
Práctica 3

Metaheurísticas

Amin Kasrou Aouam



**UNIVERSIDAD
DE GRANADA**

2021-06-22

Índice

Práctica 3	3
Introducción	3
Algoritmos	3
Enfriamiento simulado	3
Búsqueda multiarranque	4
Implementación	4
Instalación	5
Ejecución	5
Análisis de los resultados	6
Enfriamiento simulado	6
Búsqueda multiarranque	6

Práctica 3

Introducción

En esta práctica, usaremos distintos algoritmos de búsqueda, basados en trayectorias, para resolver el problema de la máxima diversidad (MDP). Implementaremos:

- Enfriamiento simulado
- Búsqueda multiarranque

Algoritmos

Enfriamiento simulado

El enfriamiento simulado es un algoritmo de búsqueda basado en trayectorias, con un criterio probabilístico basado en la termodinámica.

El procedimiento general del algoritmo queda ilustrado a continuación:

Input: A list $[a_i], i = 1, 2, \dots, n$, that contains the components of a solution
Output: Processed list

```

1 currentSol ← initialize()
2 T0 ← initialize()
3 Tf ← initialize()
4 bestSol ← currentSol
5 T ← Tf
6 while T > Tf do
7   | neighbour ← generateNeighbour(currentSol)
8   | delta ← fitness(neighbour) − fitness(currentSol)
9   | if delta > 0 ∨ acceptNeighbour(neighbour) then
10  | | currentSol ← neighbour
11  | | if fitness(currentSol) > fitness(bestSol) then
12  | | | bestSol ← currentSol
13  | | end if
14  | end if
15  | coolDown(T)
16 end while
17 return bestSolution

```

Búsqueda multiarranque

La búsqueda multiarranque es un algoritmo que itera las 2 etapas siguientes:

1. Generación de una solución inicial
2. Búsqueda local

Es una variante simple de implementar a partir de una búsqueda local

El procedimiento general del algoritmo queda ilustrado a continuación:

```
Input: A list  $[a_i], i = 1, 2, \dots, m$ , the solution  
Output: Processed list  
1 Solutions = []  
2 initialSolutions  $\leftarrow$  getRandomSolutions()  
3 Solutions.append(firstSolution)  
4 maxIterations  $\leftarrow$  1000  
5 for i  $\leftarrow$  0 to maxIterations do  
6   for element in initialSolutions do  
7     lastSolution  $\leftarrow$  getLastElement(initialSolutions)  
8     while neighbour  $\leq$  lastSolution do  
9       neighbour  $\leftarrow$  getNeighbouringSolution(lastSolution)  
10      Solutions.append(neighbour)  
11      lastSolution  $\leftarrow$  getLastElement(neighbour)  
12    end while  
13    finalSolution  $\leftarrow$  getLastElement(Solutions)  
14  end for  
15 end for  
16 return finalSolution
```

Implementación

La práctica ha sido implementada en *Python*, usando las siguientes bibliotecas:

- NumPy
- Pandas

Instalación

Para ejecutar el programa es preciso instalar Python, junto con las bibliotecas **Pandas** y **NumPy**.

Se proporciona el archivo `shell.nix` para facilitar la instalación de las dependencias, con el gestor de paquetes `Nix`. Tras instalar la herramienta `Nix`, únicamente habría que ejecutar el siguiente comando en la raíz del proyecto:

```
1 nix-shell
```

Ejecución

La ejecución del programa se realiza mediante el siguiente comando:

```
1 python src/main.py <dataset> <algoritmo>
```

Los parámetros posibles son:

dataset	algoritmo
Cualquier archivo de la carpeta data	annealing
	multistart

También se proporciona un script que ejecuta 1 iteración de cada algoritmo, sobre cada uno de los *datasets*, y guarda los resultados en una hoja de cálculo. Se puede ejecutar mediante el siguiente comando:

```
1 python src/execution.py
```

Nota: se precisa instalar la biblioteca `XlsxWriter` para la exportación de los resultados a un archivo Excel.

Análisis de los resultados

Enfriamiento simulado

A	B	C	D	E
dataset	desviacion distancia	desviacion tiempo	media distancia	media tiempo
MDG-a_40_n2000_m200.txt	0	0	6286	39.0129680633545
MDG-b_8_n500_m50.txt	0	0	617131.93	13.4176297187805
GKD-c_20_n500_m50.txt	0	0	15461.40652	12.9850316047669
MDG-a_38_n2000_m200.txt	0	0	6005	39.2592134475708
MDG-b_9_n500_m50.txt	0	0	620289.250000001	13.2476000785808
MDG-a_31_n2000_m200.txt	0	0	6054	38.4116442203522
MDG-a_36_n2000_m200.txt	0	0	6155	38.9764051437378
MDG-b_5_n500_m50.txt	0	0	618252.26	14.0317611694336
MDG-b_4_n500_m50.txt	0	0	617502.56	13.2216610900508
MDG-b_6_n500_m50.txt	0	0	612951.49	13.0719683170319
MDG-b_7_n500_m50.txt	0	0	618987.4	13.4407684803009
MDG-b_2_n500_m50.txt	0	0	615027.350000001	13.3080396652222
MDG-a_34_n2000_m200.txt	0	0	6065	38.6733222007751
MDG-b_3_n500_m50.txt	0	0	597546.030000001	13.1623504161835
MDG-b_1_n500_m50.txt	0	0	624716.770000001	13.0831868648529
MDG-a_33_n2000_m200.txt	0	0	6173	38.4686946868896
GKD-c_13_n500_m50.txt	0	0	15824.43113	12.9513597488403
MDG-a_35_n2000_m200.txt	0	0	6002	38.5037672519684
GKD-c_12_n500_m50.txt	0	0	15676.7212	13.06109002309418
MDG-a_32_n2000_m200.txt	0	0	6060	38.5681200027466
MDG-b_10_n500_m50.txt	0	0	623030.43	13.0907363891602
GKD-c_11_n500_m50.txt	0	0	15946.70298	13.3723454475403
GKD-c_14_n500_m50.txt	0	0	15592.69983	13.1872181892395
GKD-c_15_n500_m50.txt	0	0	15492.44688	13.0802743434906
GKD-c_17_n500_m50.txt	0	0	15344.24749	13.4428441524506
GKD-c_16_n500_m50.txt	0	0	16053.98084	12.8006107807159
MDG-a_37_n2000_m200.txt	0	0	6342	38.8236746788025
GKD-c_19_n500_m50.txt	0	0	15666.0342	13.1313874721527
MDG-a_39_n2000_m200.txt	0	0	6089	38.8069708347321
GKD-c_18_n500_m50.txt	0	0	15613.95454	13.4720151424408

Figura 1: Enfriamiento simulado

El tiempo de ejecución varía ligeramente según el dataset:

- Dataset con n=500: 13 segundos
- Dataset con n=2000: 38-39 segundos

La distancia total obtenida, toma valores muy diferentes según el dataset en el que se ejecuta.

Búsqueda multiarranque

Desafortunadamente, un *bug* en el script de ejecución ha hecho que la segunda pestaña de la hoja de cálculo únicamente tenga el valor de los datasets.