

# Memetic Algorithm

## 簡要報告

戴宏達

2012/12/20

# Memetic Algorithm(MA)

---

- 是使用Local Search優化個體的 Evaluation Algorithm(EA)
- 根據文化進化(Meme)的概念，多用來求解組合最佳化問題
- 與Genetic Algorithm(GA)類似，但有些相異處
  - ▣ MA在每一代會做區域優化
  - ▣ 若某世代符合特定條件，MA會對大部分個體進行突變；GA則是對某一個體進行，改變程度較小

# 執行步驟

---

- Step1 產生 Initial Solution
  - Step1.1 - 產生第一代Population
  - Step1.2 - Local Search優化
- Step2. 選出最好的個體作為解答
- Step3. 產生下一代
  - Step3.1 -個體競爭選出父母
  - Step3.2 -父母交配產生子代
  - Step3.3 -子代有一定機率變異
  - Step3.4 -檢查變異後的子代有無錯誤，並修復
  - Step3.5 - Local Search優化
- Step4. 合併兩個世代
- Step5. 若未達成終止條件，則goto Step2

# 程式實做 (1/3)

---

- 以Java實做
- 求解Travelling Salesman Problem(TSP)問題
  - ▣ 產生Solution方式
    - Greedy
    - Random
  - ▣ LocalSearch方式
    - None
    - 2-opt First improvement
    - 2-opt Best improvement
- 可視化
  - 使用 Jung Graph Framework 完成
  - 參考影片

# 程式實做 (2/3)

---

## □ 第一個世代的Solution

### ▣ Greedy

- 從第一個City開始，每次都找距離自己最近的City作為下一個City，直到全部連接

### ▣ Random

- 亂數把所有City連接起來

## □ 第二個世代後一律用Random法產生子代

# 程式實做 (3/3)

---

## □ LocalSearch方式解釋

### ▣ None

- 不做LocalSearch

### ▣ 2-opt First

- 循序對每個個體的City兩兩交換，直到找到第一個比目前好的狀況，就使用該狀況

### ▣ 2-opt Best

- 循序對每個個體的City兩兩交換，找出所有結果，並挑出最好的狀況使用

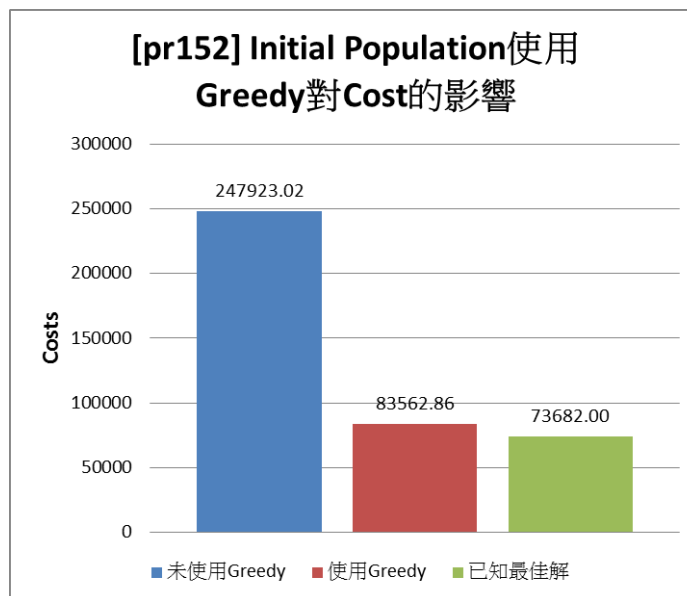
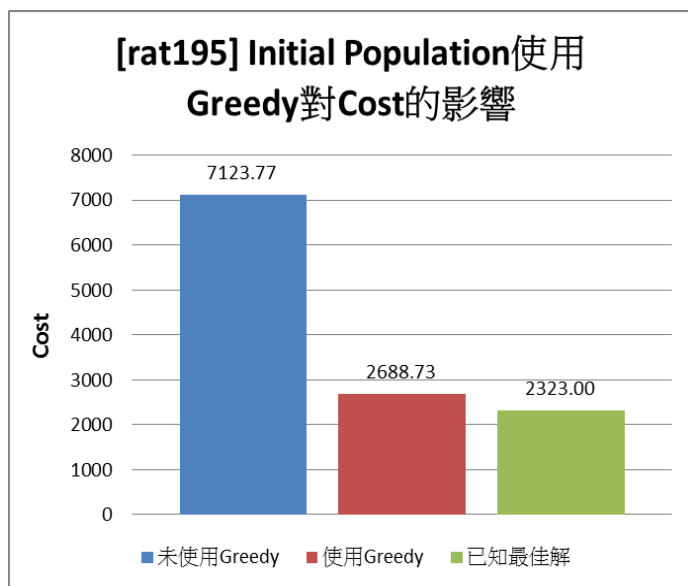
# 測試參數

