

Práctica 2.a:

Técnicas de Búsqueda basadas en Poblaciones para el
Problema de la Mínima Dispersión Diferencial (MDD)

Juan Antonio Martínez Sánchez
Curso 2021/22

?abstractname?

En este documento comentaremos técnicas de Búsqueda basadas en Poblaciones en la resolución del problema de la mínima dispersión (MDD) utilizando Algoritmos Genéticos y ALgoritmos Meméticos.

?contentsname?

1	Introducción	3
2	Algoritmos Genéticos	4
2.1	Aplicacion de los Algoritmos Genéticos	4
2.2	Operadores	5
2.2.1	Operador de Selección	5
2.2.2	Operadores de Cruce	6
2.2.3	Operador de Mutación	7
2.2.4	Otros operadores	7
2.3	Algoritmo	8
3	Algoritmos Meméticos	10
4	Experimentos y análisis de resultados	11

1 Introducción

El objetivo de esta práctica es estudiar el funcionamiento de las Técnicas de Búsqueda basadas en Poblaciones en la resolución del problema de la mínima dispersión (MDD) descrito en las transparencias del Seminario 2. Para ello, se requerirá que el estudiante adapte los siguientes algoritmos a dicho problema:

- Algoritmos Genéticos: Dos variantes generacionales elitistas (AGGs) y otras dos estacionarias (AGEs), descritas en el Seminario 3. Aparte del esquema de evolución, la única diferencia entre los dos modelos de AGGs y AGE será el operador de cruce empleado.
- Algoritmos Meméticos: Tres variantes de algoritmos meméticos (AMs) basadas en un AGG, descritas en el Seminario 3. La única diferencia entre las tres variantes de AMs serán los parámetros considerados para definir la aplicación de la búsqueda local.

Para diseñar los AMs, se utilizará el método de Búsqueda Local (BL) desarrollado en la Práctica 1.a.

Se utilizarán 50 casos seleccionados de varios de los conjuntos de instancias todas disponibles en la [MDPLIB](#), pertenecientes al grupo GDK-b con distancias aleatorias reales con, n entre $\{25, 50, 75, 100, 125, 150\}$, y m entre 2 y 45 (GDK-bGKD-b_1_n25_m2.txt a GKD-b_50_n150_m45.txt)

Los algoritmos de esta práctica tienen en común las siguientes componentes:

- Esquema de representación: Se seguirá la representación en forma de representación binaria; un vector binario $Sel = (x_1, \dots, x_n)$ en el que las posiciones del vector representan los elementos y su valor, 0 o 1, la no selección o selección de los mismos.
- $Minimize \quad Max_{i \in M} \{ \sum_{j \in M} d_{ij} \} - Min_{i \in M} \{ \sum_{j \in M} d_{ij} \}$
 $Subject \ to \quad M \subset N, |M| = m$
- Generación de la solución inicial o la población inicial: La(s) solución/soluciones iniciales se generarán de forma aleatoria en todos los casos.

2 Algoritmos Genéticos

2.1 Aplicación de los Algoritmos Genéticos

Para la aplicación de algoritmos genéticos se considerarán dos versiones, una basada en el esquema evolutivo generacional con elitismo (AGG) y otra basada en el esquema evolutivo estacionario (AGE). En el primero se seleccionará una población de padres del mismo tamaño que la población genética mientras que en el segundo se seleccionarán únicamente dos padres.

Para la selección de individuos se realizarán torneos binarios que consisten en elegir aleatoriamente dos individuos de la población y seleccionar el mejor de ellos. En el esquema generacional, se aplicarán tantos torneos como individuos existan en la población genética, incluyendo los individuos ganadores en la población de padres. En el esquema estacionario, se aplicará dos veces el torneo para elegir los dos padres que serán posteriormente recombinados (cruzados).

En el esquema generacional, la población de hijos sustituye automáticamente a la actual. Para conservar el elitismo, si la mejor solución de la generación anterior no sobrevive, sustituye directamente la peor solución de la nueva población. En el estacionario, los dos descendientes generados tras el cruce y la mutación (esta última aplicada según una determinada probabilidad) sustituyen a los dos peores de la población actual, en caso de ser mejores que ellos.

Para el cruce entre dos individuos tendremos en cuenta dos maneras de cruce, cruce uniforme o cruce por posición. Por tanto desarrollaremos 4 AGs distintos, dos generacionales (AGG-uniforme y AGG-posición) y dos estacionarios (AGE-uniforme y AGE-posición).

La mutación consistirá en intercambiar el valor del gen a mutar x_i por el de otro gen x_j escogido aleatoriamente con el valor contrario.

Para la realización de la práctica utilizaremos los siguientes valores:

El tamaño de la población será de 50 cromosomas. La probabilidad de cruce será 0,7 en el AGG y 1 en el AGE (siempre se cruzan los dos padres). La probabilidad de mutación (por individuo) será de 0,1 en ambos casos. El criterio de parada en las dos versiones del AG consistirá en realizar 100000 evaluaciones de la función objetivo.

