資工三A 蔡政育 104502510 作業四

程式介面說明

在ubuntu中執行執行檔,會印出執行結果。

E-C-F-B-A-I-D-G-H-J Path Length: 106.0

每一次執行結果皆不同。

程式碼說明

利用python進行實作,所有程式都在SA_TSP.py。

def readfile():

讀取TSP.txt檔案,並將檔案轉成array,讓程式比較好處理。 def generate_new_state(S):

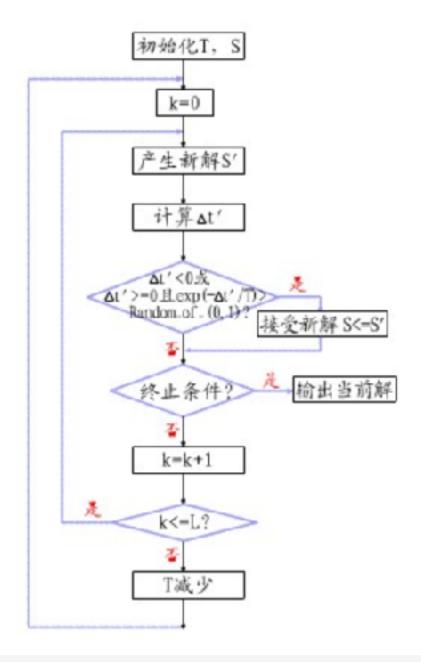
產生新的路徑,方法為隨機交換兩個城市。 def calculate distance(S):

計算路徑所要花的距離,使用for迴圈迭代計算。

main

實作退火演算法,主要架構下面會說明。

Simulated Annealing演算法說明



```
Procedure TSPSA:
begin
  init-of-T; { T為初始溫度}
  S={1, ....., n}; {S為初始值}
  termination=false;
  while termination=false
    begin
      for i=1 to L do
         begin
             generate(S'form S); { 從當前迴路S產生新迴路S'}
             Δt:=f(S'))-f(S);{f(S)為路徑總長}
             IF(\Delta t < 0) OR (EXP(-\Delta t / T)>Random-of-[0,1])
             S=S';
             IF the-halt-condition-is-TRUE THEN
             termination=true;
         End;
      T lower;
    End;
End
```

模提退火演算法新鲜的產生和接受可分為如下四個步驟;

- 第一步是由一個產生函數侵需前額產生一個位於損空間的新解:為便於後續的計算和接受,減少減算古耗時,通常選擇由當前新額經過簡單地變換却可產生新報的方法,如對構成新解的全部或部分元素進行置換。互換等,註意到產生新報的變換方法決定了智前新報的鄰班結構,因而對冷卻進度表的進取有一定的影響。
- 第二步是計算與新報所對應的目標函數差。因為目標函數差值主變換総分產生。所以目標函數差的計算最好按請量計算、事實表明。對大多數應用查言。這是計算目標函数差的最快方法。
- 第三步是判斷訴解是否被接受。判斷的依據是一個接受準則。嚴常用的接受準則是Metropolb準則: 者Δt'<0則接受5'作為新的當前解3。否則以根準exp/~Δt'/T)接受3'作為新的當前解3。
- 第四步是置新解技確定接受時, 円折解代替當前額, 這隻需將置前額中對應於產生新解時的變換部分予以實明, 同時修正日標函數值即可。此時, 當前解實現了一次迭代, 可在此基礎上開始下一輪試驗, 而當新解被判定為搶棄時, 則在原當前額的基礎上繼續下一輪試驗。

執行結果

執行的路徑結果大部分的數值都在100~200之間。

A-H-E-J-B-D-G-I-C-F Path Length: 85.0

作業四因為網路上有比較多的資源、不會有太多的複雜計算,所以實作起來比 之前作業簡單。

參考資料

http://wiki.mbalib.com/zh-tw/ %E6%A8%A1%E6%8B%9F%E9%80%80%E7%81%AB%E7%AE%97%E 6%B3%95

https://ithelp.ithome.com.tw/articles/10186172

https://www.cnblogs.com/heaad/archive/2010/12/20/1911614.html

http://wiki.mbalib.com/zh-tw/ %E6%A8%A1%E6%8B%9F%E9%80%80%E7%81%AB%E7%AE%97%E 6%B3%95