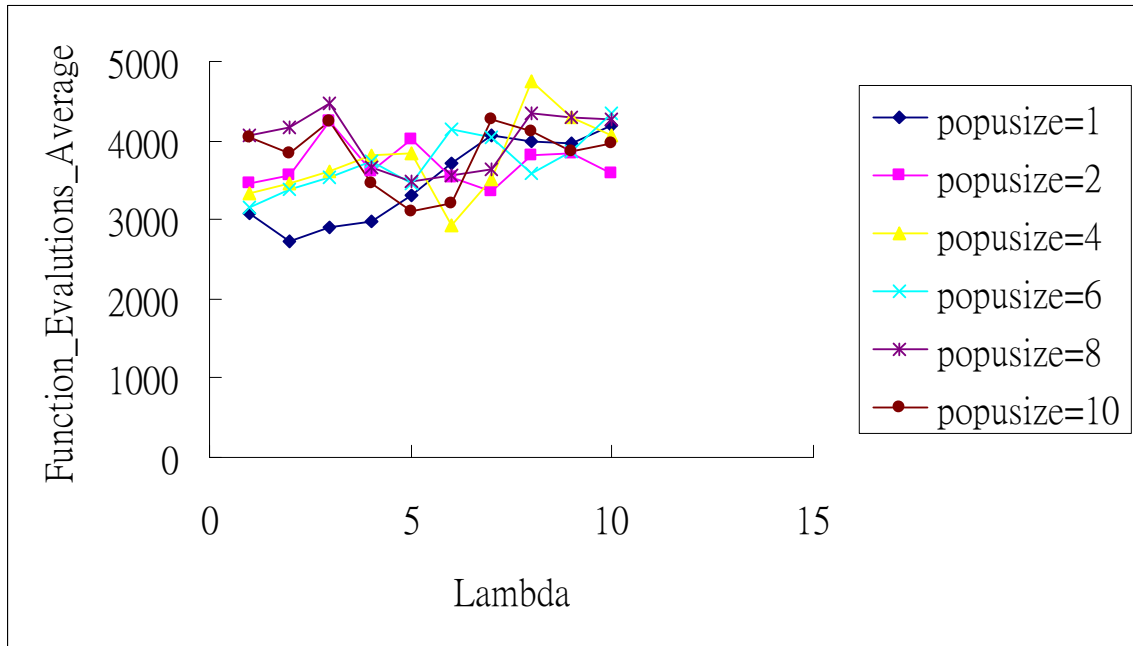


圖 4-31 μ 與 λ 對 QGSES 在 $x\sin(1/x)$ function with 5 variables 中搜尋效率的影響

表 4-26 與圖 4-32 為 $x\sin(1/x)$ function 在固定 $\mu=1$ 、 $\beta=10\sigma$ 與 $\sigma=0.03$ 下，不同的 λ 對 QGSES 演算法的搜尋效率的影響，在固定的 μ 、 β 與 σ 設定下，最佳的 λ 的參數設定值為 $\lambda=3$ ，此時 QGSES 演算法有最佳的搜尋效率。

表 4-26 應變數 λ 在 $x\sin(1/x)$ function with 5 variables 中的實驗數據

Lambda	Best-Value		Time (sec.)		Function-Evaluations	
	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
1	-1.081088881	0.001595641	0.68334	0.54896967	3074.24	2890.066136
2	-1.081258014	0.001446598	0.41224	0.339895095	2728.6	2269.381252
3	-1.081294575	0.001426406	0.461	0.314522833	2909.56	2299.667662
4	-1.081164714	0.001541508	0.44998	0.380711873	2970.12	2940.868348
5	-1.081011986	0.001656491	0.47754	0.396750022	3313.4	3190.919233
6	-1.080878706	0.001414522	0.51528	0.413394314	3715.72	3386.493358
7	-1.080622318	0.001301868	0.55442	0.466321256	4055.68	3847.216022
8	-1.081149063	0.001565062	0.53814	0.486815098	3997.8	4126.736715
9	-1.080810946	0.001226899	0.5295	0.490242974	3952.72	4157.419058
10	-1.081311306	0.001784399	0.54658	0.504555415	4184.2	4336.770848

備註：自變數 $\mu=1$ 、 $\sigma=0.03$ 、 $\beta=10\sigma$

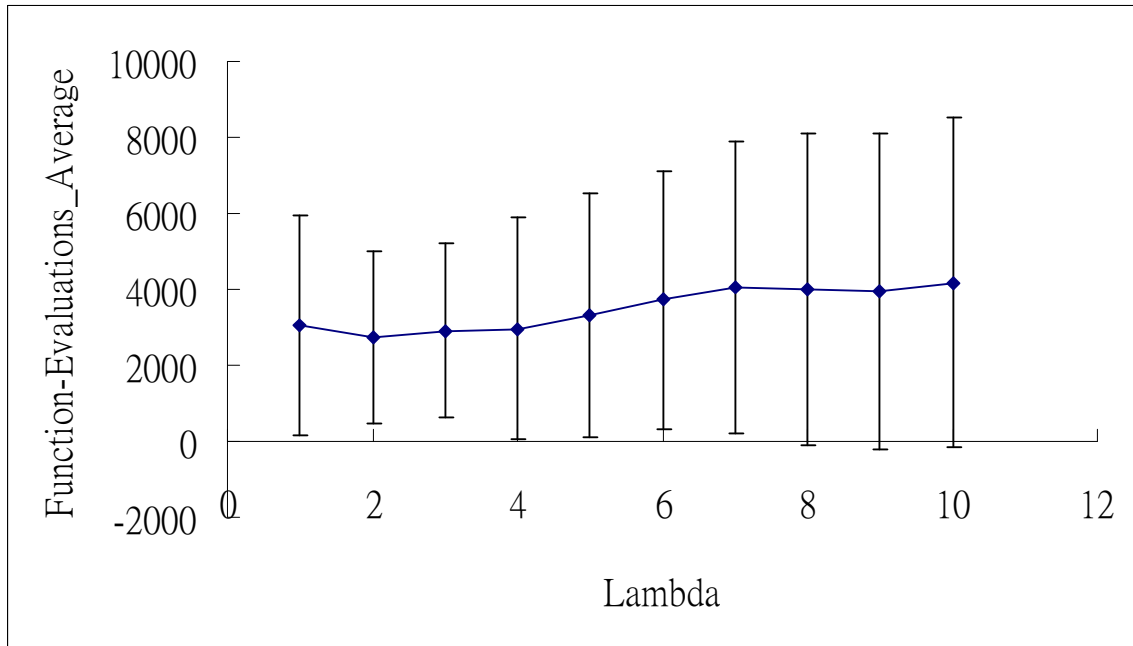


圖 4-32 λ 對 QGSES 在 $x\sin(1/x)$ function with 5 variables 中搜尋效率的影響

表 4-27 與圖 4-33 為 $x\sin(1/x)$ function 在固定 $\lambda=3$ 、 $\beta=10\sigma$ 與 $\sigma=0.03$ 下，不同的 μ 對 QGSES 演算法的搜尋效率的影響，在固定的 λ 、 β 與 σ 設定下，最佳的 μ 的參數設定值為 $\mu=1$ ，此時 QGSES 演算法有最佳的搜尋效率。

表 4-27 應變數 μ 在 $x\sin(1/x)$ function with 5 variables 中的實驗數據

Population size	Best-Value		Time (sec.)		Function-Evaluations	
	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
1	-1.081294575	0.001426406	0.461	0.314522833	2909.56	2299.667662
2	-1.081010122	0.001550047	0.5542	0.389116844	4252.4	3390.389972
4	-1.081480416	0.001554876	0.44584	0.399049859	3619.36	3761.421821
6	-1.081208671	0.001674854	0.4235	0.291806495	3529.68	2858.139237
8	-1.081729809	0.001666603	0.51354	0.46389171	4467.2	4582.103251
10	-1.080750977	0.001324436	0.48502	0.350284869	4232.8	3489.337719

備註：自變數 $\lambda=3$ 、 $\sigma=0.03$ 、 $\beta=10\sigma$

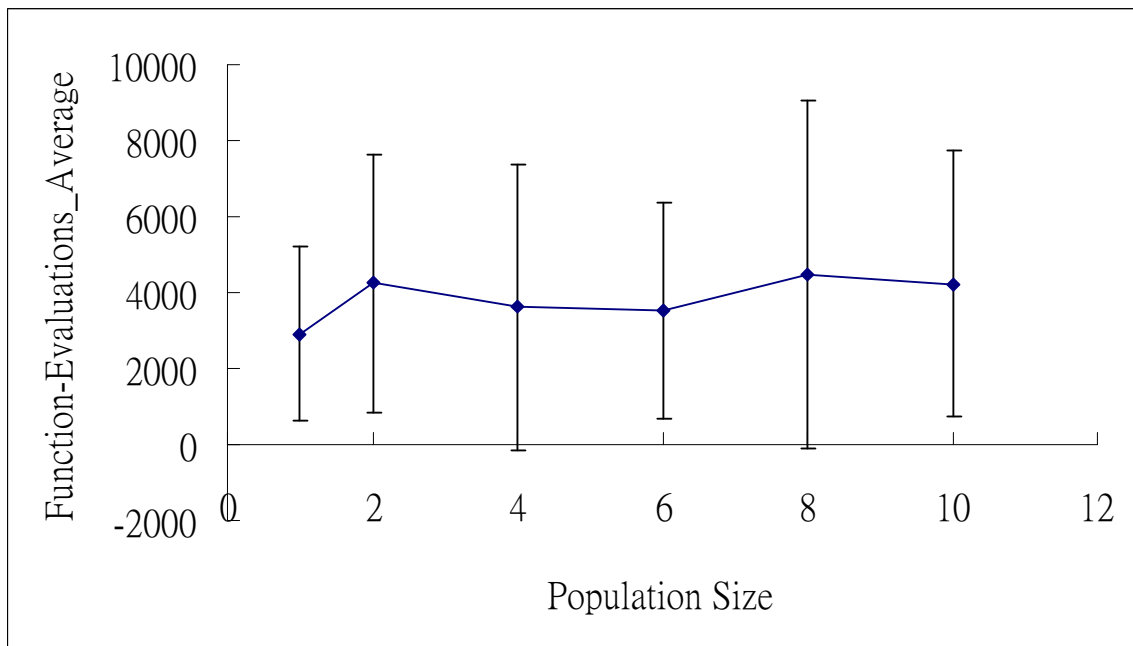


圖 4-33 μ 對 QGSES 在 $x\sin(1/x)$ function with 5 variables 中搜尋效率的影響

表 4-28 與圖 4-34 為 $x\sin(1/x)$ function 在最佳的 μ 與 λ 的設定下 ($\mu=1$, $\lambda=3$), 固定 $\sigma=0.03$, 測試不同的 β 值對 QGSES 演算法的搜尋效率的影響, 實驗結果顯示最佳的 β 的參數設定值為 $\beta=20\sigma$, 此時 QGSES 演算法有最佳的搜尋效率。