2.2 Operadores

2.2.1 Operador de Selección

Tenemos uno para el AG Generacional y otro diferente para el AG Estacionario:

Operador de Selección para AGG:

```
OperadorSeleccion (poblacion , m, M, evaluaciones) {  
    while poblacion.size() < 50 do:  
        p1, p2 in range(0,49);          // Donde p1!=p2  
        mejor_padre = MejorSolucion(poblacion[p1], poblacion[p2], m, M,  
            evaluaciones);  
        nueva_poblacion.push_back(mejor_padre);  
        poblacion = nueva_poblacion;  
}
```

Este operador consiste en dada una poblacion inicial de tamaño 50, generar dos índices aleatorios en el rango [0, 49] diferentes para posteriormente escoger de padres los respectivos a esos índices dentro de la población. Una vez obtenidos los aplicaremos un torneo binarios, en este caso utilizaremos la función *MejorSolucion()* que nos devuelve el mejor y seguidamente lo insertaremos en la nueva población.

Operador de Selección para AGE:

```
OperadorSeleccion (poblacion , padres , m, M, evaluaciones) {  
    while padres.size() < 2 do:  
        p1, p2 in range(0,49);          // Donde p1!=p2  
        mejor_padre = MejorSolucion(poblacion[p1], poblacion[p2], m, M,  
            evaluaciones);  
        padres.push_back(mejor_padre);  
}
```

Este operador es igual al del modelo Generacional pero esta vez en vez de generar una nueva población, solo nos quedaremos con dos individuos los cuales obtendremos de 2 torneos binarios pasando dos padres aleatorios por *MejorSolucion()*.

Dentro de este apartado podemos distinguir en la función **MejorSolucion()** el parámetro *evaluaciones* el cual nos ayudará con el conteo de evaulaciones de la función objetivo que se aplica internamente en dicha función. Esta función consiste en dados 2 cromosomas de la población, calcule los puntos seleccionados de la solucion y asu vez generar las distancias entre los puntos almacenadas en *M* y asi poder llamar a la función objetivo *CalculaDispersion()* y así comparar las dispersiones que genera cada cromosoma y quedarnos con el mejor.

2.2.2 Operadores de Cruce

Para este operador distinguiremos dos operadores diferentes donde AGG uniforme y AGE uniforme tendrán uno común, y AGG posición y AGE posición otro en común.

Analizaremos primero el cruce uniforme:

Operador de Cruce Uniforme:

```
OperadorCruce (poblacion , n, m, M) {  
    num_a_cruzar = poblacion.size()*0,7;  
    for i in range(0,num_a_cruzar) do:  
        hijo = Cruce();  
        Reparacion(hijo);  
        nueva_poblacion.push_back(hijo);  
    while i < poblacion.size() do:  
        nueva_poblacion.push_back(poblacion[i]);  
        i++;  
    poblacion = nueva_poblacion;  
}
```

Este operador consiste en dado una probabilidad de cruce de 0,7, cruzar el 70% de la población y el resto lo mantenemos. Como podemos ver para los *num_a_cruzar* primeros cromosomas de la población realizaremos un cruce uniforme mediante la función *Cruce()*. Este modo de cruce tiene el problema de que necesita 'reparación', es decir, hay posibilidad de que después de cruzar, el hijo resultante sea una solución no válida porque no dispondrá de *m* 1s obligatorios dependientes del caso.

Por tanto, las funciones clave son *Cruce()* y *Reparacion()*.

Cruce(): Consiste en dados dos cromosomas padres, identifique los índices donde el valor de los padres coincidan y construir el hijo a partir de la intersección de los padres. El resto del hijo se construirá con una probabilidad del 50% de tener el cromosoma de un padre o del otro y así hasta completarlo.

Reparacion(): Consiste en dado un hijo previamente cruzado y un conjunto de índices que guarden las posiciones del hijo que comparte con los dos padres, compruebe si faltan o sobran 1s. Si faltan, se añadirán hasta que sea un hijo factible llamando a la función *AñadePunto()* que añade el punto más cercano a la media de distancias entre los puntos. Si sobran, se eliminarán hasta que sea un hijo factible llamando a la función *QuitaPunto()* que elimina el punto más lejano a la media de distancias entre los puntos.

Operador de Cruce por Posición:

```
OperadorCruce (poblacion , n, m, M) {  
    num_a_cruzar = poblacion.size()*0,7;  
    for i in range(0,num_a_cruzar) do:  
        Cruce(hijo1 , hijo2);  
        nueva_poblacion.push_back(hijo1 , hijo2);  
    while i < poblacion.size() do:  
        nueva_poblacion.push_back(poblacion[i]);  
        i++;  
    poblacion = nueva_poblacion;  
}
```

A diferencia con el uniforme no necesita reparación y genera dos hijos a la vez.