---

- 問題參考TSPLIB
  - att48, gr137, pr152, rat195
- 自變數
  - ▣ LocalSearch
    - None, 2-opt First, 2-opt Best
  - ▣ PopulationSize
    - 2, 10, 20
  - ▣ MutationProbability
    - 0.3, 0.5, 0.7
  - ▣ Initial Solution Greedy
    - False, True
- 依變數
  - ▣ Path Cost
    - 行經路徑長，越低越好
- 實驗方法
  - ▣ 以下結果為每個case重複20次取平均
  - ▣ 最佳解參考
    - <http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/STSP.html>

# 測試結果 – 實驗一

## □ 第一代Population使用Greedy對Cost的影響



模擬狀況

LocalSearch: 2-opt First

popSize: 10

MutProbability: 0.5

## □ 小結：

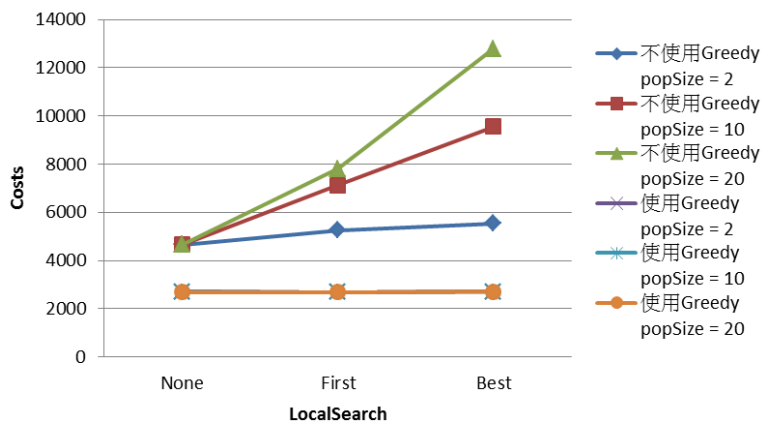
- 未使用Greedy與使用Greedy的Cost差距很大，且使用Greedy後的Cost與最佳解差距都不超過15%，效果顯著
- 適當使用Greedy對Cost有顯著影響



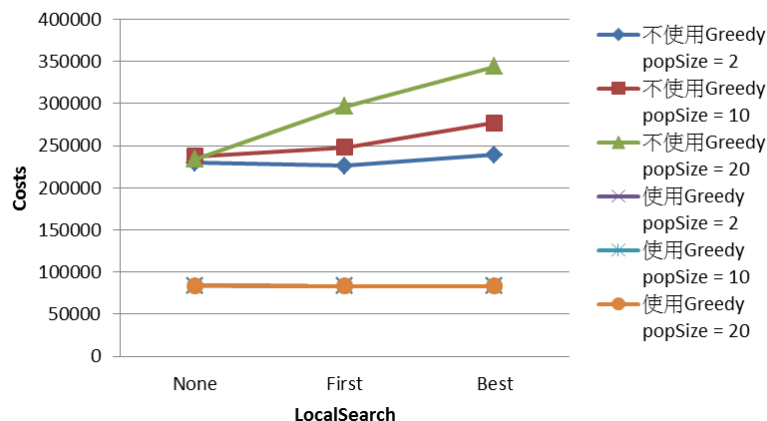
# 測試結果 – 實驗二 (1/2)