表 4-28 應變數 β 在 $x\sin(1/x)$ function with 5 variables 中的實驗數據

Beta	Best-Value		Time (sec.)		Function-Evaluations	
	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
5σ	-1.081128847	0.001860287	0.6269	0.456111124	4046.44	3304.703934
10σ	-1.081205782	0.001545819	0.56412	0.427629137	3596.2	3055.166891
20σ	-1.08098876	0.001472786	0.47304	0.417518615	2957.2	3017.065969
30σ	-1.080991038	0.001395964	0.50936	0.461781986	3217.12	3347.197071
40σ	-1.081256561	0.001574638	0.53504	0.3785242	3394.36	2737.129777
50σ	-1.081286674	0.001613241	0.62358	0.46839762	4030.84	3376.915894

備註：自變數 $\mu=1$ 、 $\lambda=3$ 、 $\sigma=0.03$

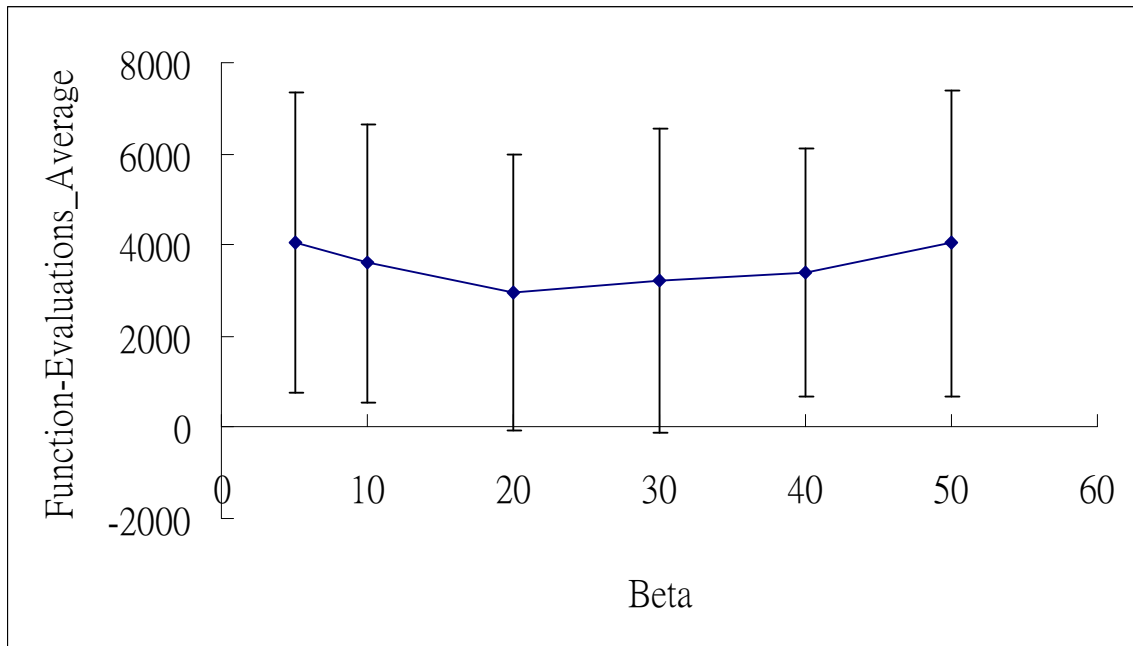


圖 4-34 β 對 QGSES 在 $x\sin(1/x)$ function with 5 variables 中搜尋效率的影響

表 4-29 與圖 4-35 為 $x\sin(1/x)$ function 在最佳的 μ 、 λ 與 β 的設定下 ($\mu=1$, $\lambda=3$, $\beta=20\sigma$)，測試不同的 σ 值對 QGSES 演算法的搜尋效率的影響，實驗結果顯示最佳的 σ 的參數設定值為 $\sigma=0.02$ ，此時 QGSES 演算法有最佳的搜尋效率。

表 4-29 應變數 σ 在 $x\sin(1/x)$ function with 5 variables 中的實驗數據

Sigma	Best-Value		Time (sec.)		Function-Evaluations	
	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
0.005	-1.021383039	0.143893429	13.10808	30.15567077	96094.96	222155.4011
0.01	-1.079049842	0.017238981	1.72408	11.63853624	12094.48	84839.33458
0.02	-1.081210235	0.001595584	0.13346	0.051521824	504.28	369.0368623
0.03	-1.080911048	0.001552214	0.5074	0.425787913	3222.88	3098.402601
0.04	-1.081413047	0.001679267	1.79564	1.704090215	12616.24	12388.83408
0.05	-1.081615684	0.001778362	5.1851	4.371363553	37295.92	31832.42087

備註：自變數 $\mu=1$ 、 $\lambda=3$ 、 $\beta=10\sigma$

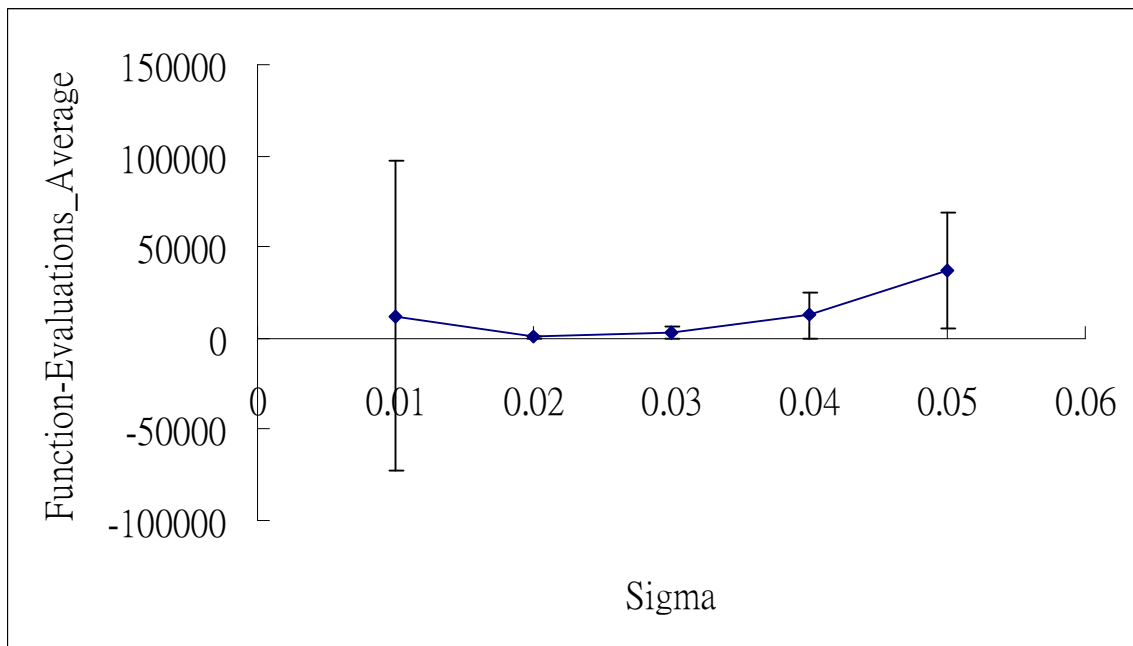


圖 4-35 σ 對 QGSES 在 $x\sin(1/x)$ function with 5 variables 中搜尋效率的影響

從以上的數據顯示 QGSES 演算法在 $x\sin(1/x)$ function 中最佳的參數設計為 $\mu=1$ 、 $\lambda=3$ 、 $\sigma=0.02$ 、 $\beta=20\sigma$ ，在最佳的參數設計下，此時 QGSES 演算法有最佳的搜尋效率，收斂百分比為 100%，達收斂時的平均運算次數為 504.28 次，達收斂時的平均次數之標準差為 369.036862270269。

(4) QGSES 與其他搜尋方法的比較

表 4-30 為 $x\sin(1/x)$ function with 5 variables 在 QGSES、CMA、Y-L 與 G3PCX 與四種方法的實驗結果，測試次數皆為 50 次的重複試驗。圖 4-36 為 $x\sin(1/x)$ function with 5 variables 在 QGSES、CMA、Y-L 與 G3PCX 與四種方法之平均運算次數比較圖。由表 4-30 中的實驗數據顯示，在有多重區域解的函數中，QGSES 有最好的收斂效率，而且有 100% 的收斂比率，而 CMA 的收斂比率在 $x\sin(1/x)$ function with 5 variables 為 4%，G3PCX 的收斂比率為 4%，表示在 $x\sin(1/x)$ function with 5 variables 中 CMA 與 G3PCX 會造成陷入區域解的情況。Y-L 雖然有 100% 的收斂比率，然而其收斂效率與 QGSES 相比相對較差。

