



UNIVERSIDAD DE GRANADA

GRUPO 1

Metaheurísticas — Práctica 3. Búsquedas por Trayectorias para el Problema del Agrupamiento con Restricciones

Juan Mota Martínez
juanmotam@correo.ugr.es

25 de mayo de 2020

Índice general

1. Correcciones y representación de los datos	2
1.1. Corrección de la búsqueda local	2
1.2. Representación de los datos	3
2. Práctica 3	4
2.1. función objetivo	4
2.2. Generación de soluciones aleatorias	5
3. Metaheurísticas a implementar	6
3.1. Enfriamiento Simulado	6
3.2. Búsqueda Multiarranque Básica	7
3.3. Búsqueda Local Reiterada	7
3.3.1. Operador de Mutación	8
3.4. Hibridación ILS-ES	8
4. Resultados	10

Capítulo 1:

Correcciones y representación de los datos

En esta sección se recogen las diferencias en el almacenamiento de los datos con respecto a la práctica anterior, además de las correcciones realizadas al algoritmo búsqueda local que a diferencia de las prácticas anteriores ahora si funciona correctamente.

§1.1: Corrección de la búsqueda local

La búsqueda local anterior presentaba buenas soluciones en todos los conjuntos de datos salvo en *ecoli*, esto se debía a que no se estaban recorriendo aleatoriamente los datos de forma correcta, de manera que para un conjunto de datos tan extenso como *ecoli*, en 100000 evaluaciones sólo podía mejorar los primeros datos, esto ya se ha corregido y como se puede ver en las tablas, se proporcionan los resultados esperados.

El algoritmo parte de una asignación aleatoria, a la que hemos aplicado un operador de reparación para evitar que se queden cluster vacíos. Para reducir el tiempo de ejecución en BL se calculan parcialmente las infactibilidades, empujando la función que aparece en secciones posteriores. La condición de parada para búsqueda local es o bien se cumple el número máximo de evaluaciones o se ha recorrido todo el vecindario y no se ha producido ningún cambio en la solución, en cualquiera de los casos devolvemos la asignación obtenida.

```
1 INICIO(1)
2   Mientras no se cumpla la condición de parada continuamos:
3   INICIO(2)
4     Desordenamos aleatoriamente datos
5     Para cada d en datos:
6     INICIO(3)
7       Para cada c en cluster:
8         //Asignamos d a c y comprobamos si la valoración mejora en este caso
9         conservamos el cambio y repetimos desde INICIO(2).
10        //En caso contrario, deshacemos la asignación y continuamos el bucle.
11      FIN(3)
12    FIN(2)
13  //Se ha recorrido todo el vecindario sin mejor o se han alcanzado las evaluaciones má
14  ximas luego paramos
15 FIN(1)
16 \begin{lstlisting}
17 \section{Corrección del almacenamiento de restricciones}
18 Los tiempos desmesurados de Ecoli y algunos otros conjuntos de datos y restricciones se
19 debían a que si bien se ignoraban las restricciones de tipo 0, al leerse de una matriz no
20 se tenía en cuenta que dichas restricciones podían aparecer repetidas, resultando en que
  la restricción 2-1 y 1-2, se considereasen como dos distintas, duplicando las
  infactibilidades, lo cuál provocaba que para conjunto grande como Ecoli en el cuál es dif
  ícil de encontrar una solución de infactibilidad 0, mostrase un error desmedido. Se ha
  corregido este error con el siguiente pseudocódigo para almacenar las restricciones en
  una lista con la forma de: dato1 dato2 restricción.
\begin{lstlisting}
INICIO(1)
```

```
21  val1 = 0;
22  val2 = 0;
23  AbrirFichero(fich_restricciones)
24  Para toda linea en (fich_restricciones) mientras val1 < num_datos:
25  INICIO(2)
26      Para toda restriccion en linea:
27      INICIO(3)
28          Si val2 < num_datos y restriccion != 0:
29              //Guardamos la restricción en la lista de tuplas de la siguiente forma: val1
                val2 restriccion
30          val2++
31      FIN(3)
32  val1++
33  FIN(2)
34  FIN(1)
```

§1.2: Representación de los datos

Al igual que la práctica anterior las soluciones se representan como un tipo de dato denominado población, consiste básicamente en un vector de n posiciones para representar cada dato, en el cuál se almacena el cluster al que pertenece, es decir, para el indicar que el dato 15 está en el cluster 4 almacenamos un 4 en la posición 15, para indicar que un dato aún no ha sido asignado se representa con un -1.

