**群體智慧期中報告**

**多種超啟發式演算法實作單一目標函數最佳化**

許哲維

國立中山大學

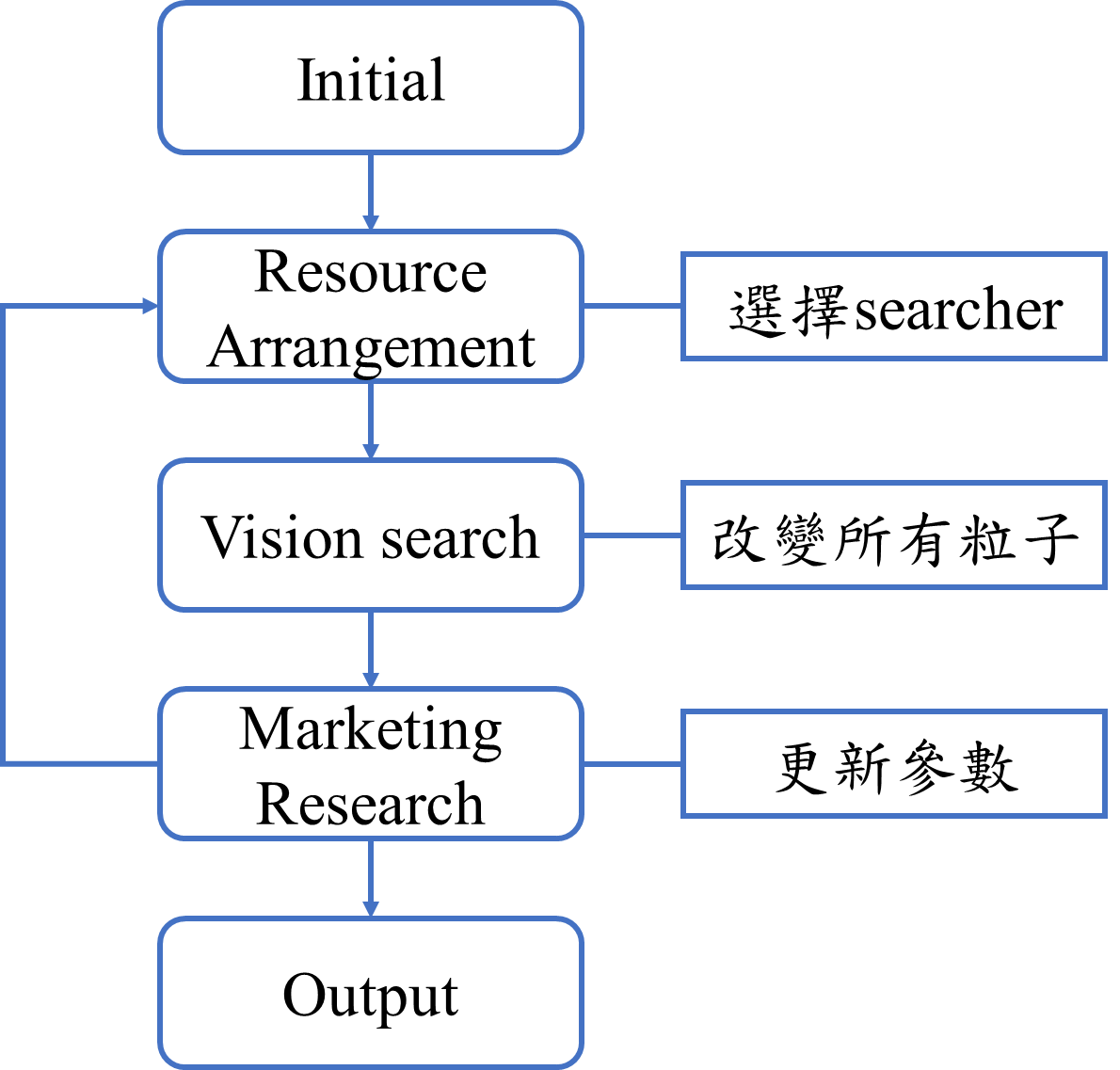
M093040082

# 摘要

粒子群演算法 (Particle Swarm Optimization, PSO) 是一個簡單有效的的單一目標函數最佳化的演算法，而多群最佳化演算法 (Multi-Swarm Optimization, MSO) 是在 PSO 原先的機制上增加多個子群改良而來的。本次期中報告是融合 MSO 與搜尋經濟演算法 (Search Economics ,SE) 的機制，設計搜尋經濟多群演算法(Search Economics with Multi-Swarm Optimization, SEMSO)。透過不同的子群來區分 SE 中不同的區域，並更改更新機制除了參考自己與所在子群的最佳解還會參考每個搜尋者所在的位置。另外為了增加每個區域的區別性，每個區域會有獨立的參數調整以達到有些子群善於全域搜尋有些子群善於區域搜尋。