表 4- 30 不同的演算法對 $x\sin(1/x)$ function with 5 variables 的測試結果

Method	Convergence Percentage	Best-Value		Time (sec.)		Function-Evaluations	
		Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
QGSES	100%	-1.081210235	0.001595584	0.13346	0.051521824	504.28	369.0368623
CMA	4%	-0.716284273	0.190052498	89.93476	18.53835477	768008.64	158316.1757
Y-L	100%	-1.079928911	0.000796999	231.8436	608.3671151	8889.22	15592.62603
G3PCX	4%	-0.853208388	0.113155753	88.20338	18.14898229	192059.48	39542.819

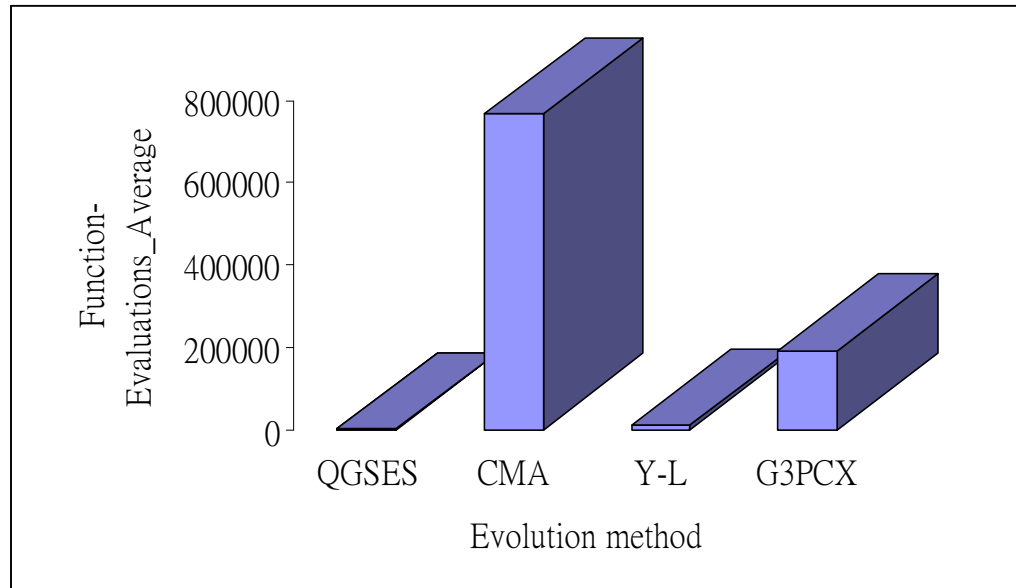


圖 4- 36 $x\sin(1/x)$ function with 5 variables 之平均運算次數比較圖

4.1.7 函數 7：xsin(1/x) function with 10 variables

(1) 目標函數

$$\text{Min. } F(X) = \sum_{i=1}^{10} x_i \sin\left(\frac{1}{x_i}\right)$$

where $x_i \in [0, 0.25]$

此函數目的在求函數的最小值，變數數量為 10 個變數，函數之理想 (Ideal) 最佳值為 -2.17。此函數有無限多個區域最佳解。

(2) 測試環境與參數設定

QGSES 演算法在 xsin(1/x) function with 10 variables 之相關變數設定如函數 6 之最佳設定 $\mu=1$ 、 $\lambda=3$ 、 $\sigma=0.02$ 、 $\beta=20\sigma$ ，收斂條件為與最佳值的誤差小於 10^{-2} 則視為收斂，測試次數為 50 次，最大演化代數為 1000000 代。

(3) QGSES 與其他搜尋方法的比較

表 4-31 為 xsin(1/x) function with 10 variables 在 QGSES、CMA、Y-L 與 G3PCX 與四種方法的實驗結果，測試次數皆為 30 次的重複試驗。圖 4-37 為 xsin(1/x) function with 10 variables 在 QGSES、CMA、Y-L 與 G3PCX 與四種方法之平均運算次數比較圖。由表 4-31 中的實驗數據顯示，在有多重區域解的函數中，QGSES 有最好的收斂效率，而且有 100% 的收斂比率，而 CMA 的收斂比率在 xsin(1/x) function with 10 variables 為 0%，G3PCX 的收斂比率為 0%，表示在 xsin(1/x) function with 10 variables 中 CMA 與 G3PCX 會有陷入區域解的情況。

表 4-31 不同的演算法對 $x\sin(1/x)$ function with 10 variables 的測試結果

Method	Convergence Percentage	Best-Value		Time (sec.)		Function-Evaluations	
		Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
QGSES	100%	-2.162005383	0.002009061	17.35356667	15.86193327	92641.8	85413.32518
CMA	0%	-1.416086332	0.285492487	1201.588667	16.66723065	10000000	0
Y-L	0%	-2.145777675	0.005852162	160476.09	3207.923388	479951	0
G3PCX	0%	-1.38557	0.162189	947.5093	3.49613	2000050	0

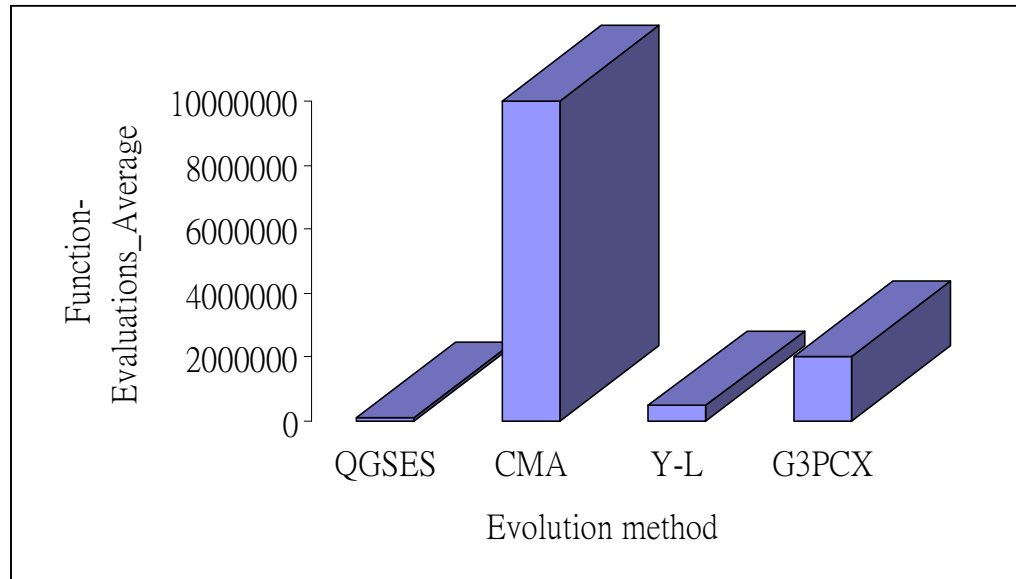


圖 4-37 $x\sin(1/x)$ function with 10 variables 之平均運算次數比較圖

4.2 實驗結論

4.2.1 QGSES 與其他演算法的搜尋效率比較

本研究提出不同的函數搭配不同的方法組合來進行驗證，針對本研究所驗證的部分可得到以下的結論：

1. 在上述 7 個函數的實驗結果中可得知，QGSES 所測試出來的結果確實比 Y-L 的架構來的好很多。雖然在單一最佳解的函數中 QGSES 的收斂效率較 CMA 與 G3PCX 略差，但是在有多重區域解的函數中 QGSES 可以 100% 找出全域最佳解，而 CMA 與 G3PCX 都有落入區域解而無法跳脫的情形。
2. 在單一最佳解的函數中雖然 QGSES 的收斂效率較 CMA 與 G3PCX 略差，但是隨著函數變數數量增加，QGSES 與 CMA、G3PCX 之間收斂效率的差距快速的縮小，顯示出 QGSES 根據母體的梯度向量與母體歷史移動向量判斷子代往最佳解區域逼近的最佳方向是有效的，此外，依據移動參數 ξ 加速演化策略整體的演化速度的方法在多變數的函數中效果較顯著。
3. 在有多重區域解的函數中，QGSES 有最好的收斂效率，而且有 100% 的收斂比率，而 CMA 與 G3PCX 的收斂比率則隨著區域解的增加快速的下降，顯示出在多重區域解函數中 QGSES 演算法有著優秀的全域搜尋能力，此外，在函數 4 中若不考慮 G3PCX 陷入區域解的情況，QGSES 的收斂效率依然優於 G3PCX。
4. 由函數 5 到函數 7 的實驗數據中，在有多重區域解的函數中，隨著變數數量的增加，QGSES 為唯一依然可以百分之百收斂的演算法，而 CMA 與 G3PCX 的收斂比率則隨著區域解的增加快速的下降，顯示出在多重區域解函數中 QGSES 演算法穩定的的全域搜尋能力。

4.2.2 QGSES 與其他演算法的搜尋原理分析

根據以上的實驗數據，本研究對 QGSES、Y-L、CMA 與 G3PCX 四種方法在不同特性的函數中的搜尋效能提出以下的分析：

1. 在單一最佳解的函數中，CMA 以 Covariance matrix 記錄整個族群數代演進的歷史，計算出目標變數變異數與共變數，藉此縮小突變強度，達到快速收斂的目的。G3PCX 在 PCX 重構方法下，產生的子代會圍繞在

最佳的母體周圍，因此在單一最佳解函數的細部搜尋上有相當好的效率。QGSES 由於移動參數 ξ 對於子代的移動方向與距離無法作非常精準的判定，因此在作細部搜尋時必須多次的修正，因此在單一最佳解的函數中 QGSES 的搜尋效率略遜於 CMA 與 G3PCX。Y-L 承襲基因演算法的缺點，從最佳解附近區域逼近到最佳解上時，需要耗費很多的時間去演化最佳的解，因此在單一最佳解的函數中 Y-L 的搜尋效率與其他方法相比相對較差。