## 是否使用Greedy在各population下LocalSearch對Cost的影響

[rat195] 是否使用Greedy在各population下LocalSearch對Cost的影響

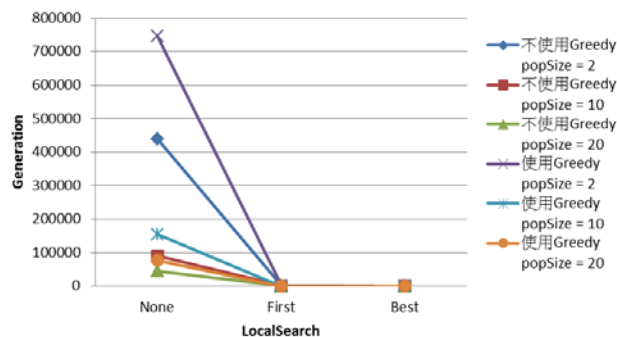


[pr152] 是否使用Greedy在各population下LocalSearch對Cost的影響

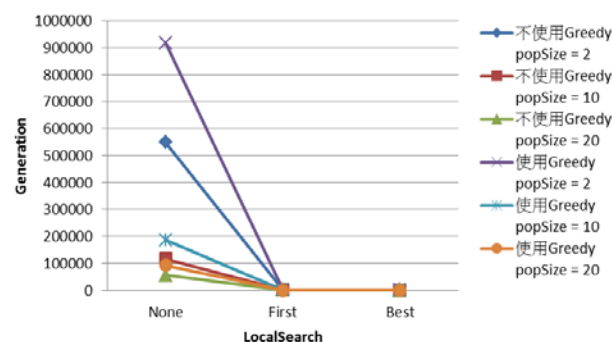


模擬狀況  
MutProbability: 0.5

[rat195] 是否使用Greedy在各population下LocalSearch對Generation的影響



[pr152] 是否使用Greedy在各population下LocalSearch對Generation的影響



# 測試結果 – 實驗二 (2/2)

---

## □ 小結：

### ▣ 不使用Greedy

- LocalSearch與PopulationSize的影響很明顯，不進行LocalSearch(就是None)且Population Size越小，在同樣時間可經過更多世代，Cost較低、結果較好。
- 相反地，用2-opt Best進行LocalSearch且PopulationSize越大，世代數目差距很大，造成較壞的結果。
- LocalSearch花費越多時間，越不利於世代交替，使結果變壞。且差距會被放大，使Cost的結果差距極大
  - 費時程度：Best > First > None

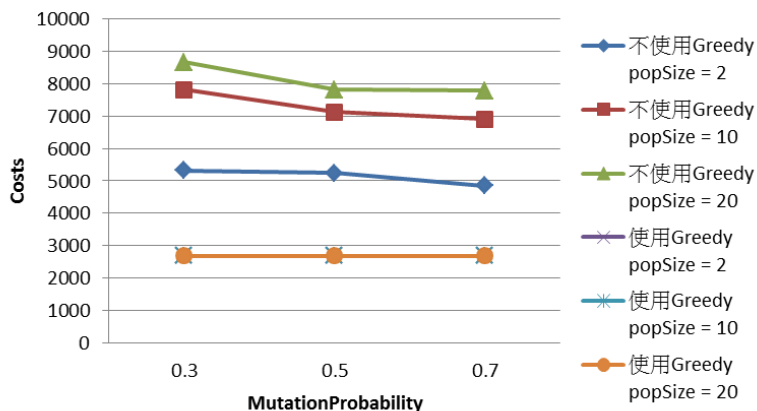
### ▣ 使用Greedy

- 使用Greedy的狀況下都有相對低的Cost，LocalSearch與PopulationSize的影響極不明顯。

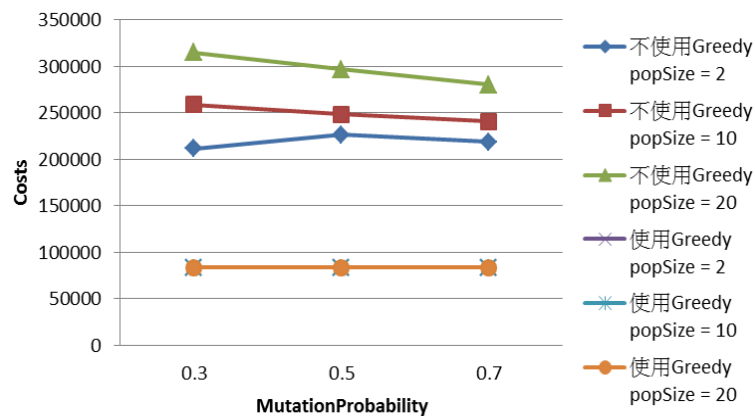
# 測試結果 – 實驗三 (1/2)