Como en esta práctica no se trabaja con poblaciones, la clase PAR, presente en la P2 no es necesaria aunque se ha mantenido el código debido a que borrarlo puede provocar fallos.

Para finalizar esta sección, se recuerda que como en prácticas anteriores todos los cálculos de las distancias se han realizado sin empelar raíces cuadradas y simplemente tomando la distancia media entre los centros y el dato a asignar.

Capítulo 2:

Práctica 3

Esta sección recoge el problema resolver en esta práctica, se nos presentan 4 conjuntos de datos cada uno con un 10 y 20 % de restricciones respectivamente, se nos pide realizar una asignación de los datos en distintos clusters de forma que el valor de la función objetivo de la solución sea lo más bajo posible. Los conjuntos con los que se va a trabajar y son los siguientes:

- Iris, 3 clusters
- Ecoli, 8 clusters
- Rand, 3 clusters
- Newthyroid, 3 clusters

§2.1: función objetivo

Se trata de una forma de asignar un valor a una población concreta se calcula mediante la siguiente fórmula:

$$f(x) = \overline{C} + (infactividad \times \lambda)$$

Dónde lambda es igual al cociente entre el número de datos y el número de restricciones del problema, y \overline{C} es la desviación general de la solución. La infactividad como en las prácticas anteriores es el número de restricciones incumplidas en la solución, y se puede calcular de dos formas distintas, una total y una parcial, la total consiste en recorrer todas las restricciones y comprobar si se cumplen, mientras que a la parcial se le suministra un dato y un cluster al que asignar esta indica el error que generaría dicha asignación, es necesario indicar que los datos no asignados a un cluster no se tienen en cuenta en el cálculo de restricciones. Los pseudocódigos para cada función son los siguientes:

Calculo de Restricciones Total

```
1 INICIO(1)
2   //las tuplas siguen la siguiente forma tupla(0) = dato1 tupla(1) = dato2 tupla(2) =
   restriccion
3   errores = 0
4   Para cada tupla en el conjunto de restricciones:
5   //Asegurándonos que tupla(0) > tupla(1) para no contar dos veces la misma restricción.
6   INICIO(2)
7       cluster1 = calculaCluster(tupla(0))
8       cluster2 = calculaCluster(tupla(1))
9       Si cluster1 == cluster2 y tupla(2) == -1:
10          errores++
11       Si cluster1 != cluster2 y tupla(2) == 1:
12          errores++
13   FIN(2)
14   return errores
15 FIN(1)
```

Calculo de Restricciones Parcial

```

1 INICIO(1)
2   errores = 0
3   Para cada tupla en el conjunto de restricciones:
4   //Comprobamos que el dato suministrado sea igual a la tupla 0 o 1 y actuamos en
   consecuencia
5   INICIO(2)
6     cluster1 = calculaCluster(tupla(0))
7     cluster2 = calculaCluster(tupla(1))
8     Si cluster1 == cluster2 y tupla(2) == -1:
9       errores++
10    Si cluster1 != cluster2 y tupla(2) == 1:
11      errores++
12  FIN(2)
13  return errores
14 FIN(1)

```

§2.2: Generación de soluciones aleatorias

Para BMB e ILS generamos soluciones aleatorias para favorecer la exploración, la función para obtener dichas soluciones es la misma para ambos algoritmos, recorremos los datos del vector y para cada índice asignamos un número aleatorio entre 0 y el máximo número de clusters que hay en el problema menos 1, para evitar que se de un cluster vacío, tras generar la solución aplicamos un algoritmo de reparación, que revisa si algunos de los clusters tiene tamaño 0 y en ese caso cambia aleatoriamente un dato al cluster vacío.