2. 在單一最佳解的函數中，隨著函數變數數量增加，QGSES 與 CMA、G3PCX 之間收斂效率的差距快速的縮小，原因在於 CMA 與 G3PCX 擁有隨機搜尋的特性，函數變數數量增加造成隨機產生的子代命中率降低，顯示出 QGSES 依據移動參數 ξ 判斷子代的移動方向與距離的方法在多變數的函數中效果較顯著。
3. 在有多重區域解的函數中，由於 QGSES 使用固定的突變強度，只要突變強度的設定值不要太小，QGSES 在陷入區域解時就保有一定的跳脫能力，因此 QGSES 在函數 3 與函數 4 中能達到 100%的收斂比率。CMA 與 G3PCX 的收斂比率隨著區域解的增加快速的下降，原因在於 CMA 以縮小突變強度來達到快速收斂的目的，因此當母體陷入區域解時，太小的突變強度使的 CMA 無法跳脫區域解，例如在函數 4 中，跳脫所有區域解所需的最小突變強度約為 0.1，而 CMA 陷入區域解時突變強度往往達到 $10^{-7} \sim 10^{-9}$ 之間，因而 CMA 在多重區域解的函數中容易陷入區域解。G3PCX 由於產生的子代會圍繞在最佳的母體周圍的特性，當最佳的母體陷入區域解時，產生的子代跳脫區域解的能力相對較差。綜合以上結果，顯示出在多重區域解函數中 QGSES 演算法有著較佳的全域搜尋能力。
4. 由函數 5 到函數 7 的實驗數據中，在有多重區域解的函數中，隨著變數數量的增加，QGSES 為唯一依然可以百分之百收斂的演算法。由於 $x\sin(1/x)$ function 大部分的區域解深度較淺，因此 CMA 與 G3PCX 在函數 5 中在變數較少的情況下有一定的機率可以跳脫區域解，因而 CMA 與 G3PCX 在函數 5 中的收斂比率約為 60%。隨著變數數量的增加，CMA 與 G3PCX 在函數 6 中的收斂比率快速下降到 4%，在函數 7 中 10 個變數的 $x\sin(1/x)$ function 設定下，CMA、G3PCX 與 Y-L 的收

斂比率皆為 0%，而 QGSES 的收斂比率依然為 100%。由函數 5 到函數 7 的實驗數據中可以顯示出在多重區域解函數中，隨著變數數量的增加，QGSES 演算法穩定的全域搜尋能力。

第五章 結論與建議

5.1 研究總結

本研究提出的 QGSES 主要重點在於母體搜尋最佳解時會應先將周圍的資訊先比較過後再決定子代突變的方向與距離，而非演化策略中子代個別隨機突變的運作方式，藉由這樣的機制使整體族群在搜尋時能快速且有效的得到最佳解所在位置的資訊並向附近的區域快速的包圍住，以達到快速收斂的目的，並以 7 個函數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. QGSES 保留演化策略中突變的方法，並提出移動參數 ξ 的概念，根據母體隨機突變的子代判斷母體梯度向量，再經由母體的梯度資訊與母體歷史的移動向量的綜合判斷，決定出子代應該跳躍的方向，再以移動參數 ξ 決定移動的距離，以期望可以更有效率的作多變量的函數全域最佳解搜尋與快速的收斂。
2. 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函數的效果比 Y-L 所提出的架構有更佳的結果。在有多重區域解的函數中，QGSES 有最好的收斂效率，100%的收斂比率顯示出在多重區域解函數中 QGSES 優秀的全域搜尋能力，此外，在函數 3 到函數 7 中更顯示出 QGSES 優秀的收斂效率。
3. 本研究所提出的架構擁有優秀的多重區域解函數的全域搜尋能力以及穩定的最佳解收斂效率，是一優良的演算法。

5.2 後續研究建議

本研究目的在發展出一套新的演算法演算機制，經由母體的梯度資訊與母體歷史的移動向量的綜合判斷，決定出子代的移動方向與距離，以期望可以更有效率的作多變量的函數全域最佳解搜尋與快速的收斂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. 突變強度的自我學習能力

本研究在演算法中使用固定的突變強度，若將突變強度學習模式導入本研究之演算法中，可以在演算的過程中自動調整，相信對於實驗結果或許會有幫助。

2. 演算法之合併

本研究提出的是依全新的演化策略的架構，若可以與類神經網路或模糊理論合併，根據母體的梯度資訊更正確的預測子代的移動方向與距離，相信對於搜尋效率會有幫助。

参考文献

- [1] Arabas, J., Mulawka, J. and Pokrasniewicz, J., “A new class of the crossover operators for the numerical optimization,” *Proceedings of the 6th International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo, CA, pp. 42-48, 1995.
- [2] Bäck, T., *Evolutionary Algorithms in Theory and Practice*. New York: Oxford Univ. Press, 1996.
- [3] Beyer, H.-G., “Toward a theory of evolution strategies: On the benefit of sex—the $(\mu/\mu, \sigma)$ -Theory,” *Evol. Comput.*, vol. 3, no. 1, pp. 81–111, 1995.
- [4] Beyer, H.-G. “Toward a theory of evolution strategies: Self-adaptation,” *Evolutionary Computation Journal* 3(1), 81-111, 1995b.
- [5] Beyer, H.-G., “Toward a theory of evolution strategies: Self-adaptation,” *Evolutionary Computation*, vol. 3, no. 3, pp. 311–348, 1995.
- [6] Cantù-Paz, E. ., *A summary of research on parallel genetic algorithms*. IlliGAL Report No. 95007, University of Illinois at Urbana-Champaign, July, 1995.
- [7] Carlos Andrés Peña-Reyes, Moshe Sipper, “Evolutionary Computation in Medicine: an overview,” *Artificial Intelligence in Medicine*, 19, pp. 1-23, 2000.
- [8] Colomi, A., Dorigo, M. and Maniezzo, V., “An investigation of some properties of an ant algorithm,” *Proceedings of the Parallel Problem Solving from Nature Conference*, R. Manner and B. Manderick Eds. Brussels, Belgium: Elsevier, pp.509-520, 1992.
- [9] Colomi, A., Dorigo, M. and Maniezzo, V., “Distributed optimization by ant colonies,” *Proceedings of the First European Conference Artificial Life*, F. Varela and P. Bourguine, Eds. Paris, France: Elsevier, pp.134-142, 1991.
- [10] Darrell Whitley, L. and Kauth, J., GENITOR: a different genetic algorithm, *Proceedings of the 1988 Rocky Mountain Conference on Artificial Intelligence*, 1988.
- [11] Darrell Whitley, L. and Starkweather, T., “GENITOR II: a distributed genetic algorithm,” *Journal of Experimental and Theoretical Artificial Intelligence* 2, pp. 189-214, 1990.
- [12] Darrell Whitley, L., The GENITOR algorithm and selective pressure: why rank based allocation of reproductive trials is best, in: J. D. Schaffer (Ed.), *Proceedings of the Third International Conference on GAs*, Morgan Kaufmann, Los Atlos, CA, pp. 116-121., 1989
- [13] Deb, K. and Agrawal, R. B., “Simulated binary crossover for continuous search space.” *Complex Systems*, 9(2):115–148, 1995.
- [14] Deb, K., Anand, A. and Joshi, D., “A Computationally Efficient Evolutionary Algorithm for Real-Parameter Optimization,” *Evolutionary Computation* 10(4), 371-395, 2002.
- [15] Dorigo, M., Bonabeau, E. and Theraulaz, G., “Ant algorithm and stigmergy,” *Future Generation Computer Systems*, 16, pp. 851-871, 2000.
- [16] Dorigo, M., Caro, G.D. and Gambarsella, L.M., “Ant algorithms for discrete

- optimization,” *Artificial Life*, 5, pp. 137-172, 1999.
- [17] Dorigo, M., Maniezzo, V. and Colormi, A., “Positive feedback as a research strategy,” *Technology Report 91-016*, Politecnico di Milano, 1991.
- [18] Dorigo, M. and Maria, L., “Ant colony system: a cooperative learning approach to the traveling salesman problem,” *IEEE Transactions evolution Computer*, 1, pp.53–66, 1997.
- [19] Eshelman, L., The CHC adaptive search algorithm. How to have safe search when engaging in nontraditional genetic recombination, in: *G. Rawlins (Eds.), FOGA-1, Morgan Kaufmann, Los Atlos, CA*, pp. 265-283, 1991.
- [20] Eshelman, L. J. and Schaffer, D., Preventing premature convergence in genetic algorithms by preventing incest, in: L. Booker, R. Belew (Eds.), *Proceedings of the Fourth International Conference on GAs*, Morgan Kaufmann, Los Atlos, CA, 1991.
- [21] Eshelman, L. J. and Schaffer, J. D., Real-coded genetic algorithms and intervalschemata. In *Foundations of Genetic Algorithms 2 (FOGA-2)*, pp.187-202, 1993.
- [22] Fogel, L. J., “Autonomous automata,” *Ind. Res.*, vol. 4, pp. 14–19, 1962.
- [23] Fogel, L. J., “On the organization of intellect,” Ph.D. dissertation, University of California, Los Angeles, 1964.
- [24] Fogel, L. J., Owens, A. J., & Walsh, M. J., *Artificial intelligence through simulated evolution*. New York: Wiley, 1966.
- [25] Goldberg, D. E., *Genetic Algorithms in Search, Optimazation and Machine Learning*, Addison-Wesley, Reading, MA, 1989.
- [26] Goldberg, D. E., Real-coded genetic algorithms, virtual alphabets, and blocking. *Complex Systems 5(2)*, 139-168, 1991.
- [27] Goldberg, D. E., Deb, K. and Korb, B., Messy genetic algorithms revisited: Nonuniform size and scale. *Complex Systems 4(4)*, pp. 415-444, 1990.
- [28] Goldberg, D. E. and Deb, K., A comparison of selection schemes used in genetic algorithms. In *Foundations of Genetic Algorithms 1 (FOGA-1)*, pp. 69-93, 1991.
- [29] Goldberg, D. E., Korb, B. And Deb, K., Messy genetic algorithms: Motivation, analysis and first results. *Complex Systems 3(5)*, pp. 493-530, 1989.
- [30] Hansen, N. and Ostermeier, A., “Adapting arbitrary normal mutation distributions in evolution strageties: The covariance matrix adaptation.” In *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp. 312-317, 1996.
- [31] Hansen, N. and Ostermeier, “Completely Derandomized Self-Adaptation in Evolution Strategies,” *Evolutionary Computation 9(2)*, 159-195, 2001.
- [32] Higuchi, T., Tsutsui, S., and Yamamura, M., “Theoretical analysis of simplex crossover for real-coded genetic algorithms.” In Schoenauer, M. et al., editors, *Parallel Problem Solving from Nature (PPSN-VI)*, pages 365–374, Springer, Berlin, Germany, 2000.
- [33] Holland, J. H., “Outline for a logical theory of adaptive systems,” *J. Assoc. Comput. Mach.*, vol. 3, pp. 297–314, 1962.