## 搜尋經濟多群演算法

* 1. **實驗流程圖**



* 1. **演算法**

在多群演算法中如何交換不同子群的資訊一個重要的議題，因此本期中報告就利用SE的特性來完成不同子群的資訊交換。

搜尋者的選擇分成兩的步驟。首先計算每個子群的潛力值，並依照潛力值高低影響區域被選擇的機率。再從選擇的區域中隨機挑出適應值較好的點當作新的搜尋者。

粒子在移動的過程中除了會參考過去該例子的最佳解和所在子群的全域最佳解外，還會參考所有的搜尋者。透過參考屬於不同子群的搜尋者來達到群之間的資訊交換。

for *i*th swarm *j*th solutions

for *i*th swarm *j*th sol’s fitness

for all *i*th searchers

= Initialization()

= Fitness () // update same time

Do

*s* = Resource\_Arrangement( )

= mean(*s*)

=

= +

= Mutation()

= Fitness ()

Marketing\_Research ( )

While

Output

* 1. **公式**

粒子的移動的距離會受到三個方向吸引：1. 該子群所在的全域最佳解，2. 該粒子的過去最佳解，3. 目前所有搜尋者的平均位置。分別代表三個力的影響力度。而為了增加多樣性與增加不同區域的差別，每個區域會有獨立且不盡相同的。而為了增加更多的多樣性，每個影響係數會在乘上 0~1 的亂數 (每個維度獨立，更多的多樣性)。如同一般的 PSO ，我的移動也會加入慣性的機制會把上次移動距離乘上衰退值一並加入下次移動距離。

=

突變則是以當下速度為基準，以正負速度為區間做隨機成為當下的突變。並以 控制突變量。而突變的機率為每個維度都有50%的機會產生突變。

## 實驗結果

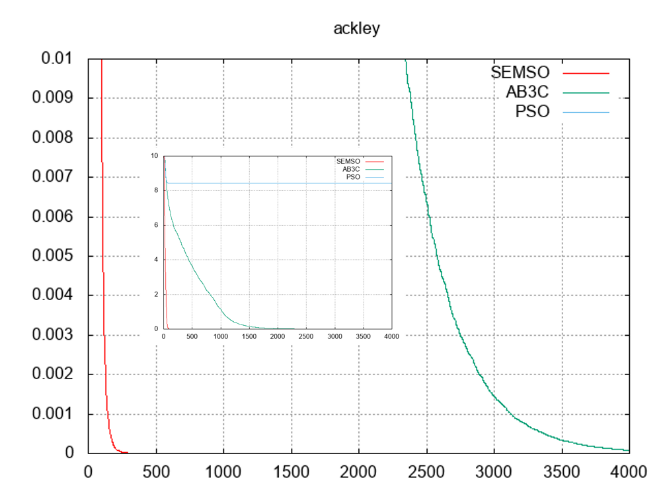
在本次研究中，演算法都是以30個回合數來作平均 (PSO為3 run)，且迭代數量皆為4000，以得到公平之實驗數據。總共有8個搜尋者、4個區域 (子群) 、每個區域有25個粒子、初始速度範圍的0.0001%、速度衰退率為0.7，其他參數如下：

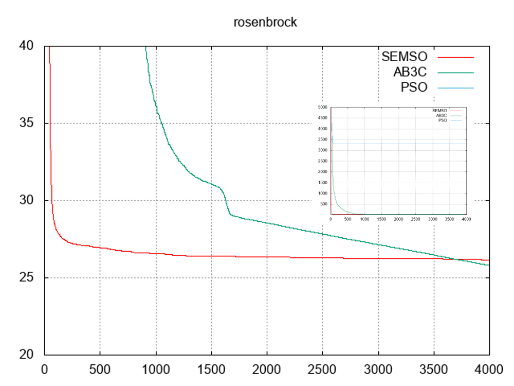
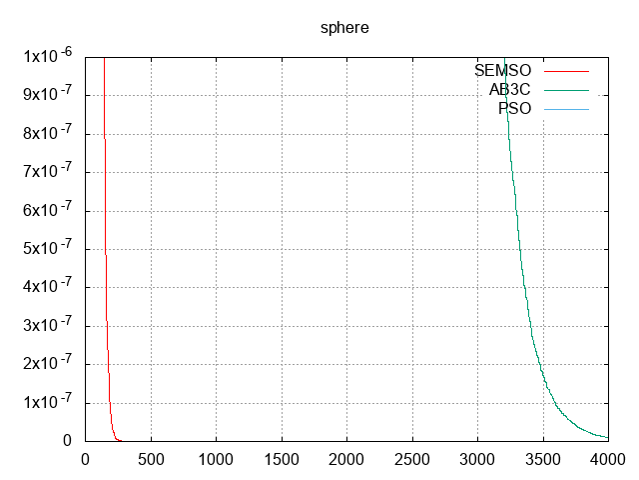
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  | 1.5 | 1 | 1 | 1 |
|  | 1 | 1.5 | 1 | 1 |
|  | 1 | 1 | 1.5 | 1 |
|  | 1 | 1 | 1 | 1 |

