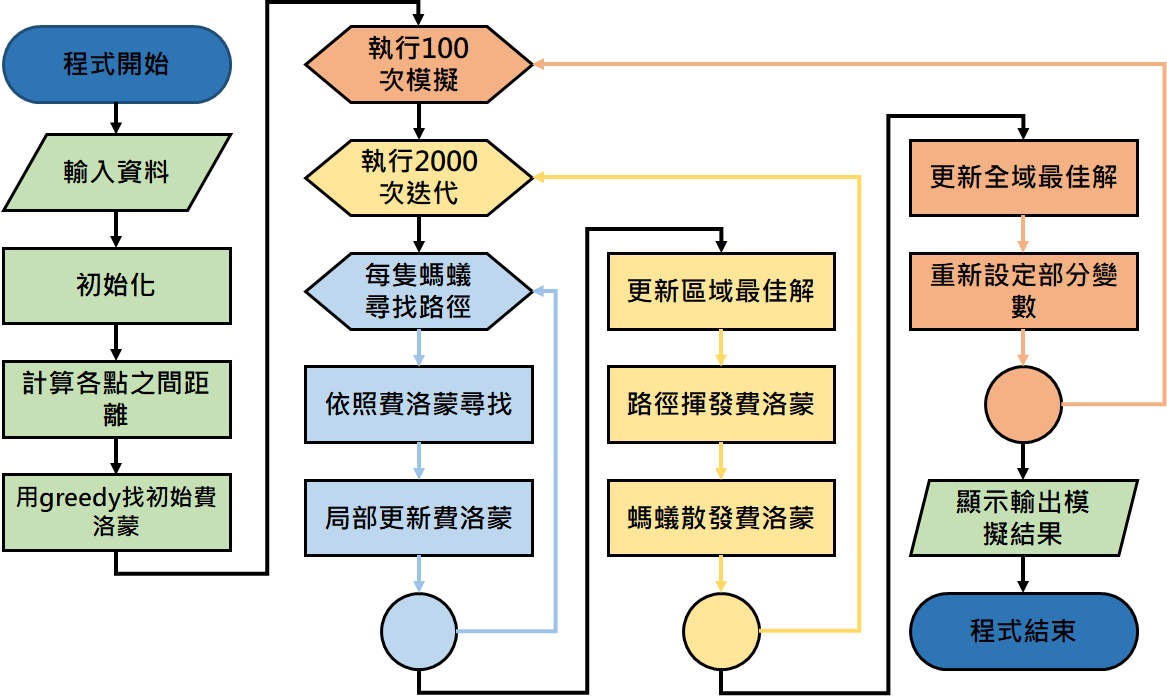
**基因演算法 期中報告**

1. 介紹

我們實作M. Dorigo和L. M. Gambardella所寫的” Optimization, learning and natural algorithms”[1]和"Ant colony system: a cooperative learning approach to the traveling salesman problem"[2] 這兩篇論文。作者透過觀察螞蟻的行為發展了螞蟻演算(Ant System, AS)法與蟻群演算法 (Ant Colony System, ACS) ，這是解決旅行銷售員(TSP)問題的其中一種作法。透過模擬螞蟻利用費洛蒙信息找到從食物源到巢穴的最短路徑，根據兩點之間的路徑長短和費洛蒙的濃度來影響權重，而權重大小會影響抉擇路徑的結果。

本報告在第貳章說明實作該論文的過程與方式，第參章中使用了AS、ACS without local updating、ACS with=0、ACS with=0等四種方式來比較模擬結果，第肆章是結論。

1. 實作方法
   1. 整體流程圖



* 1. 相關變數(預設)
     1. 資料集：eil51
     2. 模擬次數(round)：100
     3. 迭代次數(iter)：2000
     4. 螞蟻數量：15
     5. 開發率(q0)：0.9
     6. 長度因子()：2.0
     7. 區域衰退率()：0.1
     8. 全域衰退率()：0.1
     9. 起點r：該螞蟻目前的位置
     10. 目標s：該螞蟻的目標位置
     11. 費洛蒙(r,s)：從第r點到第s點的費洛蒙
     12. 初始費洛蒙()：用greedy求解的倒數
     13. 距離l(r,s)：從第r點到第s點的距離
  2. 初始化
     1. 計算各點之間的距離
     2. 用greedy尋找一個不錯的解，依此決定初始費洛蒙(0)
     3. 為螞蟻設定一個起點
  3. 尋找路徑

在論文中，提到了兩種尋找路徑的方法：探索(exploration)與開發(exploitation)，在原本的AS中，全部的螞蟻皆用來探索，而ACS中，則是將大部分的螞蟻(90%)跟隨著前人的腳步，而一部分的螞蟻(10%)則是探索其他的可行路徑，來搜尋可能的結果。

權重的計算：該作者提供了一個公式來計算每個選擇的權重。當然，已經走過的路徑不會被考慮在內，其中r為我們已知的起始點，k為螞蟻可選的移動目標。

開發：在可選的路徑中，選擇權重最大的一條來移動。

探索：在可選的路徑中，依照權重隨機選擇一條來移動，在實作中，由於浮點數精度問題，所有的權重總和不一定為100%，因此我們使用累積權重的方式，使得浮點數誤差的因素可被排除。

浮點數的精度問題，是源自於我們生活中十進位的小數點，在轉換成由(IEEE 754)規範中的二進位表示法的時候，並沒有辦法完全的對應，因此在小數點後面的幾個位數中，往往和實際的數字不合，當我們累積很多次的誤差之後，很容易讓原本極小的誤差進行放大，導致在計算最終結果時，誤差已經成為不容忽視的結果。