- [34] Holland, J. H., *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. of Michigan Press, 1975.
- [35] Holland, J. H. and Reitman, J. S., "Cognitive systems based on adaptive algorithms," in *Pattern-Directed Inference Systems*, D. A. Waterman and F. Hayes-Roth, Eds. New York: Academic, 1978.
- [36] Koza, JR., *Genetic programming: a paradigm for genetically breeding populations of computer programs to solve problems*. Technical Report STANCS-90-1314, Department of Computer Science, Stanford University, June 1990.
- [37] Koza, JR., *Genetic Programming*. Cambridge, MA: MIT Press, 1992.
- [38] Koza, John R., Bennett, Forrest H. III and Andre, David, Classifying proteins as extracellular using programmatic motifs and genetic programming, *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*, pp. 212-217, 1998.
- [39] Mitsuo Gen, Runwei Cheng, *Genetic Algorithms and Engineering Design*, Wiley, New York, 1997.
- [40] Ono, I. and Kobayashi, S., "A real-coded genetic algorithm for function optimization using unimodal normal distribution crossover." In Bäck, T., editor, *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA-7)*, pages 246–253, Morgan Kaufmann, San Francisco, California, 1997.
- [41] Pierreval, H., Caux, C., Paris J.L. and Viguier, F., "Evolutionary Approaches to the Design and Organization of Manufacturing Systems," *Computers & Industrial Engineering*, 44, pp. 339-364, 2003.
- [42] Rechenberg, I., *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Stuttgart, Germany: Frommann-Holzboog, 1973.
- [43] Rechenberg, I., *Evolutionstrategie '94*. Frommann-Holzboog, Stuttgart, 1994.
- [44] Schwefel, H.-P., "Kybemetische evolution als strategie der experimentellen forschung in der strmungstechnik," Diploma thesis, Technical Univ. of Berlin, 1965.
- [45] Schwefel, H.-P., *Numerical Optimization of Computer Models*. Chichester: Wiley, 1981.
- [46] Schwefel, H.-P. "Collective intelligence in evolving systems," in W. Wolff, C. J. Soeder and F. Drepper (Eds), *Ecodynamics – Contributions to Theoretical Ecology*, pp. 95-100. Berlin: Springer, 1987a.
- [47] Schwefel, H.-P., *Evolution and Optimum Seeking*. New York: Wiley, 1995 (Sixth-Generation Computer Technology Series).
- [48] Schwefel, H.-P., *Evolution and Optimum Seeking*, Wiley , New York, 1995.
- [49] Schwefel, H.-P. and Bäck, T. Artificial evolution: How and why? In D. Quagliarella, J. Périaux, C. Poloni and G. Winter (Eds), *Genetic Algorithms an Evolution Strategies in Engineering an Computer Science: Recent Advances and Industrial Applications*, pp. 1-19. Chichester, UK: Wiley, 1998.
- [50] Schwefel, H.-P.,and Rudolph, G., "Contemporary evolution strategies," in *Advances in Artificial Life. 3rd Int. Conf. on Artificial Life* (Lecture Notes in Artificial Intelligence,

vol. 929), F. Mor'an, A. Moreno, J. J. Merelo, and P. Chac'on, Eds. Berlin, Germany: Springer, 1995, pp. 893–907.

- [51] Syswerda, G., "Uniform crossover in genetic algorithms," *Proceedings of the 3rd International Conference on Genetic Algorithms*, pp.2-9, 1989.
- [52] Yen, J. and B. Lee, "A simplex genetic algorithm hybrid", *IEEE*, pp. 175-180, 1997.
- [53] 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 μ 與 λ 對 QGSES 搜尋效率的影響數據

表： μ 與 λ 對 QGSES 在 Colville function 中搜尋效率的影響數據

Population size	Lambda	Best-Value		Time (sec.)		Function-Evaluations	
		Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
1	1	0.000947788	9.39711E-05	1.79968	0.380784016	10017	2269.80499
1	2	0.000771765	0.00022469	1.15786	0.42761102	8230.44	3210.79727
1	3	0.000796931	0.000144846	1.2126	0.346543987	9751.72	2941.603804
1	4	0.000831962	0.000141314	1.17058	0.322659174	9925.64	2884.837248
1	5	0.000736777	0.000171571	1.20968	0.350735579	10670.2	3249.91544
1	6	0.000711813	0.000203543	1.25786	0.343841089	11415.4	3274.72727
1	7	0.000755458	0.000184707	1.28216	0.350357157	11868.24	3407.196631
1	8	0.0006983	0.000212049	1.32584	0.309135421	12393	3038.439557
1	9	0.000754865	0.00022064	1.35626	0.394146653	12863.08	3913.498293
1	10	0.000717553	0.000185482	1.42092	0.337792524	13529.8	3362.846608
2	1	0.00084578	0.000147015	1.24344	0.357340183	9281.92	2799.598979
2	2	0.000710533	0.000203005	1.16304	0.250293946	10317.52	2327.672695
2	3	0.000733442	0.000172841	1.149	0.281206802	10929.68	2843.422582
2	4	0.000741619	0.000197414	1.15532	0.384799202	11402.32	4002.381972
2	5	0.000671463	0.000194492	1.17778	0.371642592	11966.4	3972.906796
2	6	0.000713383	0.000243227	1.21246	0.312378643	12477.68	3368.301197
2	7	0.000713691	0.000215549	1.21748	0.395101291	12560	4292.010252
2	8	0.000741902	0.000201693	1.24256	0.393225135	12917.2	4289.734648
2	9	0.000711468	0.000197496	1.41682	0.294614012	14926.16	3234.83801
2	10	0.000753654	0.00019354	1.50498	0.449231248	15990	4978.767981
4	1	0.000739849	0.000203654	1.10666	0.317392734	9972.16	3025.9743
4	2	0.000723829	0.000260327	1.0957	0.38263369	11102.24	4112.986157

4	3	0.000712663	0.000220371	1.15968	0.354880977	12208.96	3969.510539
4	4	0.000717182	0.000220765	1.21944	0.346190649	13131.68	3896.962865
4	5	0.000713066	0.000221523	1.29066	0.383490979	14064	4416.941411
4	6	0.000703457	0.000227064	1.35196	0.30367645	14906.08	3474.998258
4	7	0.0006998	0.000226161	1.46286	0.40583273	16289.92	4720.687272
4	8	0.000693436	0.000204858	1.47526	0.440470241	16476.32	5147.646014
4	9	0.000698627	0.000200107	1.51186	0.399583917	16913.92	4665.271871
4	10	0.000734094	0.0002101	1.60964	0.460482434	18090.4	5350.252239
6	1	0.000746137	0.000206852	1.18086	0.366182848	11567.28	3779.631205
6	2	0.000737783	0.000230142	1.25692	0.429032221	13427.76	4816.950219
6	3	0.000661046	0.00024824	1.28472	0.387585147	14115.84	4478.743061
6	4	0.000743994	0.000186604	1.3115	0.406241722	14590.32	4733.061681
6	5	0.000691815	0.000189297	1.4251	0.400481611	16087.2	4705.401863
6	6	0.000685176	0.000245625	1.46716	0.404960132	16591.92	4803.544422
6	7	0.000735263	0.000208148	1.53058	0.501637323	17375.52	5946.294604
6	8	0.000731482	0.000187314	1.65492	0.423786616	18810.48	5003.389456
6	9	0.000714079	0.00021695	1.93746	0.563419741	22146	6612.722432
6	10	0.000691009	0.000211486	1.98594	0.548374794	22676.4	6461.462828
8	1	0.000753626	0.000199637	1.41286	0.318247947	14571.2	3427.050329
8	2	0.000677111	0.000215647	1.39916	0.458575806	15339.2	5251.197632
8	3	0.000711307	0.000235398	1.44848	0.421833425	16230.08	4926.130194
8	4	0.000734518	0.000213611	1.46808	0.442493945	16671.04	5227.024443
8	5	0.00065608	0.000204045	1.61248	0.479391631	18428.8	5703.574189
8	6	0.000776667	0.000195401	1.70854	0.508933988	19557.44	6047.91479
8	7	0.000763617	0.000228786	1.7687	0.490632048	20228.48	5796.029979
8	8	0.000706749	0.000247155	2.00308	0.61369077	23053.12	7290.19851
8	9	0.000740411	0.000224468	2.06378	0.469985727	23776.64	5578.87206
8	10	0.000713955	0.000222036	2.3675	0.464840866	27476.8	5548.727613
10	1	0.000734392	0.000231	1.4513	0.475934966	15287.6	5248.075472
10	2	0.000674641	0.000208466	1.49758	0.38130298	16706	4436.457495
10	3	0.00066314	0.000247313	1.52868	0.463188393	17390.8	5472.122985