Los pseudocódigos para ambas funciones son los siguientes:

Generación aleatoria de soluciones

```

1 INICIO(1)
2   Para cada d en datos:
3   INICIO(2)
4     d = NumeroAleatorio $ \% $ num$_clusters
5   FIN(2)
6 FIN(1)

```

Algoritmo Reparador

```

1 INICIO(1)
2   Para cada c en numero$_cluster
3   INICIO(2)
4     Si tamaño de c == 0:
5       //Generamos un dato aleatorio y lo asignamos a c
6   FIN(2)
7 FIN(1)

```

Capítulo 3:

Metaheurísticas a implementar

§3.1: Enfriamiento Simulado

Este algoritmo como los anteriores optimiza la solución tomando siempre aquellos cambios que impliquen una mejora en la valoración de la población a devolver, sin embargo existe una componente de exploración en este algoritmo que se implementa mediante la posibilidad de que se acepte una solución con una valoración peor si se da la condición necesaria. Para aceptar una solución peor se genera un número aleatorio y este se compara con el resultado de un fórmula que emplea una variable denominada temperatura, una explicación mas detallada aparece en las siguientes secciones. El algoritmo finaliza siempre que se cumpla el número máximo de evaluaciones (cada generación de vecino se considera una evaluación), el número de éxitos sea igual a 0 o la temperatura se reduzca hasta el mínimo. Este algoritmo presenta también el concepto de iteración, se considera una iteración todos los cambios realizados sobre la solución inicial y dicha iteración termina cuando se haya generado el número máximo de vecinos o hayan sido aceptados un número de vecinos prefijado.

$$T_0 = \frac{\mu * C(S_0)}{-\ln(\phi)}$$

Dónde μ y ϕ valen 0.3 ambas y $C(S_0)$ es la valoración de la solución inicial que se ha generado aleatoriamente. A lo largo de la ejecución vamos disminuyendo la temperatura hasta un tope de 0.001. Para la disminución se emplea la siguiente fórmula:

$$T_{k+1} = \frac{T_k}{1 + \beta \times T_k}$$

$$\beta = \frac{T_0 - T_f}{M \times T_0 \times T_f}$$

La temperatura inicial se representa como una constante que se calcula mediante una función que replica la fórmula mencionada anteriormente. Mientras que el enfriamiento se replica recalculando la temperatura usando la segunda fórmula al final de cada iteración.

La explotación de la solución se realiza generando un dato y un clúster aleatorios y asignando dicho dato al clúster, tras esto comprobamos si la solución mejora o no, siempre que obtengamos una solución mejor la aceptaremos y seguiremos explorando y explotando sus vecinos. Sin embargo, como se ha mencionado antes se puede aceptar una solución peor si al generar un número aleatorio comprendido entre 0 y 1 este resulta ser menor que:

$$-1 * \exp(val * temperatura)$$

Dónde val es el resultado de restar la valoración de la nueva solución a la valoración inicial.

Como se puede observar cuanto mayor sea la temperatura más posibilidad hay de que se acepte una solución poco favorable. Se parte de una temperatura inicial dada por la fórmula:

Enfriamiento Simulado

```

1 INICIO(1)
2   sol = GeneradorAleatoriodeSoluciones()
3   temp_ini = CalculaTemperatura(sol)
4   evaluaciones = 0
5   vecinos_aceptados = 1
6   Mientras que temp_ini > tem_fin && evaluaciones <= 100000 && vecinos_aceptados > 0):
7     INICIO(2)
8       vecinos_aceptados = 0
9       vecinos_explorados = 0
10      Mientras que vecinos_aceptados < max_aceptados && vecinos_explorados <
          max_generados:
11        INICIO(3)
12          sol_nueva = ReasignacionAleatoria(sol)
13          //generamos un numero U entre 0 y 1
14          Si Valoracion(sol_nueva) < Valoracion(sol) || U < (-1*e^(diferencia*
          temperatura_ini)):
15            sol = sol_nueva
16            vecinos_aceptados++
17            evaluaciones++
18            vecinos_explorados++
19          FIN(3)
20        FIN(2)
21      FIN(1)

```

§3.2: Búsqueda Multiarranque Básica

Probablemente el algoritmo más simple de esta práctica, BMB simplemente genera 10 poblaciones aleatorias sobre las que se aplica un optimización mediante búsqueda local, tras esto se comparan y se devuelve la mejor solución. El algoritmo de búsqueda local que empleado y el uso para el resto de metaheurísticas es el mismo que se ha descrito al principio del documento.