是否使用Greedy在各population下mutation對Cost的影響

[rat195] 各population下mutation對Cost的影響

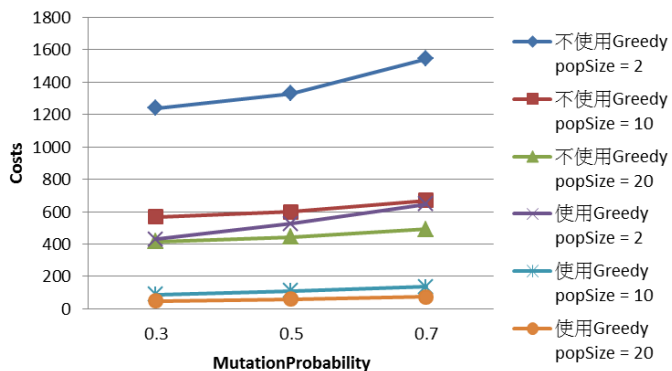


[pr152] 各population下mutation對Cost的影響

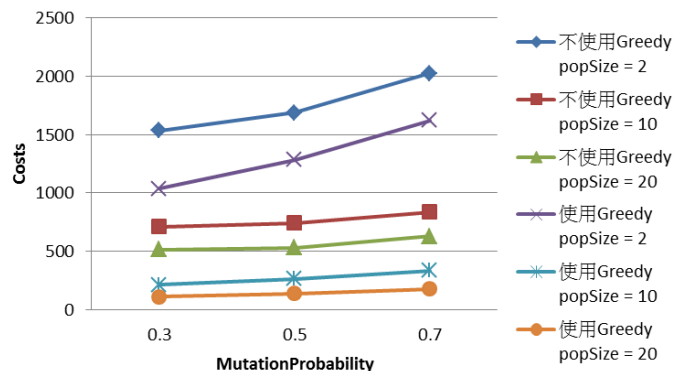


模擬狀況  
LocalSearch: 2-opt First

[rat195] 各population下mutation對Generation的影響



[pr152] 各population下mutation對Generation的影響



# 測試結果 – 實驗三 (2/2)

---

## □ 小結：

### ▣ 不使用Greedy

- MutationProbability越大，會產生較多的世代，有較好的結果。
- PopulationSize越小時，會產生較多的世代，而且差距非常明顯，會導致較好的結果。

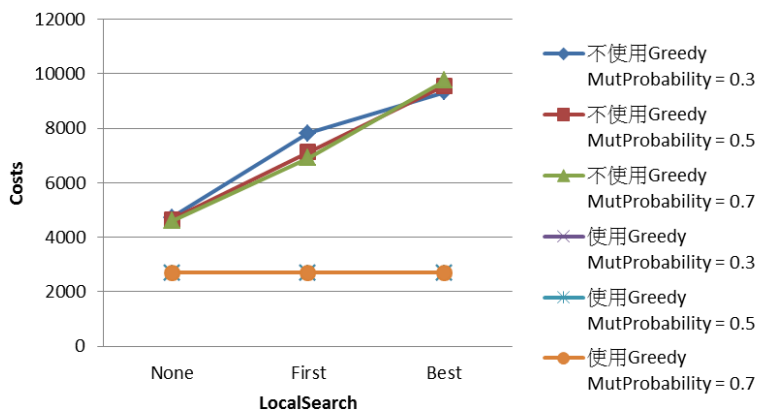
### ▣ 使用Greedy

- 使用Greedy的狀況下都有相對低的Cost，PopulationSize與MutationProbability的影響極不明顯。

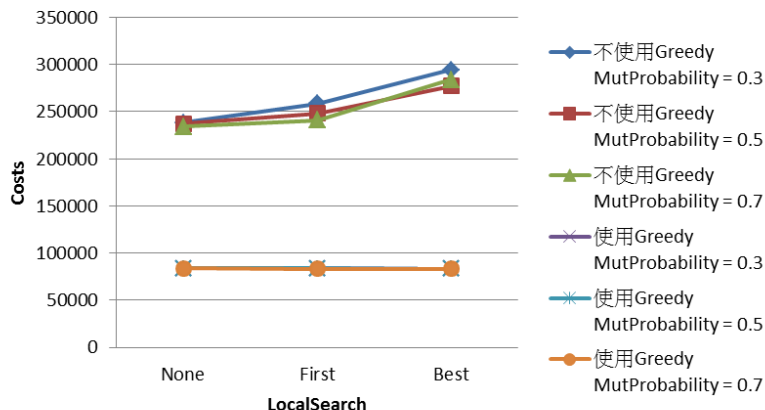
# 測試結果 – 實驗四 (1/2)