10	4	0.000701918	0.000217906	1.78426	0.547700529	20515.6	6500.46534
10	5	0.000723797	0.000217822	1.83368	0.479357923	21124	5690.98858
10	6	0.000701217	0.000209934	1.9772	0.513912126	22810	6102.95345
10	7	0.00066267	0.000237547	2.0205	0.609426068	23320	7261.638933
10	8	0.000698926	0.000196561	2.2182	0.649613449	25642	7743.749026
10	9	0.000613368	0.000223711	2.27654	0.610160073	26387.2	7277.116507
10	10	0.000712499	0.000261303	2.53614	0.653453307	29370	7779.827394

表： μ 與 λ 對 QGSES 在 Rosenbrock's function 中搜尋效率的影響數據

Population size	Lambda	Best-Value		Time (sec.)		Function-Evaluations	
		Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
1	1	0.000984487	2.27294E-05	1.83184	0.849032327	6501.64	2810.238117
1	2	0.000977795	3.56274E-05	1.99402	0.868856215	8738.92	3924.593081
1	3	0.000971176	6.13024E-05	2.17722	0.808838227	10256.08	3935.228214
1	4	0.000984394	3.04606E-05	2.3934	1.03878412	11696.84	5207.156517
1	5	0.000985504	2.81916E-05	2.58732	1.037570759	13049.6	5362.442335
1	6	0.000981516	3.79001E-05	2.72534	1.283733977	13846.6	6679.280861
1	7	0.000982889	2.86227E-05	3.17596	1.232173589	16375.12	6460.656988
1	8	0.00097257	5.50339E-05	3.40404	1.304066543	17749.48	6929.227993
1	9	0.000991212	1.35605E-05	3.50586	1.54116807	18395.2	8228.238141
1	10	0.000988664	2.28873E-05	3.53818	1.707160606	18665.8	9152.61226
2	1	0.000975804	4.57363E-05	1.78006	0.712544003	7960.24	3305.699139
2	2	0.000971885	3.54144E-05	2.2007	0.909508978	11068.24	4714.319388
2	3	0.000980039	2.97827E-05	2.46658	1.026683964	12924.32	5510.414575
2	4	0.000972579	3.86722E-05	2.74846	1.248000985	14979.28	6958.254036
2	5	0.000977936	4.05919E-05	2.88814	1.47730665	15766	8240.766756
2	6	0.000974239	5.64147E-05	3.82158	1.58497826	21132.56	8897.186008
2	7	0.000984234	2.79768E-05	4.09274	1.744737683	22720.64	9828.788556
2	8	0.000985172	3.10737E-05	4.75906	1.737485991	26538.32	9821.064636
2	9	0.000990046	1.56799E-05	5.25272	2.099446128	29412.56	11902.47587
2	10	0.000978494	4.19199E-05	5.40654	2.434837657	30309.2	13779.42255
4	1	0.000967509	4.57365E-05	2.1641	0.80506948	10971.36	4204.616577
4	2	0.000977295	3.71931E-05	2.91726	1.183728265	15894.88	6581.263053
4	3	0.000972746	3.67845E-05	3.1475	1.326774349	17600.32	7586.84941
4	4	0.000970664	3.84655E-05	3.86282	1.39281472	21880.48	8025.358041
4	5	0.000967177	6.27633E-05	4.01066	1.990402037	22761.6	11473.72625
4	6	0.000977001	3.30286E-05	5.14498	1.949862889	29398.24	11277.82547

4	7	0.000975691	3.09412E-05	5.51966	2.446316803	31714.56	14227.54426
4	8	0.000963746	6.45815E-05	6.07626	2.61748693	34960.8	15201.41261
4	9	0.00097695	4.66845E-05	7.0875	3.219955883	40918.72	18769.48016
4	10	0.000974288	5.17175E-05	7.25934	3.788780309	42284	22245.42732
6	1	0.000956462	5.83622E-05	2.70468	1.053989398	14563.44	5805.168127
6	2	0.000956034	7.32386E-05	2.90408	1.270907467	16461.84	7360.499449
6	3	0.000961306	4.8512E-05	4.12162	1.659672489	23784	9740.102342
6	4	0.000967108	5.26359E-05	4.28836	1.727610001	24864.24	10146.79045
6	5	0.00095958	5.63135E-05	4.67716	2.296902034	27268.8	13556.41215
6	6	0.000936457	0.000109205	5.41382	2.821462131	31611.12	16682.01413
6	7	0.000971428	3.59597E-05	6.76908	2.672763355	39670.8	15806.13653
6	8	0.000958767	8.25137E-05	7.13876	3.248641484	41556.72	19088.55844
6	9	0.000963898	5.04684E-05	7.78136	3.614157405	45456.72	21287.27318
6	10	0.000964927	8.17862E-05	8.78156	3.770971656	51368.4	22215.03284
8	1	0.000964476	4.33304E-05	2.80716	0.915832093	15306.56	5110.619325
8	2	0.000968155	3.44055E-05	3.39736	1.367372406	19311.68	7935.527465
8	3	0.000941978	9.72728E-05	4.26314	2.087422334	24541.76	12189.10963
8	4	0.000964344	3.92011E-05	5.15658	1.991906595	29915.2	11678.71975
8	5	0.000968023	3.77801E-05	5.74694	2.176983509	33376	12777.80985
8	6	0.000968944	3.57047E-05	6.47716	2.348712269	37808.96	13848.23128
8	7	0.000955325	6.44441E-05	7.63254	3.215382633	44234.56	18766.17403
8	8	0.000979723	2.35585E-05	8.66094	3.206130043	50058.56	18657.60328
8	9	0.000969209	5.16862E-05	9.28408	3.834763984	53869.76	22405.27886
8	10	0.000959135	7.82114E-05	9.3266	4.21488066	54238.4	24685.51132
10	1	0.000952569	5.55581E-05	3.03844	1.142614549	16898.8	6466.996477
10	2	0.000957764	5.93646E-05	4.16406	1.644012257	23941.2	9590.972082
10	3	0.0009713	3.22636E-05	4.93806	1.846660609	28649.2	10846.65731
10	4	0.000954341	6.13859E-05	5.28344	2.357707942	30784.4	13899.18632
10	5	0.000963698	6.39722E-05	6.30188	2.854640609	36644	16741.82847
10	6	0.000953811	9.17752E-05	7.02158	3.102864997	40790.8	18187.93902
10	7	0.000943632	8.16874E-05	7.47258	3.747931092	42172.4	21432.41209

10	8	0.000966801	5.38183E-05	8.7366	3.888781164	49638.8	22265.83002
10	9	0.000963953	6.94399E-05	10.21958	3.954488584	59413.6	23094.7354
10	10	0.000968799	4.43072E-05	10.67374	4.733230673	61218	27335.79428

表： μ 與 λ 對 QGSES 在 Shubert function 中搜尋效率的影響數據

Population size	Lambda	Best-Value		Time (sec.)		Function-Evaluations	
		Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
1	1	-186.7059092	0.037718162	1.48444	1.544605255	9085.84	9875.117228
1	2	-186.7219603	0.022510104	0.56592	1.423488742	4248.12	12038.693
1	3	-186.7283138	0.001496897	0.10348	0.007677797	381.64	30.62196091
1	4	-186.7284337	0.001592326	0.11192	0.007283865	494.6	41.49846321
1	5	-186.7286497	0.001504867	0.12188	0.008925176	612.4	58.02919744
1	6	-186.7286609	0.001684781	0.13124	0.008299398	739	42.14842578
1	7	-186.7289112	0.001309551	0.13918	0.009635585	855.84	85.43582387
1	8	-186.7287566	0.001530204	0.1494	0.009549014	968.36	70.31757698
1	9	-186.7286275	0.001366697	0.159	0.009839529	1088.2	81.64032617
1	10	-186.7285581	0.001403641	0.16528	0.009521276	1179.4	97.96625825
2	1	-186.725576	0.012315752	0.45434	1.233058262	3452.16	10887.42364
2	2	-186.7287407	0.001422778	0.10996	0.008400583	502.96	46.56045445
2	3	-186.7282089	0.001136662	0.12536	0.007403143	720.8	60.95967218
2	4	-186.7288925	0.001357504	0.1435	0.008074273	966.48	70.82321769
2	5	-186.7287109	0.00152865	0.15778	0.008567356	1190.4	81.42506258
2	6	-186.7289217	0.001356979	0.17492	0.011638851	1414.64	104.8952364
2	7	-186.7287569	0.001287849	0.18936	0.015097993	1628.8	178.549713
2	8	-186.7286117	0.001447669	0.20432	0.013297905	1830.48	144.8515412
2	9	-186.7288345	0.001432852	0.22744	0.011473412	2128.16	119.1567926
2	10	-186.7285652	0.001554751	0.23856	0.01436998	2288.4	178.219768
4	1	-186.7283167	0.001506689	0.1116	0.008359279	534.24	41.64895079
4	2	-186.7288249	0.001371574	0.13876	0.009723861	945.44	91.44914054
4	3	-186.7283871	0.001254832	0.1669	0.011157957	1366.72	121.3433382
4	4	-186.7284276	0.001339178	0.2	0.0114107	1836.96	121.4727583
4	5	-186.7287007	0.001330612	0.22594	0.016173056	2221.6	213.9860762
4	6	-186.7281603	0.001306906	0.25434	0.022531257	2622.88	315.0049329

