

Introduction to Artificial Intelligence -

Homework 2

NE6114011 人工智慧科技碩士學位學程碩一 楊雲翔

1 Brute Force Method 實現方式

本次作業的題目為工作指派問題，為一經典的組合問題，我採用的 Brute Force method 為窮舉法，舉出所有可能的組合，並計算出各組合的 cost，進而找出最佳組合，使用這個方法必定可以找到最佳解。

2 Genetic Algorithm 實現方式

以下是本次作業中，我實現的基因演算法的步驟

1. Initialize :

初始化母體，隨機產生 N 條染色體(解)作為第一代

2. Crossover :

從母體中隨機選擇 K 組父母進行交配(crossover)，產生 K 組子代，並將子代加入母體中，本次作業採用的交配方法採用的是 partially mapped crossover，作法為：「隨機選擇兩個斷點，兩染色體交換該片段以產生子代」

3. Mutation :

從母體中隨機選擇 P 條染色體進行突變，並將突變後的染色體加入子代在本次作業我總共實現了三種突變方法，分別為：

- A. Inversion mutation：隨機挑選染色體的某一個片段，並反轉該片段
- B. Swap mutation：隨機挑選染色體的兩個基因，交換彼此
- C. Scramble mutation：隨機挑選染色體的某一個片段，隨機打亂該片段內的基因

4. Evaluation :

評估經選擇、交配、突變後存在於母體內的 N+K+P 條染色體的適應度，以作為下一輪選擇要保留染色體的依據，本次作業我採用的適應度計算公式如下：

$$fitness(cost) = \max(\alpha * (cost_{max} - cost_{min}), 10e - 5) + (cost_{max} - cost)$$

- $cost$ ：該染色體在工作指派問題中的成本
- $cost_{max}$ ：在本輪母體中最高的成本
- $cost_{min}$ ：在本輪母體中最低的成本
- α ：超參數，用來控制本輪的最小適應度

5. Selection：

根據計算出的適應度，來選擇要保留到下一代的染色體，本次作業中我實現了兩種不同的選擇方法：

A. Deterministic：根據適應度大小排序染色體，保留排名前 K 條染色體

B. Random：隨機抽選 K 條染色體，根據適應度大小決定抽中的機率

重複步驟(2)~(5) t 次，在 t 次中擁有最佳解的染色體即為最佳解

3 Genetic Algorithm 分析

實驗說明：

隨機產生 100 組長度為 8 的 problem，使用窮舉法得到實際答案，再利用不同參數的基因演算法去獲取最佳解，去評估正確率與花費時間，正確率計算公式為（ $\frac{\text{基因演算法最佳解成本} - \text{實際最佳解成本}}{\text{總問題數}}$ ）之問題數/總問題數目

參數說明：

參數	N	K	P	Iteration	Selection	Mutation
說明	父母總數	子代數目	突變數目	迭代次數	選擇方法	突變方法

實驗結果

N	K	P	Iteration	Selection	Mutation	Accuracy
50	10	5	300	random	Inversion	0.45
					Swap1	0.75
					Swap2	0.75
					Scramble	0.07
				Deterministic	Inversion	0.65
					Swap1	0.93
					Swap2	0.93
					Scramble	0.32
100	20	10	300	random	Inversion	0.67
					Swap1	0.86
					Swap2	0.89
					Scramble	0.28
				Deterministic	Inversion	0.85

					Swap1	0.98
					Swap2	0.98
					Scramble	0.69
200	40	20	300	random	Inversion	0.85
					Swap1	0.95
					Swap2	0.96
					Scramble	0.58
				Deterministic	Inversion	0.95
					Swap1	0.98
					Swap2	0.99
					Scramble	0.91
300	60	30	300	random	Inversion	0.96
					Swap1	0.96
					Swap2	0.98
					Scramble	0.67
				Deterministic	Inversion	0.98
					Swap1	0.99
					Swap2	0.99
					Scramble	0.98
300	60	30	500	random	Inversion	0.93
					Swap1	0.95
					Swap2	0.98
					Scramble	0.78
				Deterministic	Inversion	0.99
					Swap1	0.99
					Swap2	1.0
					Scramble	0.99

*swap1 表交換一次，swap2 表交換兩次

通過以上實驗，可以觀察到：

1. Population size：提升 population size 一定可以提升找到最佳解的機率
2. Selection：在大多數情況下，Deterministic 的選擇法往往比 random 更佔優勢
3. Iteration：提升 Iteration 次數普遍可以提升找到最佳解的機率
4. Mutation：在三種方法中，Swap 往往表現地比 Inversion 與 Scramble 來的好，並且交換兩次也比交換一次來得好

綜合以上分析，最後採用的參數組合為：

N=300, K=60, P=30, Iteration=500, Selection=Deterministic,
Mutation=Swap2

4 比較 Brute-Force 與 Genetic algorithm

方法	優點	缺點
Brute-Force	1. 當組合數較少時，速度相較基因演算法來的快 2. 一定可以找出最佳組合	當組合數很多時，速度非常慢，演算法複雜度為 $O(n!)$
Genetic algorithm	速度不會受到問題的影響，就算組合數很多，依然可以保持一定速度	可能會陷入局部最佳解

實作時的心得：

1. 暴力法：

由於我採用的暴力法為窮舉法，實作起來相較於簡單，唯一需要思考的部分只有遞迴的部分要怎麼實現，才不至於導致效率過差

2. 基因演算法：

基因演算法的實作相對困難，同時，基因演算法的各個環節都有著許多種不同的版本，需要花費時間去一一實現，其中最難的部分是 crossover 過程中需要避免兩染色體交換片段後，會有重複的基因被使用，要如何透過映射的方式去解決這樣的情況。