因此我們更改了實作方式，將權重變更為累進制，並將其上限修正為1，如此一來，從浮點數表示法而來的誤差將可分散在各個數值中，而不會因此而放大問題。

* 1. 局部更新費洛蒙

在AS中，並沒有局部更新費洛蒙的過程，螞蟻要走完全部的路徑(一整圈)後，才會進行更新費洛蒙。到了ACS，作者提供了四種局部更新的方法，與一個通用的公式，而我們的實作中，實作了其中三種，關於ant-Q的部分，由於作者並沒有提到的值，若要比較不同的差異所造成的影響，則和本次實作的方向有所不同，因此就沒有實作ant-Q的部分。

* 1. 全域揮發費洛蒙

在全域更新費洛蒙的階段中，由於不同的方法在這個階段有所差異，我們將其拆分成兩個部分，分別為全域揮發費洛蒙、螞蟻散發費洛蒙兩個部分，並依照不同的方法進行說明。

不論是AS或ACS中，作者賦予了一個衰退值，使得費洛蒙不會不斷的累積，造成後續的行為影響力降低。

* 1. 螞蟻散發費洛蒙

在AS中，所有的螞蟻都會散發費洛蒙，使得在探索的過程中，表現最佳的螞蟻會被其他的螞蟻給稀釋，使得找到最佳解的影響力不足。而ACS改善了此缺點，只有在該次模擬中，表現最好的那隻螞蟻才有能力留下費洛蒙，使得在探索的過程中，最佳解仍然可以發揮他的影響力。

1. 模擬結果
   1. 各方法收斂過程

在模擬過程中，我們發現在250次迭代後，四種方法都幾乎不再進行變化，因此我們將前250次迭代的結果做成圖，而在之後的列表中列出我們的所有數據，分別是該方法的最佳值，以及模擬100次的平均結果。

ACS有三種方式是ACS with full version、ACS with 和ACS without local updating，其最終收斂的平均值約落在440、480和460，ACS中最好的ACS with full version收斂的平均值也優於AS。可知ACS with full version距離的平均值是所有方法中最靠近最佳解的方法。

|  |  |
| --- | --- |
|  |  |
| 圖一：Ant system | 圖二：ACS without local updating |
|  |  |
| 圖三：ACS with =0 | 圖四：ACS with full version |

* 1. 各方法最終結果分布

為了避免模擬的結果受到隨機性影響，我們在各種方法都模擬100次，將其結果依照5的區間列出，來看出他們分布的關係。分布範圍越發散的方法，找出最佳解的機率就會降低。

由下面四張圖可知，分布區間最發散的是ACS with 這個方法，依序是ACS without local updating、AS，最集中的則是ACS with full version這個方法，而集中的地方也靠近該方法的最佳解。

|  |  |
| --- | --- |
|  |  |
| 圖五：Ant system | 圖六：ACS without local updating |
|  |  |
| 圖七：ACS with =0 | 圖八：ACS with full version |

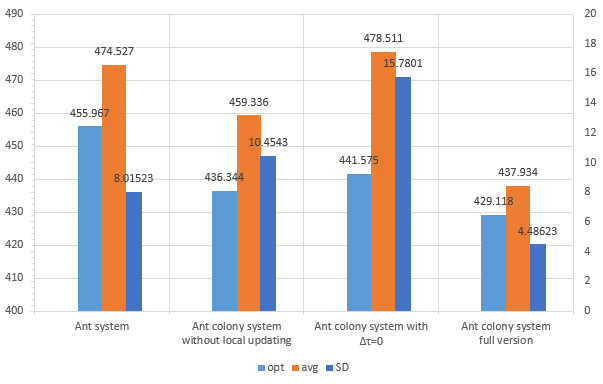
* 1. 各方法最佳結果路徑圖

在各種方法分別模擬100次後，我們紀錄他最好的那個結果，並且把路徑畫出來。其中AS是所有方法中得到距離結果最長的方法，在畫圖的過程中甚至可以看到路徑圖上有兩個地方交叉。而ACS三個方法中，又以ACS with full version的結果最好。

|  |  |
| --- | --- |
|  |  |
| 圖九：Ant system | 圖十：ACS without local updating |
|  |  |
| 圖十一：ACS with =0 | 圖十二：ACS with full version |

* 1. 各方法比較

根據收斂過程的圖表比較可知，ACS with full version與最佳解之間的差距最小，且模擬結果的分佈最集中，大多集中於435~440之間。其平均值結果更是逼近其他方法的最佳解。根據下圖可知，不管是最佳解、平均距離還是標準差皆顯示ACS with full version是四種方法中最為傑出的。



圖十三：四種方法綜合比較圖

1. 結論

就結果而言，ACS with full version有著最好的結果，並且在平均值與數據的分布上，也有不俗的表現，而且即使是平均值，也是幾乎比其他的方法好上許多。而第三個方法ACS with =0的結果，反而是相當糟糕的，不僅是平均值遠超其他的方法，而標準差也是遠遠的不如。

而同樣沒有進行局部費洛蒙更新的AS和ACS without local updating，則是發現ACS會比AS好一些，但是在數值的分布上略為發散。

再來是收斂的速度，我們可以發現四種方法大概在50個迭代之後，收斂就會趨緩，然而在後續的演化中，仍然會有後續的進步，因此在模擬時，我們仍然將其迭代2000次。

1. 參考資料

[1] M. Dorigo, “Optimization, learning and natural algorithms,” Ph.D. dissertation, DEI, Politecnico di Milano, Italy, 1992 (in Italian).

[2] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," in IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, pp. 53-66, April 1997.

[3] 陳士杰，Foundations of Ant Algorithm，http://debussy.im.nuu.edu.tw/sjchen/Project\_Courses/ML/AntAlgo.pdf