4	7	-186.728258	0.001629538	0.29056	0.018722637	3146.72	235.8684379
4	8	-186.728498	0.001455492	0.31878	0.022308862	3535.52	281.4129249
4	9	-186.7286444	0.001434963	0.35596	0.026740598	4061.92	358.899215
4	10	-186.7288295	0.001437898	0.38246	0.049002795	4450.4	657.5333002
6	1	-186.7292252	0.00133107	0.12468	0.008629423	732.24	69.81332543
6	2	-186.7282979	0.001454524	0.1668	0.012444161	1379.28	148.8402842
6	3	-186.7286734	0.001435273	0.2118	0.017038971	2032.8	193.1483516
6	4	-186.7286487	0.001520388	0.25412	0.018606165	2655.6	223.2523967
6	5	-186.7282864	0.001461193	0.30478	0.021290306	3379.2	286.9291315
6	6	-186.7281766	0.001410302	0.34124	0.024375933	3934.32	329.1942817
6	7	-186.7285722	0.001573499	0.38352	0.041674267	4533.6	600.5397572
6	8	-186.728706	0.001395723	0.43994	0.024902594	5347.44	333.5476688
6	9	-186.7286481	0.001484376	0.48162	0.052364067	5926.56	731.4151829
6	10	-186.727841	0.001224115	0.5375	0.028549456	6733.2	387.4513268
8	1	-186.7285322	0.001388448	0.14126	0.01001185	976.96	88.23922956
8	2	-186.7284906	0.001451849	0.19572	0.014202673	1798.72	169.9391296
8	3	-186.7285288	0.001447992	0.24686	0.024111146	2571.2	320.6035125
8	4	-186.7287464	0.001394943	0.30376	0.025958806	3398.72	367.4222972
8	5	-186.7286069	0.001252657	0.36868	0.025226678	4340.8	344.1347624
8	6	-186.7281605	0.001384465	0.4321	0.028855198	5264.96	385.2518371
8	7	-186.7287056	0.001358299	0.48438	0.039285397	6040.32	562.5132176
8	8	-186.7282773	0.001455783	0.5357	0.085463502	6740.8	1232.625338
8	9	-186.7282741	0.001416522	0.6025	0.079995472	7720.64	1148.011053
8	10	-186.7282915	0.001352608	0.6522	0.100064061	8427.2	1445.57289
10	1	-186.7282548	0.001438871	0.15472	0.011984752	1174	130.4309967
10	2	-186.7286353	0.001368629	0.22066	0.015634355	2181.2	197.7902414
10	3	-186.7289572	0.001320682	0.29304	0.023272968	3247.6	288.8863911
10	4	-186.7282043	0.001506652	0.3634	0.037821816	4291.6	537.3227662
10	5	-186.7290144	0.001116844	0.44004	0.028002157	5402	400.9376765
10	6	-186.7281052	0.001361501	0.50654	0.040940579	6384.4	584.1867188
10	7	-186.7282446	0.001467263	0.59102	0.04129856	7575.6	594.6158424

10	8	-186.7283561	0.001464467	0.65208	0.065174754	8483.6	929.5637774
10	9	-186.7281135	0.001284856	0.73194	0.076259602	9604	1103.16983
10	10	-186.7283446	0.001341553	0.8031	0.05623248	10614	809.6257639

表： μ 與 λ 對 QGSES 在 SinCos function 中搜尋效率的影響數據

Population size	Lambda	Best-Value		Time (sec.)		Function-Evaluations	
		Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
4	1	0.00200059	0.002774057	2.82994	3.103286891	28386.72	31929.88332
4	2	0.000626185	0.000304946	0.83412	1.904830772	8731.36	22222.47913
4	3	0.000632366	0.000294788	0.46564	0.300539363	4685.92	3635.91502
4	4	0.000589368	0.000240899	0.34852	0.135867468	3286.56	1654.555174
4	5	0.000645906	0.0002887	0.38808	0.173155214	3857.6	2155.590511
4	6	0.000568255	0.000265485	0.4218	0.232283694	4321.12	2945.498625
4	7	0.00059452	0.000253621	0.42254	0.165389239	4332.8	2084.679352
4	8	0.000612947	0.000237812	0.43184	0.08947588	4469.92	1132.555352
4	9	0.000580325	0.000248577	0.45416	0.113571763	4771.84	1435.753821
4	10	0.000545721	0.000281159	0.50722	0.158833155	5460	2030.10807
6	1	0.000695982	0.000441824	1.2131	2.450978424	12790.56	27660.33429
6	2	0.000629941	0.000255266	0.40728	0.285813771	4034.64	3470.947302
6	3	0.000581849	0.000293454	0.38126	0.198885754	3819.84	2530.112294
6	4	0.000571975	0.000248216	0.42588	0.187537603	4435.44	2427.371234
6	5	0.000605288	0.000239902	0.44348	0.16907277	4695.6	2192.220904
6	6	0.000567028	0.000260989	0.4937	0.164253444	5354.16	2131.652682
6	7	0.000541052	0.000253008	0.51062	0.152755091	5578.56	1980.880531
6	8	0.00048282	0.000263793	0.53718	0.073317506	5938.8	957.4496736
6	9	0.000563033	0.000283462	0.57874	0.056043646	6505.44	707.5947232
6	10	0.000605667	0.000267873	0.64274	0.10412626	7338	1372.124823
8	1	0.000609202	0.000265385	0.55594	0.351458153	5626.88	4155.853117
8	2	0.000584835	0.000287156	0.36998	0.153733601	3647.04	1940.430511
8	3	0.000565704	0.000295408	0.47536	0.199617876	5098.88	2578.056099
8	4	0.000578476	0.000252624	0.49216	0.179036236	5346.88	2318.506351
8	5	0.000619225	0.000242183	0.51934	0.173544514	5742.4	2275.43058
8	6	0.00057537	0.0002669	0.53594	0.094433307	5963.84	1239.146239

8	7	0.000631549	0.000258831	0.58402	0.061005015	6604.8	768.4165537
8	8	0.000580844	0.000272309	0.64812	0.052127254	7465.28	673.7535139
8	9	0.000605302	0.000236781	0.72496	0.066024411	8475.2	876.6422024
8	10	0.000626602	0.000244225	0.766	0.070259287	9022.4	912.3465149
10	1	0.000631059	0.000261928	0.44436	0.268881186	4291.6	3115.532752
10	2	0.000552279	0.00027016	0.42312	0.210959774	4392.4	2571.176623
10	3	0.00060826	0.000217183	0.43402	0.130232522	4595.2	1622.881338
10	4	0.000639013	0.000230306	0.5038	0.154096842	5536.4	1952.168611
10	5	0.000649971	0.000244412	0.5609	0.127934334	6238	1634.080906
10	6	0.000600616	0.000259533	0.6378	0.107892236	7246	1341.147848
20	1	0.000595638	0.000267052	0.55098	0.33851214	5920	4105.739139
20	2	0.000538962	0.00027284	0.53988	0.087263766	5965.6	1109.271061
20	3	0.000538721	0.000288579	0.66742	0.113468883	7623.2	1470.645339
20	4	0.000548917	0.000270675	0.81752	0.065912818	9520.8	827.220179
20	5	0.000610712	0.000259863	0.97382	0.053089332	11444	637.136451
20	6	0.000623826	0.000232204	1.13248	0.067886231	13508	846.7946767
30	1	0.000536103	0.000298086	0.53974	0.148311697	5865.6	1845.021055
30	2	0.000561745	0.000245474	0.70462	0.076311898	8103.6	970.0615592
30	3	0.000612427	0.000229087	0.9126	0.097011466	10725.6	1200.661654
30	4	0.000577389	0.000241216	1.13504	0.084187827	13537.2	1062.348265
30	5	0.000549768	0.000287452	1.37622	0.079650586	16554	1010.265675
30	6	0.000586755	0.000262365	1.5747	0.191893067	19002	2402.600632

表： μ 與 λ 對 $x\sin(1/x)$ function with 2 variables 中搜尋效率的影響數據

Population size	Lambda	Best-Value		Time (sec.)		Function-Evaluations	
		Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
1	1	-0.433926229	0.000306441	0.096	0.02599529	194.6	149.4741804
1	2	-0.433894811	0.000306454	0.08628	0.018700693	186.12	148.3889869
1	3	-0.433913363	0.000323263	0.0791	0.014241718	134.68	116.3508838
1	4	-0.433917467	0.000311091	0.08368	0.016717875	175.56	161.6057384
1	5	-0.433984624	0.000314829	0.08312	0.022581951	183.2	221.4911432
1	6	-0.433919922	0.00032699	0.08242	0.014876485	188.68	131.9167084
1	7	-0.434018569	0.000268946	0.07968	0.020511261	189.72	226.1900086
1	8	-0.433971435	0.000321334	0.08056	0.01490673	194.28	145.0605841
1	9	-0.433904842	0.000352027	0.08726	0.016942652	225.64	198.4249984
1	10	-0.433933026	0.000333217	0.083	0.015576015	232.6	147.4734151
2	1	-0.433993868	0.000308096	0.084	0.023264232	185.12	198.7492154
2	2	-0.433939776	0.000332964	0.08096	0.016319464	179.76	173.4455817
2	3	-0.433946673	0.000283724	0.079	0.015484027	186.56	159.4910477
2	4	-0.43400815	0.00029328	0.07972	0.012719967	193.68	140.3246149
2	5	-0.433897889	0.000339107	0.07932	0.012998336	196	141.6088317
2	6	-0.433948396	0.000315387	0.07848	0.013572541	193.04	141.0272748
2	7	-0.434082869	0.000311598	0.08034	0.012201723	202.48	131.6188436
2	8	-0.433932128	0.00028884	0.08132	0.014819237	215.76	149.8759433
2	9	-0.43396003	0.000310901	0.08216	0.016960169	241.76	197.2619268
2	10	-0.43396962	0.000304354	0.08378	0.01400188	245.2	181.2140464
4	1	-0.433954218	0.000320416	0.07842	0.015549906	164.64	159.0653517
4	2	-0.433894715	0.000319861	0.08212	0.016263945	236	193.5245594
4	3	-0.433898799	0.000328578	0.08278	0.016138798	245.92	202.3054712
4	4	-0.43393274	0.000317071	0.08158	0.017568629	246.56	182.5242684
4	5	-0.433971377	0.000306995	0.08182	0.012792999	256	165.8620031
4	6	-0.43394671	0.000321665	0.08242	0.011369652	256.48	140.8003711