Búsqueda Local Multiarranque

```

1 INICIO(1)
2   Repetimos 10:
3     mejor_solucion = GeneradorDeSolucionesAleatorias()
4     INICIO(2)
5       nueva_sol = GeneradorDeSolucionesAleatorias()
6       //BL sobre nueva_sol
7       Si Valoracion(mejor_solucion) > Valoracion(nueva_sol):
8         mejor_solucion = nueva_sol
9     FIN(2)
10  FIN(1)

```

§3.3: Búsqueda Local Reiterada

Partiendo de una solución aleatoria optimizada con búsqueda local, generamos una segunda solución aleatoria optimizada de la misma forma, tras esto comparamos ambas y sobre la mejor de ellas aplicamos un algoritmo de mutación por segmento, este algoritmo selecciona un 10% de la población y las asigna aleatoriamente a clusters nuevos, siempre aplicando un operador de reparación tras esto para no dejar clusters vacíos. Una vez obtenida la solución mutada volvemos a aplicarle BL y repetimos el proceso, hasta 10 veces.

Búsqueda Local Reiterada


```

1 INICIO(1)
2   sol1 = GeneradorDeSolucionesAleatorias()
3   sol2 = GeneradorDeSolucionesAleatorias()
4   //Aplicamos BL sobre ambas
5   Repetimos este bucle 10 veces:
6   INICIO(2)
7     Si Valoracion(sol1) <= Valoracion(sol2):
8       Mutar(sol1)
9       //BL sobre sol1
10    Si Valoracion(sol1) > Valoracion(sol2):
11      Mutar(sol2)
12      //BL sobre sol2
13  FIN(2)
14  //Devolvemos aquella que tenga mejor valoración
15 FIN(1)

```

§3.3.1: Operador de Mutación

El operador de mutación es una función que genera un número aleatorio entre 0 y el máximo número de datos del problema, este número será el inicio de nuestro segmento. Tras esto se comprueba si es posible generar un fin de segmento válido sumándole al inicio el 10 % de la población, es decir partiendo de un inicio = 5 para una población de 100 datos el fin correspondiente sería 14. Decimos fin de segmento válido puesto que si generamos un inicio en 99 el fin de segmento quedaría fuera del vector. Esto se soluciona generando un nuevo inicio de segmento aleatorio hasta que se puede obtener un fin de segmento válido.

Inicialmente se planteó la opción de que se continuase por el principio del vector si se generaba un inicio cercano a los límites, de forma que para un inicio en 99, el fin estaría en 7. Esto funcionaba correctamente y proporcionaba buenos resultados, pero la opción descrita anteriormente genera soluciones casi idénticas y tiene una implementación mucho más simple de forma que se conservó el cambio.

Operador de Mutación

```

1 INICIO(1)
2   ini = generadorAleatorio()
3   fin = ini
4   Mientras (fin - ini) != 10% de la población:
5   INICIO(2)
6     ini = generadorAleatorio()
7     fin = ini + 10% poblacion mod num_datos
8   FIN(2)
9 FIN(1)

```

§3.4: Hibridación ILS-ES

Idéntico a ILS pero en este caso en lugar de emplear búsqueda local para refinar las soluciones emplearemos el algoritmo de enfriamiento simulado descrito anteriormente, con dos cambios, en primer lugar se le suministra una solución a mejorar, no genera una propia, y el número máximo de evaluaciones es 10000. El pseudocódigo es el mismo que el de búsqueda local reiterada, pero en lugar de llamar a BL hacemos una llamada a ES.

ILS-ES

```

1 INICIO(1)
2   sol1 = GeneradorDeSolucionesAleatorias()
3   sol2 = GeneradorDeSolucionesAleatorias()
4   //Aplicamos ES sobre ambas
5   Repetimos este bucle 10 veces:

```

```
6  INICIO(2)
7      Si Valoracion(sol1) <= Valoracion(sol2):
8          Mutar(sol1)
9          //ES sobre sol1
10     Si Valoracion(sol1) > Valoracion(sol2):
11         Mutar(sol2)
12         //ES sobre sol2
13     FIN(2)
14     //Devolvemos aquella que tenga mejor valoración
15 FIN(1)
```

Capítulo 4:

Resultados

Se han tomado los resultados de la tablas anteriores para greedy y BL, pero se han repetido los cálculo de BL en Ecoli, para ambos conjuntos de restricciones.

