Universidad de La Laguna

Práctica 9: Maximum diversity problem

Informe

Autor: Guillermo Hernández González

Alu: Alu0101121529

C/ Padre Herrera s/n
38207 La Laguna
Santa Cruz de Tenerife. España

T: 900 43 25 26

ull.es

Índice

Local Search con Greedy Constructivo:

Local Search con Greedy Destructivo:

Explicación de los algoritmos:

Primera implementación: Greedy Constructivo

Esta aproximación a la resolución del problema *Maximum diversity problem* se basa en ir construyendo la solución a base de ir cogiendo los mejores nodos en cada caso.

Lo que hay que hacer, es usando la distancia entre el centro y un nodo como la heurística que queremos maximizar para seleccionar en cada iteración los nodos que pertenecen a la solución.

Primero llenamos un vector con todos los elementos del problema, usando todos esos elementos hacemos una primera aproximación del centro. Una vez tengamos ese centro, calculamos la distancia desde un punto a ese centro y el que esté más lejos lo metemos en la solución y lo eliminamos del vector de elementos, seguimos así hasta que la solución sea del tamaño que deseamos.

Segunda implementación: Greedy Destructivo

La idea de este algoritmo es la misma que el anterior, pero tiene una diferencia es como se empieza el mismo.

En el anterior empezábamos con un vector de elementos que contenía todos los nodos del problema y un vector solución vacío, en este caso, el vector solución tiene todos los elementos y vamos a ir eliminando elementos de este vector dependiendo de qué vector esté a menor distancia del centro.

Seguiremos eliminando hasta que el vector solución tenga el tamaño que deseamos.

Tercera implementación: Local Search

Este método se basa en la mejora de de la soluciones de los algoritmos anteriores y esto se consigue haciendo un ‘swap’ de algunos elementos de la solución si se encuentra unos elementos mejores en una solución vecina.

Para hacer esto lo que tenemos que hacer es encontrar el peor nodo de la solución actual y lo eliminamos. Luego buscamos un nodo mejor usando la

función objetivo y lo añadimos. Este intercambio se hará hasta que no se pueda mejorar nada más.

Estructura de clases

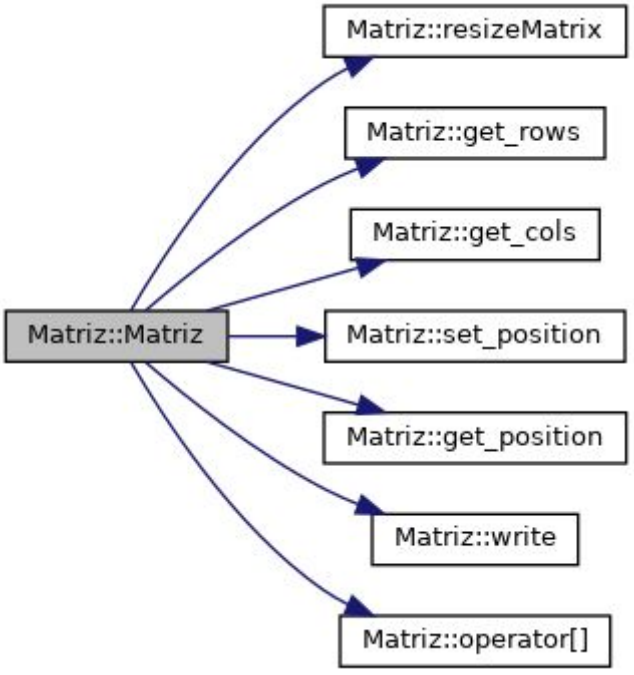
Clase Matriz:

La primera clase que se hizo fue una clase matriz en la cual guardaremos el grafo. Cada valor dentro de la matriz significa la distancia que hay entre los dos nodos que están simbolizados por su posición en la matriz. Por ejemplo, si en la posición [2,3] hay un 7, significa que del nodo 2 al nodo 3 hay una distancia de 7 unidades.

Aparte de esto, la clase matriz tendrá una serie de métodos para acceder a varios datos necesarios, como por ejemplo, el número de filas/columnas que significará el número de nodos que tendrá el grafo o tendrá un método que te devolverá todos los nodos del grafo en un vector.