4	7	-0.433993647	0.000324803	0.08396	0.010405964	268	133.7582345
4	8	-0.43394674	0.000275053	0.08578	0.012696279	270.56	161.020744
4	9	-0.433991893	0.000320106	0.08412	0.01043667	274.72	127.2771426
4	10	-0.434011766	0.000302396	0.0847	0.013456278	296.8	146.9004939
6	1	-0.433896283	0.000343881	0.0803	0.015186259	201.6	176.8167505
6	2	-0.433967998	0.000306443	0.08002	0.014703172	209.04	190.4144738
6	3	-0.433925839	0.000303443	0.08256	0.013875775	276	185.3213425
6	4	-0.434015258	0.000290665	0.08352	0.012460943	284.4	167.9650109
6	5	-0.433972392	0.000332786	0.0837	0.013683373	291.6	158.7728451
6	6	-0.433927032	0.000318958	0.08412	0.012719325	283.92	137.2362539
6	7	-0.433934255	0.000307236	0.08402	0.012539716	318.48	156.4976677
6	8	-0.433988862	0.000295603	0.08538	0.01236666	307.44	158.7661816
6	9	-0.434020038	0.000309848	0.08844	0.013209706	358.08	184.8984101
6	10	-0.433972747	0.000283154	0.09242	0.017569791	418.8	240.1958385
8	1	-0.43396597	0.000299865	0.07966	0.015505904	221.12	171.8224808
8	2	-0.43393208	0.000318045	0.0794	0.012631351	220.48	141.7526215
8	3	-0.43401562	0.000314046	0.08156	0.01391543	279.68	172.0063407
8	4	-0.43395647	0.000248528	0.08188	0.01459052	283.2	181.1347452
8	5	-0.433983448	0.000308225	0.08566	0.014916926	313.6	186.177006
8	6	-0.433943065	0.000317564	0.08836	0.015350012	361.28	213.731764
8	7	-0.433956135	0.000314692	0.08904	0.011540452	379.84	145.6615255
8	8	-0.433899438	0.000302441	0.09012	0.012848632	392	159.4111614
8	9	-0.433983736	0.000291007	0.09252	0.01305286	414.08	168.8612327
8	10	-0.433997896	0.000308902	0.09302	0.013786255	446.4	167.3729796
10	1	-0.433948401	0.000301838	0.07686	0.010453024	177.2	118.6839397
10	2	-0.433894112	0.000306118	0.08118	0.015131654	240.4	171.9154989
10	3	-0.433995182	0.000332951	0.08212	0.012902523	280	182.7343338
10	4	-0.433963292	0.000260519	0.08526	0.011876593	290	128.5396786
10	5	-0.433933209	0.000324578	0.08946	0.012726655	332	123.3710183
10	6	-0.434006662	0.000302367	0.0916	0.014512486	422.8	198.6797238
10	7	-0.434036314	0.000304789	0.09306	0.013741654	430	176.6352173

10	8	-0.434047357	0.000305275	0.09436	0.013792574	461.2	190.387789
10	9	-0.434083034	0.000291726	0.09744	0.015923491	496	203.2892782
10	10	-0.434095861	0.000318189	0.09804	0.015802299	530	217.5935173

表： μ 與 λ 對 $x\sin(1/x)$ function with 5 variables 中搜尋效率的影響數據

Population size	Lambda	Best-Value		Time (sec.)		Function-Evaluations	
		Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
1	1	-1.081088881	0.001595641	0.68334	0.54896967	3074.24	2890.066136
1	2	-1.081258014	0.001446598	0.41224	0.339895095	2728.6	2269.381252
1	3	-1.081294575	0.001426406	0.461	0.314522833	2909.56	2299.667662
1	4	-1.081164714	0.001541508	0.44998	0.380711873	2970.12	2940.868348
1	5	-1.081011986	0.001656491	0.47754	0.396750022	3313.4	3190.919233
1	6	-1.080878706	0.001414522	0.51528	0.413394314	3715.72	3386.493358
1	7	-1.080622318	0.001301868	0.55442	0.466321256	4055.68	3847.216022
1	8	-1.081149063	0.001565062	0.53814	0.486815098	3997.8	4126.736715
1	9	-1.080810946	0.001226899	0.5295	0.490242974	3952.72	4157.419058
1	10	-1.081311306	0.001784399	0.54658	0.504555415	4184.2	4336.770848
2	1	-1.081427052	0.001616155	0.5622	0.493678115	3457.12	3429.20216
2	2	-1.081264882	0.001760792	0.50842	0.378439119	3572.88	3053.319803
2	3	-1.081010122	0.001550047	0.5542	0.389116844	4252.4	3390.389972
2	4	-1.080955639	0.001574294	0.47096	0.363299953	3622.16	3222.898983
2	5	-1.0810355	0.001802933	0.5031	0.375055765	4011.6	3410.098425
2	6	-1.081055268	0.001572673	0.44692	0.350233149	3524.24	3235.807028
2	7	-1.08089752	0.001302155	0.42494	0.274275179	3363.68	2545.272701
2	8	-1.080971731	0.001409409	0.47348	0.446349878	3825.36	4154.557143
2	9	-1.081092024	0.001727953	0.47008	0.422545099	3836	3976.610987
2	10	-1.080964277	0.001390329	0.44402	0.347049693	3582.8	3260.852424
4	1	-1.081029477	0.001428787	0.46882	0.346277091	3344.48	2860.302326
4	2	-1.081107847	0.001557798	0.44438	0.272659307	3471.84	2480.984263
4	3	-1.081480416	0.001554876	0.44584	0.399049859	3619.36	3761.421821
4	4	-1.081285803	0.001645882	0.4598	0.370240628	3823.52	3555.265335
4	5	-1.081448666	0.001523759	0.45758	0.328410463	3831.2	3209.103785
4	6	-1.080778455	0.001501038	0.36278	0.226469378	2938.72	2229.889931

4	7	-1.08106849	0.001478342	0.42138	0.32846364	3504	3205.240709
4	8	-1.080913241	0.00125443	0.54994	0.344004058	4737.44	3361.339068
4	9	-1.081020187	0.001482206	0.50334	0.390804017	4299.52	3800.306552
4	10	-1.081048727	0.001722296	0.47882	0.352071403	4074.4	3448.764913
6	1	-1.081612637	0.001655213	0.41744	0.322055652	3149.76	2871.01915
6	2	-1.080880163	0.001428322	0.4175	0.311761241	3376.56	2968.205038
6	3	-1.081208671	0.001674854	0.4235	0.291806495	3529.68	2858.139237
6	4	-1.081046698	0.001367572	0.44506	0.348968767	3746.16	3420.385748
6	5	-1.080909099	0.001403022	0.41186	0.326898662	3471.6	3244.725805
6	6	-1.080847704	0.001601308	0.4806	0.328433698	4144.56	3262.930773
6	7	-1.080969001	0.001668411	0.46852	0.368358257	4041.36	3657.066103
6	8	-1.080934893	0.001534488	0.4216	0.31294128	3588.72	3134.273311
6	9	-1.081260254	0.001488616	0.45224	0.32502016	3874.56	3235.198156
6	10	-1.081010908	0.001671581	0.49514	0.458005437	4333.2	4580.410391
8	1	-1.081042668	0.001609844	0.50412	0.46694612	4060.16	4313.416977
8	2	-1.081426765	0.001782961	0.49488	0.323209835	4178.24	3121.307958
8	3	-1.081729809	0.001666603	0.51354	0.46389171	4467.2	4582.103251
8	4	-1.081074834	0.001700584	0.43092	0.285689988	3661.12	2850.074805
8	5	-1.081306408	0.001593369	0.41098	0.33309118	3481.6	3309.951452
8	6	-1.08149924	0.001726003	0.41904	0.284261805	3554.24	2835.337611
8	7	-1.080955216	0.001574049	0.4265	0.301738788	3634.56	3010.967898
8	8	-1.080678218	0.001608872	0.49908	0.325830232	4352.32	3260.749466
8	9	-1.081148832	0.001402493	0.49098	0.356445519	4299.2	3593.352229
8	10	-1.081371398	0.001445997	0.489	0.412953177	4276.8	4145.965652
10	1	-1.081437731	0.00181135	0.49374	0.337650635	4050	3169.139457
10	2	-1.081072858	0.00149607	0.45154	0.282269607	3834	2784.39969
10	3	-1.080750977	0.001324436	0.48502	0.350284869	4232.8	3489.337719
10	4	-1.081351682	0.001449314	0.40694	0.278919593	3458	2803.298349
10	5	-1.081140498	0.001559299	0.37154	0.247039963	3112	2484.276267
10	6	-1.080997118	0.001575142	0.38156	0.298003915	3211.6	2997.151954
10	7	-1.081103513	0.001378204	0.48724	0.479272304	4266	4834.12867

10	8	-1.081307065	0.001620378	0.47244	0.377135182	4128.4	3793.708734
10	9	-1.081473214	0.001682256	0.44756	0.386282975	3869.2	3903.350649
10	10	-1.081328248	0.00164009	0.4579	0.327362263	3966	3276.456523