Greedy 10

	Iris				Ecoli			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>
1345378882	0,0422314	0	0,0422314	0,226019	4,23911	26	7,17959	9,69278
572718921	0,042503	0	0,042503	0,125442	2,40	46,00	7,60	5,86
47389829	0,06	0,00	0,06	0,13	2,53	70,00	10,44	5,11
28388992	0,04	6,00	0,16	0,12	1,58	259,00	30,87	6,82
348427479	0,05	0,00	0,05	0,16	1,82	199,00	24,32	8,20
Media	0,05	1,20	0,07	0,15	2,51	120,00	16,08	7,14

	Rand				Newthyroid			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>
1345378882	0,915755	0	0,915755	0,124875	0,0592128	0	0,0592128	0,241551
572718921	2,35	0,00	2,35	0,12	1,50	66,00	7,94	0,41
47389829	2,02	0,00	2,02	0,12	1,73	2,00	1,92	0,43
28388992	1,07	0,00	1,07	0,12	1,35	0,00	1,35	0,85
348427479	0,81	0,00	0,81	0,16	2,18	84,00	10,38	0,40
Media	1,43	0,00	1,43	0,13	1,36	30,40	4,33	0,47

Greedy 20

	Iris				Ecoli			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>
1345378882	0,0592128	0	0,0592128	0,241551	3,64198	32	7,26103	8,89965
572718921	0,04	0,00	0,04	0,23	2,58	18,00	4,61	12,04
47389829	0,08	0,00	0,08	0,24	5,94	0,00	5,94	6,39
28388992	0,05	0,00	0,05	0,24	4,37	16,00	6,18	22,31
348427479	0,04	0,00	0,04	0,23	8,18	3,00	8,52	11,95
Media	0,05	0,00	0,05	0,24	4,94	13,80	6,50	12,32

	Rand				Newthyroid			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>
1345378882	0,915755	0	0,915755	0,245616	1,58951	0	1,58951	0,827824
572718921	2,35	0,00	2,35	0,23	1,53	0,00	1,53	0,79
47389829	2,02	0,00	2,02	0,22	1,71	0,00	1,71	0,60
28388992	4,37	16,00	6,18	22,31	1,49	0,00	1,49	0,60
348427479	0,81	0,00	0,81	0,23	2,13	0,00	2,13	0,76
Media	2,10	3,20	2,46	4,65	1,69	0,00	1,69	0,72

BL 10

	Iris				Ecoli			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>
1345378882	0,03	0,00	0,03	2,96	0,771961	25	3,60	16,85
572718921	0,03	0,00	0,03	3,90	0,779759	11	2,02381	16,1063
47389829	0,03	0,00	0,03	3,87	0,966551	7	1,75822	12,9155
28388992	0,03	0,00	0,03	3,94	0,553967	105	12,429	16,2201
348427479	0,03	0,00	0,03	4,25	0,758872	29	4,03863	16,7789
Media	0,03	0,00	0,03	3,78	0,77	35,40	4,77	15,77

	Rand				Newthyroid			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>
1345378882	1,71	0,00	1,71	3,60	0,26	0,00	0,26	12,39
572718921	1,71	0,00	1,71	3,63	0,26	0,00	0,26	13,72
47389829	1,71	0,00	1,71	4,36	0,26	0,00	0,26	15,26
28388992	1,71	0,00	1,71	4,27	0,26	0,00	0,26	14,83
348427479	1,71	0,00	1,71	4,03	0,26	0,00	0,26	11,46
Media	1,71	0,00	1,71	3,98	0,26	0,00	0,26	13,53

BL 20

	Iris				Ecoli			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>
1345378882	0,027954	0	0,027954	4,731	1,47429	1	1,59	30,17
572718921	0,03	0,00	0,03	5,62	1,41	1,00	1,53	21,65
47389829	0,03	0,00	0,03	6,30	1,41	1,00	1,53	24,97
28388992	0,027954	0	0,027954	6,28292	1,16	12,00	2,51	19,89
348427479	0,03	0,00	0,03	5,49	1,19389	6	1,87	25,68
Media	0,03	0,00	0,03	5,68	1,330272	4,2	1,80527	24,47342

	Rand				Newthyroid			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>
1345378882	1,71	0,00	1,71	6,54	0,26	0,00	0,26	22,46
572718921	1,71	0,00	1,71	5,80	0,26	0,00	0,26	18,53
47389829	1,71	0,00	1,71	6,39	0,26	0,00	0,26	20,96
28388992	1,71	0,00	1,71	7,31	0,26	0,00	0,26	22,45
348427479	1,71	0,00	1,71	7,00	0,26	0,00	0,26	21,96
Media	1,71	0,00	1,71	6,61	0,26	0,00	0,26	21,27