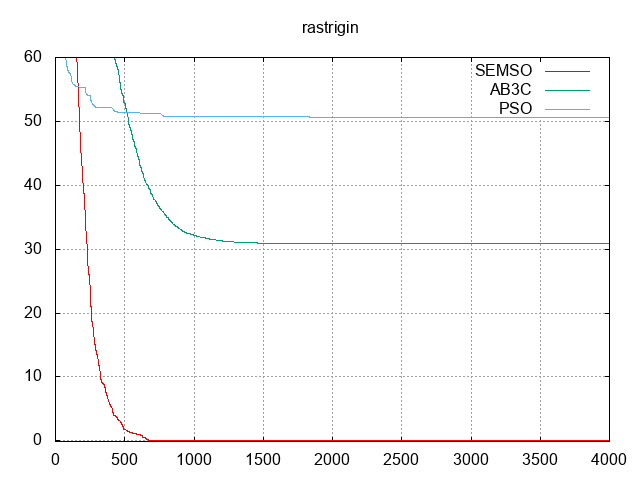
|  |  |  |
| --- | --- | --- |
|  | **Min range** | **Max range** |
| **Ackley** | -32.768 | 32.768 |
| **Sphere** | -30 | 30 |
| **Rastrigin** | -5.12 | 5.12 |
| **Rosenbrock** | -10 | 10 |
| **Michalewicz** | 0 | PI (3.1415926….) |

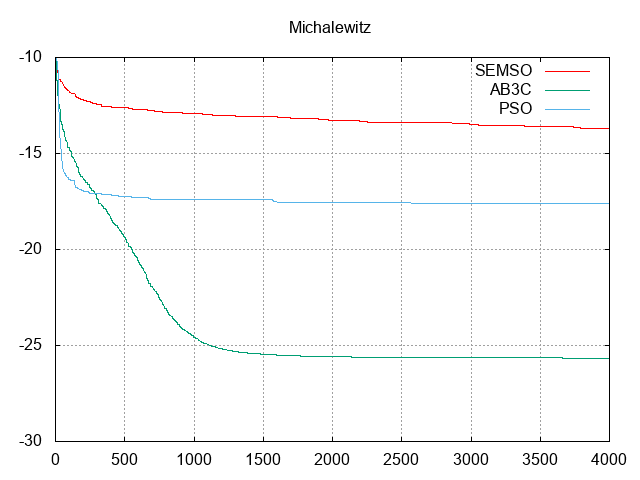
本次實驗一共做了5種testing function：1. Ackley，2. Sphere，3. Rastrigin，4. Rosenbrock和5. Michalewicz。每個testing function 皆以30維做為測試。另外我有與實驗室同學羅宇宸((Adative Big Bang–Big Crunch, AB3C))合作互相分享彼此的實驗結果，我們的結果會下列表格與收斂圖一併呈現。

|  |  |  |  |
| --- | --- | --- | --- |
|  | **SEMSO** | **AB3C** | **PSO** |
| **Ackley** | **3.99680e-15** | 8.02174e-5 | 8.4407 |
| **Sphere** | **8.44177e-82** | 9.61854e-9 | 140.5109 |
| **Rastrigin** | **0.00000** | 30.8768 | 50.6807 |
| **Rosenbrock** | 26.1569 | **25.8029** | 3298.88 |
| **Michalewicz** | -13.7046 | **-25.6593** | -17.6109 |









## 結論

SEMSO保留了PSO搜尋特性，並明顯的提升了很大的全域與區域的搜尋能力。從Rastrigin和Sphere就可以看出區域搜尋的能力，一個可以找到0另一個可以找到e-82。而與AB3C相比則是各有春秋，透過觀察可以發現SEMSO有較強的全域搜尋能力，故再有些testing function會收斂再比較好的地方，而AB3C有較好的區域搜尋能力，所以在特定testing function在後半迭代還是可以繼續收斂。