## 各MutationProbability下LocalSearch對Cost的影響

[rat195] 各MutationProbability  
下LocalSearch對Cost的影響

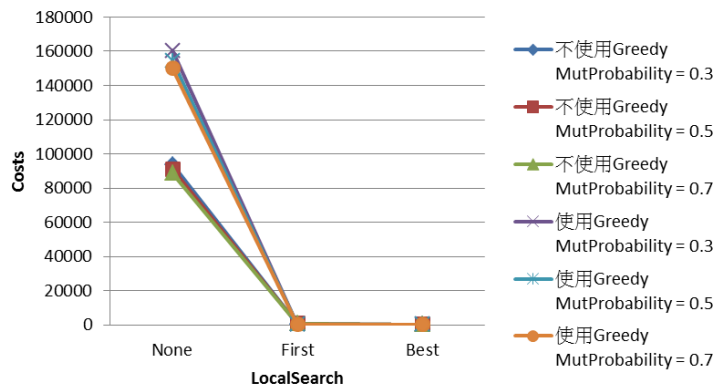


[pr152] 各MutationProbability  
下LocalSearch對Cost的影響

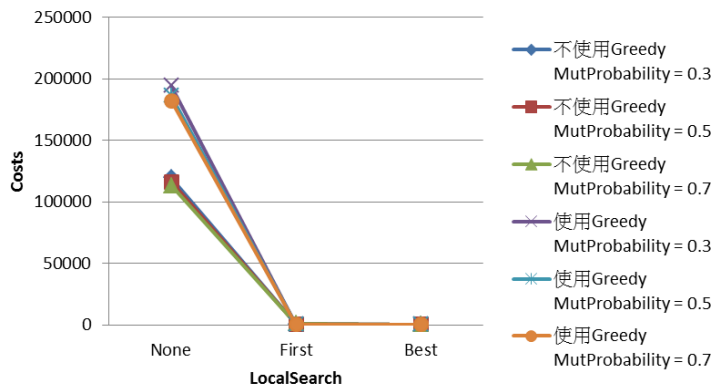


模擬狀況  
PopulationSize: 10

[rat195] 各MutationProbability  
下LocalSearch對Cost的影響



[pr152] 各MutationProbability  
下LocalSearch對Cost的影響



# 測試結果 – 實驗四 (2/2)

---

## □ 小結：

### ▣ 不使用Greedy

- LocalSearch花費越多時間，越不利於世代交替，使結果變壞。且差距會被放大，使Cost的結果差距極大
- MutationProbability的影響對世代及Cost皆不明顯

### ▣ 使用Greedy

- 使用Greedy的狀況下都有相對低的Cost，PopulationSize與Mutation的影響極不明顯。

# 測試結果 – 總結

---

## □ 總結

▣ 由以上實驗結果交叉比較，可得出下列結論：

- 第一世代使用Greedy對Cost有非常明顯的改善，且用Greedy的解與最佳解差距不大
- 使用Greedy後，其他變因相較之下影響很小
- LocalSearch花費越少時間，越有利於世代交替，使Cost變小，影響明顯。
  - 費時程度：Best > First > None
- PopulationSize越小時，會產生較多的世代，會導致較好的結果。
- MutationProbability越大，會產生較多的世代，有較好的結果，影響不明顯。

# 參考資料

---

- A Framework for Memetic Algorithms
  - ▣ 作者: Fengjie Wu
- A Tutorial for Competent Memetic Algorithms: Model, Taxonomy, and Design Issues
  - ▣ 作者: Natalio Krasnogor and Jim Smith
- Memetic algorithms and memetic computing optimization: A literature review
  - ▣ 作者: Ferrante Neri, Carlos Cotta
- Memetic Algorithms for the Resource-Constrained Project Scheduling Problem
  - ▣ 作者: Zhi-jie Chen, Dr. Chiuh-Cheng Chyu
- OpenSource Memetic Algorithm (C#)
  - ▣ 作者: ygf
  - ▣ 連結: <https://github.com/ygf/metaheuristics>
- Jung - Java Universal Network/Graph Framework
  - ▣ 連結: <http://jung.sourceforge.net/>