BMB 10

	Iris				Ecoli			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>
1345378882,00	0,02	0,00	0,02	3,12	0,87	8,00	1,77	153,76
572718921,00	0,02	0,00	0,02	3,09	1,06	4,00	1,52	140,85
34737829,00	0,02	0,00	0,02	3,45	1,03	5,00	1,59	144,28
28388992,00	0,02	0,00	0,02	3,35	0,83	8,00	1,73	157,75
9987479,00	0,02	0,00	0,02	3,34	0,94	5,00	1,50	139,02
Media	0,02	0,00	0,02	3,27	0,94	6,00	1,62	147,13

	Rand				Newthyroid			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>
1345378882,00	1,71	0,00	1,71	2,98	0,16	0,00	0,16	9,36
572718921,00	1,71	0,00	1,71	3,49	0,16	0,00	0,16	9,39
34737829,00	1,71	0,00	1,71	3,58	0,16	0,00	0,16	10,45
28388992,00	1,71	0,00	1,71	3,41	0,16	0,00	0,16	10,11
9987479,00	1,71	0,00	1,71	3,32	0,16	0,00	0,16	9,28
Media	1,71	0,00	1,71	3,36	0,16	0,00	0,16	9,72

BMB 20

	Iris				Ecoli			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>
1345378882	0,0237325	0	0,0237325	4,67156	1,3508	1	1,4639	238,744
572718921	0,0237325	0,00	0,0237325	4,28865	1,52238	0	1,52238	212,758
34737829	0,0237325	0,00	0,0237325	4,81331	1,3508	1,00	1,4639	207,362
28388992	0,0237325	0,00	0,0237325	4,61871	1,456	1	1,56909	235,684
9987479	0,0237325	0,00	0,0237325	4,42578	1,42264	1,00	1,53574	199,802
Media	0,02	0,00	0,02	4,56	1,42	0,80	1,51	218,87

	Rand				Newthyroid			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>
1345378882	1,70905	0	1,70905	5,01234	0,157404	0,00	0,157404	13,5821
572718921	1,70905	0,00	1,70905	4,42057	0,157404	0,00	0,157404	12,9212
34737829	1,70905	0,00	1,70905	4,60553	0,157404	0,00	0,157404	12,4066
28388992	1,70905	0,00	1,70905	4,64705	0,157404	0,00	0,157404	13,3659
9987479	1,70905	0,00	1,70905	4,73786	0,157404	0,00	0,157404	13,0078
Media	1,71	0,00	1,71	4,68	0,16	0,00	0,16	13,06

ILS-ES 10

	Iris				Ecoli			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	T	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	T
1345378882	0,0237326	0	0,0237326	6,08214	0,459069	153	17,76	121,96
572718921	0,02	0,00	0,02	5,65	0,63	55,00	6,85	158,45
34737829	0,02	0,00	0,02	5,09	0,75	22,00	3,24	138,34
28388992	0,02	0,00	0,02	5,94	0,69	23,00	3,29	150,31
9987479	0,0237325	0	0,0237325	4,51	0,76	24,00	3,47	133,77
Media	0,02	0,00	0,02	5,45	0,66	55,40	6,92	140,57

	Rand				Newthyroid			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	T	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	T
1345378882	1,70	0,00	1,70	4,32	0,157404	0	0,157404	21,0716
572718921	1,71	0,00	1,71	4,97	0,16	0,00	0,16	16,00
34737829	1,69071	0,00	1,69071	1,56778	0,16	0,00	0,16	21,96
28388992	1,71	0,00	1,71	5,70	0,16	0,00	0,16	18,85
9987479	1,71	0,00	1,71	5,18	0,16	0,00	0,16	17,26
Media	1,70	0,00	1,70	4,35	0,16	0,00	0,16	19,03

ILS-ES 20

	Iris				Ecoli			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	T	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	T
1345378882	0,0234608	0	0,0234608	10,4991	1,3042	5	1,86968	255,009
572718921	0,0237326	0,00	0,0237326	7,81767	1,43402	3	1,77331	178,74
34737829	0,0236993	0,00	0,0236993	6,9232	0,864754	41,00	5,50166	244,627
28388992	0,0234998	0	0,0234998	5,8052	1,456	1,00	1,57	231,67
9987479	0,0237009	0,00	0,0237009	9,78469	1,19009	11	2,43414	201,157
Media	0,02	0,00	0,02	8,17	1,25	12,20	2,63	222,24