Por último también tendrá un método para sacar los dos nodos cuya distancia sea máxima y luego otro que te devolverá un vector en el caso de que se de que hayan varios nodos en donde la distancia máxima coincida.

Grafo con la estructura:



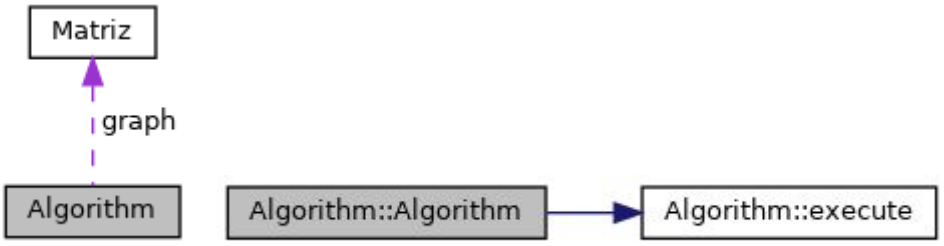
Link a la documentación de la clase

En este link se podrá ver la documentación de la clase Matrix con todos su métodos y el código fuente de los mismos.

Clase Algorithm:

La clase Algorithm sirve como clase virtual para hacer un patrón estrategia para el desarrollo de los algoritmos. Esta clase se compone de un constructor y un método virtual para ejecutar el algoritmo. La clase tiene un atributo privado que será un objeto de Matriz.

Grafo con la estructura:



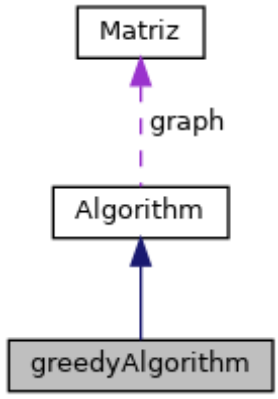
Link a la documentación de la clase

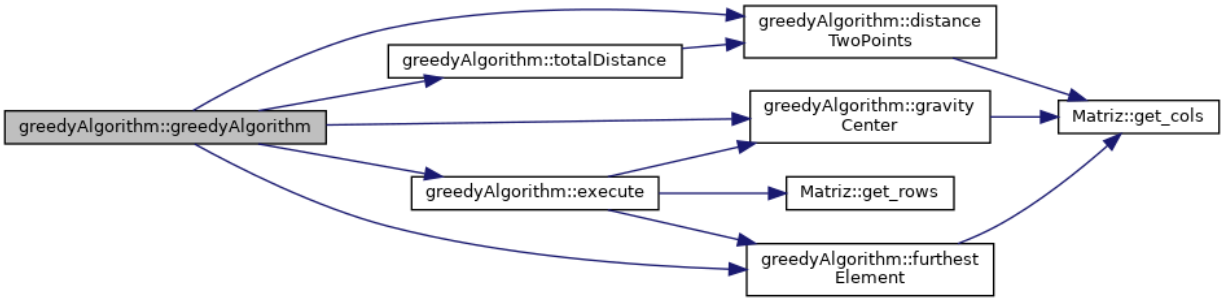
En este link se podrá ver la documentación de la clase Algorithm con todos su métodos y el código fuente de los mismos.

Clase greedyAlgorithm:

Esta clase heredará de Algorithm y es en donde se implementará el algoritmo greedy constructivo. Esta clase comparte constructor con Algorithm y añade varios métodos necesarios para el cálculo. Como por ejemplo el cálculo del centro y del nodo más alejado del centro.

Grafo con la estructura:





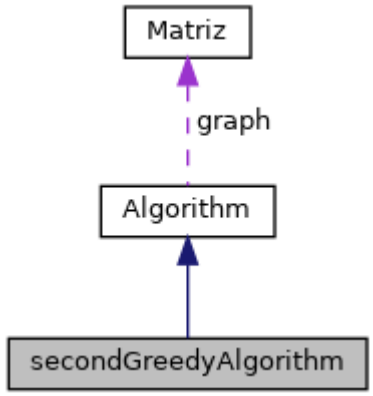
Link a la documentación de la clase:

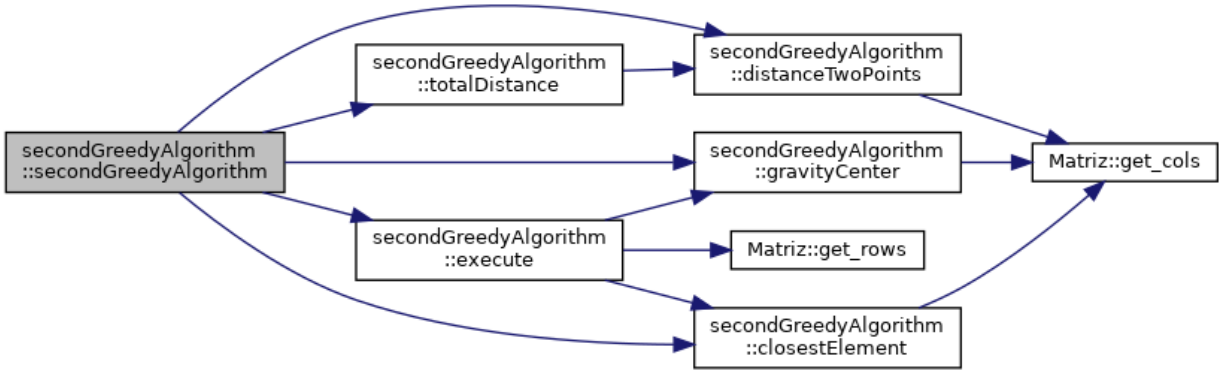
En este link se podrá ver la documentación de la clase greedyAlgorithm con todos su métodos y el código fuente de los mismos.

Clase secondGreedyAlgorithm:

Dentro de esta clase está la implementación del greedy destructivo. La estructura es muy similar a la anterior, solo que ahora el método closestElement te devuelve el elemento cuya distancia al centro es la menor

Grafo con la estructura:



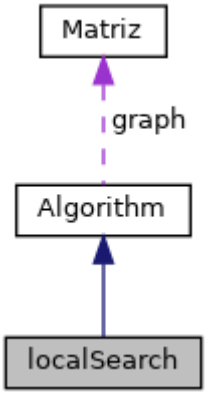


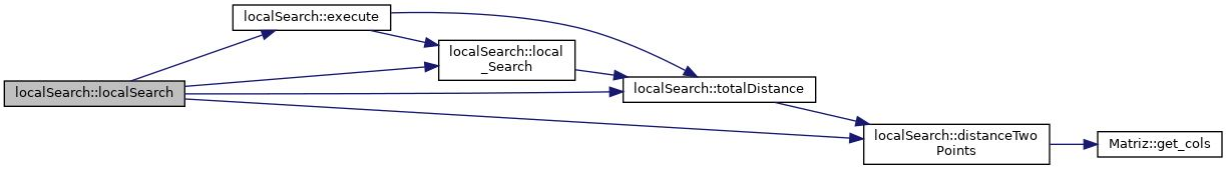
Link a la documentación de la clase:

En este link se podrá ver la documentación de la clase secondGreedyAlgorithm con todos su métodos y el código fuente de los mismos.

Clase localSearch:

Dentro de esta clase se encuentra la implementación de la clase localSearch y es una estructura parecida a la de los otros dos algoritmos ya que ésta también hereda de Algorithm solo que tiene un método extra para hacer la búsqueda local.





Link a la documentación de la clase:

Tablas y Gráficas.

Especificaciones de mi Pc:

Nombre del SO Microsoft Windows 10 Pro

Versión 10.0.18363 compilación 18363

Tipo de sistema PC basado en x64

Procesador AMD Ryzen 7 3700X 8-Core Processor, 3593 Mhz, 8 procesadores principales, 16 procesadores lógicos

Memoria física instalada (RAM) 63,9 GB

Tablas:

Tabla algoritmo Greedy constructivo:

Greedy constructivo						
Problema	n	K	m	Distancia	S	CPU (ms)
max_div_15_2.txt	2	15	2	11.8592	9 7	0.022
max_div_15_2.txt	2	15	3	25.7262	9 7 4	0.014
max_div_15_2.txt	2	15	4	48.4139	9 7 4 11	0.013
max_div_15_2.txt	2	15	5	73.5619	9 7 4 11 2	0.015
–						
max_div_15_3.txt	3	15	2	13.2732	12 9	0.011
max_div_15_3.txt	3	15	3	30.3241	12 9 5	0.014
max_div_15_3.txt	3	15	4	59.7638	12 9 5 11	0.017
max_div_15_3.txt	3	15	5	94.7487	12 9 5 11 14	0.019
–						
max_div_20_2.txt	2	20	2	8.51033	18 19	0.011
max_div_20_2.txt	2	20	3	21.9961	18 19 9	0.014
max_div_20_2.txt	2	20	4	39.5682	18 19 9 3	0.015
max_div_20_2.txt	2	20	5	61.2393	18 19 9 3 13	0.018
–						
max_div_20_3.txt	3	20	2	11.8003	13 14	0.012
max_div_20_3.txt	3	20	3	30.8727	13 14 8	0.015
max_div_20_3.txt	3	20	4	56.5347	13 14 8 3	0.017
max_div_20_3.txt	3	20	5	92.8298	13 14 8 3 17	0.021
–						
max_div_30_2.txt	2	30	2	11.6571	9 28	0.012
max_div_30_2.txt	2	30	3	28.9443	9 28 2	0.014
max_div_30_2.txt	2	30	4	52.7712	9 28 2 11	0.018
max_div_30_2.txt	2	30	5	80.9102	9 28 2 11 13	0.02
–						
max_div_30_3.txt	3	30	2	13.0737	17 7	0.013
max_div_30_3.txt	3	30	3	33.8423	17 7 24	0.018
max_div_30_3.txt	3	30	4	63.5184	17 7 24 14	0.021
max_div_30_3.txt	3	30	5	99.5088	17 7 24 14 15	0.025

Tabla algoritmo Greedy destructivo:

Greedy Destructivo						
Problema	n	K	m	Distancia	S	CPU (ms)
max_div_15_2.txt	2	15	2	11.8592	7 9	0.03
max_div_15_2.txt	2	15	3	23.7964	2 7 9	0.029
max_div_15_2.txt	2	15	4	46.6528	2 7 9 11	0.028
max_div_15_2.txt	2	15	5	73.5619	2 4 7 9 11	0.025
–						
max_div_15_3.txt	3	15	2	13.2732	9 12	0.036
max_div_15_3.txt	3	15	3	30.3241	5 9 12	0.037
max_div_15_3.txt	3	15	4	58.7287	5 9 12 14	0.033
max_div_15_3.txt	3	15	5	96.0858	4 5 9 12 14	0.031
–						
max_div_20_2.txt	2	20	2	8.51033	18 19	0.044
max_div_20_2.txt	2	20	3	21.9961	9 18 19	0.043
max_div_20_2.txt	2	20	4	39.5682	3 9 18 19	0.041
max_div_20_2.txt	2	20	5	60.4301	3 9 11 18 19	0.039
–						
max_div_20_3.txt	3	20	2	11.5909	3 17	0.053
max_div_20_3.txt	3	20	3	29.1574	3 13 17	0.052
max_div_20_3.txt	3	20	4	56.6903	3 13 14 17	0.049
max_div_20_3.txt	3	20	5	92.8298	3 8 13 14 17	0.09
–						
max_div_30_2.txt	2	30	2	11.6571	9 28	0.084
max_div_30_2.txt	2	30	3	28.9443	2 9 28	0.075
max_div_30_2.txt	2	30	4	52.7712	2 9 11 28	0.073
max_div_30_2.txt	2	30	5	80.9102	2 9 11 13 28	0.07
–						
max_div_30_3.txt	3	30	2	12.6137	6 17	0.095
max_div_30_3.txt	3	30	3	34.2905	6 17 24	0.091
max_div_30_3.txt	3	30	4	63.702	6 14 17 24	0.087
max_div_30_3.txt	3	30	5	99.592	6 14 15 17 24	0.095

Local Search con Greedy Constructivo:

Local Search greedy constructivo						
Problema	n	K	m	Distancia	S	CPU (ms)
max_div_15_2.txt	2	15	2	11.8592	9 7	0.004
max_div_15_2.txt	2	15	3	27.3727	9 7 1	0.013
max_div_15_2.txt	2	15	4	48.4139	9 7 4 11	0.009
max_div_15_2.txt	2	15	5	76.6535	9 7 1 11 2	0.039
–						
max_div_15_3.txt	3	15	2	13.2732	12 9	0.003
max_div_15_3.txt	3	15	3	30.3241	12 9 5	0.006
max_div_15_3.txt	3	15	4	59.7638	12 9 5 11	0.014
max_div_15_3.txt	3	15	5	96.0858	12 9 5 4 14	0.061
–						
max_div_20_2.txt	2	20	2	8.51033	18 19	0.003
max_div_20_2.txt	2	20	3	21.9961	18 19 9	0.006
max_div_20_2.txt	2	20	4	40.0023	2 19 9 3	0.023
max_div_20_2.txt	2	20	5	62.8729	18 19 9 3 2	0.049
–						
max_div_20_3.txt	3	20	2	11.8003	13 14	0.003
max_div_20_3.txt	3	20	3	30.8727	13 14 8	0.007
max_div_20_3.txt	3	20	4	56.5347	13 14 8 3	0.011
max_div_20_3.txt	3	20	5	92.8298	13 14 8 3 17	0.023
–						
max_div_30_2.txt	2	30	2	11.6571	9 28	0.003
max_div_30_2.txt	2	30	3	28.9443	9 28 2	0.005
max_div_30_2.txt	2	30	4	52.7712	9 28 2 11	0.01
max_div_30_2.txt	2	30	5	80.9102	9 28 2 11 13	0.018
–						
max_div_30_3.txt	3	30	2	13.0737	17 7	0.003
max_div_30_3.txt	3	30	3	33.8423	17 7 24	0.007
max_div_30_3.txt	3	30	4	63.5184	17 7 24 14	0.014
max_div_30_3.txt	3	30	5	99.5088	17 7 24 14 15	0.028

Local Search con Greedy Destructivo:

Local Search greedy destructivo						
Problema	n	K	m	Distancia	S	CPU (ms)
max_div_15_2.txt	2	15	2	11.8592	7 9	0.003
max_div_15_2.txt	2	15	3	27.3727	1 7 9	0.01
max_div_15_2.txt	2	15	4	48.4139	4 7 9 11	0.261
max_div_15_2.txt	2	15	5	76.6535	2 1 7 9 11	0.039
–						
max_div_15_3.txt	3	15	2	13.2732	9 12	0.004
max_div_15_3.txt	3	15	3	30.3241	5 9 12	0.009
max_div_15_3.txt	3	15	4	59.2779	5 4 12 14	0.035
max_div_15_3.txt	3	15	5	96.0858	4 5 9 12 14	0.025
–						
max_div_20_2.txt	2	20	2	8.51033	18 19	0.003
max_div_20_2.txt	2	20	3	21.9961	9 18 19	0.006
max_div_20_2.txt	2	20	4	40.0023	3 9 2 19	0.023
max_div_20_2.txt	2	20	5	62.8729	3 9 2 18 19	0.049
–						
max_div_20_3.txt	3	20	2	11.5909	3 17	0.004
max_div_20_3.txt	3	20	3	29.1574	3 13 17	0.005
max_div_20_3.txt	3	20	4	56.6903	3 13 14 17	0.011
max_div_20_3.txt	3	20	5	92.8298	3 8 13 14 17	0.032
–						
max_div_30_2.txt	2	30	2	11.6571	9 28	0.004
max_div_30_2.txt	2	30	3	28.9443	2 9 28	0.006
max_div_30_2.txt	2	30	4	52.7712	2 9 11 28	0.01
max_div_30_2.txt	2	30	5	80.9102	2 9 11 13 28	0.019
–						
max_div_30_3.txt	3	30	2	12.6137	6 17	0.004
max_div_30_3.txt	3	30	3	34.2905	6 17 24	0.01
max_div_30_3.txt	3	30	4	63.702	6 14 17 24	0.014
max_div_30_3.txt	3	30	5	99.592	6 14 15 17 24	0.031