	Rand				Newthyroid			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	T	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	T
1345378882	1,70905	0	1,70905	6,23267	0,157404	0,00	0,157404	25,455
572718921	1,70905	0,00	1,70905	6,98524	0,157404	0,00	0,157404	30,3852
34737829	1,70582	0,00	1,70582	7,40383	0,157404	0,00	0,157404	23,2891
28388992	1,70905	0,00	1,70905	6,74948	0,157404	0,00	0,157404	26,87
9987479	1,70177	0,00	1,70177	6,03555	0,157404	0,00	0,157404	38,0161
Media	1,71	0,00	1,71	6,68	0,16	0,00	0,16	28,80

ILS 10

	Iris				Ecoli			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>
1345378882	0,0235079	0	0,0235079	1,59364	1,17567	13	2,64591	95,3898
572718921	0,02	0,00	0,02	1,67	1,36199	6	2,04057	93,3731
34737829	0,02	0,00	0,02	1,70	1,60	3,00	1,94	100,18
28388992	0,02	0,00	0,02	1,59	1,33	24,00	4,04	74,95
9987479	0,02	0,00	0,02	1,74	1,10608	12	2,46323	102,761
Media	0,02	0,00	0,02	1,66	1,31	11,60	2,63	93,33

	Rand				Newthyroid			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>
1345378882	1,70	0,00	1,70	1,69	0,16	0,00	0,16	4,26
572718921	1,69	0,00	1,69	1,67	0,157404	0	0,157404	4,9857
34737829	1,69071	0,00	1,69071	1,56778	0,16	0,00	0,16	5,14
28388992	1,70177	0,00	1,70177	1,73085	0,16	0,00	0,16	4,54
9987479	1,70	0,00	1,70	1,91	0,157404	0	0,157404	5,23
Media	1,70	0,00	1,70	1,71	0,16	0,00	0,16	4,83

ILS 20

	Iris				Ecoli			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>
1345378882	0,0236744	0	0,0236744	2,92155	1,52049	1	1,63358	155,993
572718921	0,0235007	0,00	0,0235007	2,62116	1,26722	4	1,7196	141,266
34737829	0,023171	0	0,023171	2,71806	1,45017	1	1,56327	154,423
28388992	0,0233402	0	0,0233402	3,16456	1,84032	17	3,76294	130,6
9987479	0,0233198	0	0,0233198	2,55679	1,67606	0	1,67606	155,479
Media	0,02	0,00	0,02	2,80	1,55	4,60	2,07	147,55

	Rand				Newthyroid			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>
1345378882	1,70183	0	1,70183	2,53817	0,157404	0,00	0,157404	7,99114
572718921	1,70861	0,00	1,70861	2,73296	0,157404	0,00	0,157404	6,90486
34737829	1,68557	0	1,68557	2,76344	0,157404	0	0,157404	8,46435
28388992	1,7083	0	1,7083	2,69249	0,157404	0	0,157404	8,42668
9987479	1,70813	0	1,70813	2,75022	0,157404	0	0,157404	6,91738
Media	1,70	0,00	1,70	2,70	0,16	0,00	0,16	7,74

ES 10

	Iris				Ecoli			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	T	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	T
1345378882	0,0237325	0	0,0237325	4,30109	0,851918	12	2,20906	153,094
572718921	0,0237325	0,00	0,0237325	4,8422	0,617877	54,00	6,72502	134,152
34737829	0,0240005	0,00	0,0240005	3,29659	0,435763	475	54,156	117,127
28388992	0,0237325	0,00	0,0237325	4,39254	0,81444	31	4,32039	163,185
9987479	0,0237325	0,00	0,0237325	4,39479	0,645614	57,00	7,09204	134,113
Media	0,02	0,00	0,02	4,25	0,67	125,80	14,90	140,33

	Rand				Newthyroid			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	T	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	T
1345378882	1,70276	0	1,70276	3,87693	0,157404	0,00	0,157404	11,8942
572718921	1,70905	0,00	1,70905	4,29254	0,157404	0,00	0,157404	15,7016
34737829	1,70905	0,00	1,70905	4,38224	0,157404	0,00	0,157404	14,5983
28388992	1,70905	0	1,70905	3,33105	0,157404	0,00	0,157404	16,124
9987479	1,70905	0,00	1,70905	3,67937	0,157404	0,00	0,157404	14,8164
Media	1,71	0,00	1,71	3,91	0,16	0,00	0,16	14,63

ES 20

	Iris				Ecoli			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	T	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	T
1345378882	0,0237325	0	0,0237325	6,61007	1,17213	12	2,52927	220,554
572718921	0,0237325	0,00	0,0237325	4,78107	1,20354	4	1,65592	192,207
34737829	0,0237325	0,00	0,0237325	7,6411	1,456	1	1,56909	229,05
28388992	0,0237325	0,00	0,0237325	6,13542	1,51935	1	1,63245	237,188
9987479	0,0237325	0,00	0,0237325	6,06146	1,53498	0	1,53498	206,805
Media	0,02	0,00	0,02	6,25	1,38	3,60	1,78	217,16

	Rand				Newthyroid			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	T	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	T
1345378882	1,70905	0	1,70905	5,58688	0,157404	0	0,157404	17,4444
572718921	1,70905	0,00	1,70905	5,36618	0,157404	0,00	0,157404	20,9386
34737829	1,70905	0,00	1,70905	6,06239	0,157404	0,00	0,157404	23,4988
28388992	1,70905	0	1,70905	5,68096	0,157404	0,00	0,157404	23,3893
9987479	1,70905	0,00	1,70905	5,43299	0,157404	0,00	0,157404	25,3209
Media	1,71	0,00	1,71	5,63	0,16	0,00	0,16	22,12

Resultados 10

	Iris				Ecoli			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>
Greedy	0,07	0	0,07	0,15	2,51	120	16,08	7,14
BL	0,03	0	0,03	3,78	0,77	35,4	4,77	15,77
BMB	0,02	0	0,02	3,27	0,94	6	1,62	147,13
ILS-ES	0,02	0	0,02	5,45	0,66	55,4	6,92	140,57
ILS	0,02	0	0,02	1,66	1,31	11,6	2,63	93,33
ES	0,02	0	0,02	4,25	0,67	125,8	14,9	140,33

	Rand				Newthyroid			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>
Greedy	1,43	0	1,43	0,13	1,36	30	4,33	0,47
BL	1,71	0	1,71	3,98	26	0	0,26	13,53
BMB	1,71	0	1,71	3,36	0,16	0	0,16	9,72
ILS-ES	1,7	0	1,7	4,35	0,16	0	0,16	19,03
ILS	1,7	0	1,7	1,71	0,16	0	0,16	4,83
ES	1,71	0	1,71	3,91	0,16	0	0,16	14,63

Resultados 20

	Iris				Ecoli			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>
Greedy	0,05	0	0,05	0,24	4,94	13,8	6,5	12,32
BL	0,03	0	0,03	5,68	1,330272	4,2	1,80527	24,47342
BMB	0,02	0	0,02	4,56	1,42	0,8	1,51	218,87
ILS-ES	0,02	0	0,02	8,17	1,25	12,2	2,63	222,24
ILS	0,02	0	0,02	2,8	1,55	4,6	2,07	147,55
ES	0,02	0	0,02	6,25	1,38	3,6	1,78	217,16

	Rand				Newthyroid			
	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>	<i>Tasa_C</i>	<i>Tasa_inf</i>	<i>Agr,</i>	<i>T</i>
Greedy	2,1	3,2	2,46	4,65	1,69	0	1,69	0,72
BL	1,71	0	1,71	6,61	0,26	0	0,26	21,27
BMB	1,71	0	1,71	4,68	0,16	0	0,16	13,06
ILS-ES	1,71	0	1,71	6,68	0,16	0	0,16	28,8
ILS	1,7	0	1,7	2,7	0,16	0	0,16	7,74
ES	1,71	0	1,71	5,63	0,16	0	0,16	22,12

Los resultado obtenidos son los esperados, para todos los conjuntos de datos exceptuando ecoli se obtienen buenos reusltados en muy poco tiempo, sin embargo la implementación de la búsqueda local demuestra estar poco optimizada ya que al tener que ejecutar diversas veces el mismo algoritmo como es el caso de BMB, los tiempos de respuesta resultan ser extremadamente altos, aunque esperados, para ecoli con 10 % de restricciones

BL devuelve una solución en 15 segundos, BMB ejecuta esta misma búsqueda 10 veces, luego el tiempo que tarda en responder es aproximadamente 150 segundos.

Con la hibridación de ILS-ES no ocurre esto, puesto que la primera ejecución del ES agota todas las evaluaciones, luego sólo se realiza ES una vez.