La función **Cruce()** esta vez consiste en recorrer los padres buscando los elementos que no comparten y guardarlos en dos vectores, uno para cada padre. Ahora esos vectores los desordenaremos y comenzaremos a crear los hijos. En la construcción del hijo, en las posiciones que coincidan los dos padres los valores se mantendrán y para las que no coincidan utilizaremos los valores restantes de cada padre y los insertaremos aleatoriamente en el hijo.

2.2.3 Operador de Mutación

Este operador es común en todos los AG de la práctica y consiste en que hay que realizar una mutación, es decir intercambiar el valor de dos posiciones diferentes de distintos valores, con probabilidad de 0,1 que es el 10% de la población.

Primero generaremos índices aleatorios diferentes que representen ese 10% aleatorio de la población. Una vez tengamos los índices recorreremos los cromosomas respectivos a esos índices de la población. Ahora generaremos dos índices aleatorios diferentes y que los valores respectivos en el cromosoma sean distintos y finalmente cambiaremos los valores si hay un 0 cambiamos a 1 y viceversa. De esta manera en ningún momento los cromosomas dejarán de ser factibles y por tanto la mutación será válida.

2.2.4 Otros operadores

CalcularSolucion():

Esta función es la encargada de generar un cromosoma o una solución aleatoria la cual consiste en generar m índices aleatorios que posteriormente en un vector de tamaño n inicializado a 0 se irán recorriendo y cambiando el valor a 1.

CalcularPoblacion():

Esta función ejecutará 50 veces *CalcularSolucion()* y así crear una población con 50 soluciones.

MejorSolucion() y PeorSolucion():

Estas funciones comparan dos soluciones generando sus vectores de distancias para llamar a la función objetivo *CalculaDispersion()* y así comprobar cual es la mejor o la peor solución y devolverla.

CalculaMejorSolucionPoblacion() y CalculaPeorSolucionPoblacion():

Estas funciones obtienen la mejor y la peor solución de la población.

CalculaDispersion():

Función objetivo que calcula la dispersion entre las distancias de un conjunto de puntos.

2.3 Algoritmo

AG Generacional:

En el modelo Generacional lo primero que haremos será calcular la mejor solución de la población actual y seguidamente pasaremos la población actual por los operadores de selección, cruce y mutación. Después calcularemos la peor solución de la nueva población y compararemos si la mejor es mejor solución que la peor actual y si se cumple esto, sustituiremos la peor por la mejor de la población anterior. Todo bajo la restricción de que no superemos las 100000 evaluaciones de la función objetivo.

```
AGG (poblacion , n, m, M) {  
    CalcularPoblacion(poblacion);  
  
    while evaluaciones < 100000 do:  
        mejor_anterior = CalculaMejorSolucionPoblacion(poblacion);  
        poblacion_siguiente = poblacion  
  
        SELECCION(poblacion);  
        CRUCE(poblacion);  
        MUTACION(poblacion);  
  
        peor_actual = CalculaPeorSolucionPoblacion(  
            poblacion_siguiente);  
  
        if mejor_anterior < peor_actual:  
            Intercambia mejor_anterior por peor_actual en poblacion  
            ;  
  
        poblacion = nueva_poblacion;  
    end;  
}
```


AG Estacionario:

En el modelo Estacionario lo primero que haremos será pasaremos la poblacion actual por el operador de selección del cual obtendremos una pareja de padres que la pasaremos por los operadores de cruce y mutación. Después calcularemos la peor solución de la población y compararemos si el hijos[0] es mejor solución que la peor y si se cumple esto, sustituiremos la peor por el hijos[0]. Lo mismo haremos para el hijos[1] Todo bajo la restricción de que no superemos las 100000 evaluaciones de la función objetivo.

```
AGE ( poblacion , n, m, M) {  
    CalcularPoblacion(poblacion);  
  
    while evaluaciones < 100000 do:  
  
        SELECCION(poblacion , hijos);  
        CRUCE(hijos);  
        MUTACION(hijos);  
  
        peor_actual = CalculaPeorSolucionPoblacion(poblacion);  
  
        if hijo[0] < peor_actual:  
            Intercambia hijo[0] por peor_actual en poblacion;  
  
        peor_actual = CalculaPeorSolucionPoblacion(poblacion);  
  
        if hijo[1] < peor_actual:  
            Intercambia hijo[1] por peor_actual en poblacion;  
  
    end;  
}
```

3 Algoritmos Meméticos

El AM consistirá en hibridar el algoritmo genético generacional (AGG) que mejor resultado haya proporcionado con la BL desarrollada en la Práctica 1.a.

Se estudiarán las tres posibilidades de hibridación siguientes:

- AM-(10,1.0): Cada 10 generaciones, se aplica la BL sobre todos los cromosomas de la población.
- AM-(10,0.1): Cada 10 generaciones, se aplica la BL sobre un subconjunto de cromosomas de la población seleccionado aleatoriamente con probabilidad p LS igual a 0.1 para cada cromosoma.
- AM-(10,0.1mej): Cada 10 generaciones, aplicar la BL sobre los $0.1 \cdot N$ mejores cromosomas de la población actual (N es el tamaño de ésta).

Utilizaremos los siguientes valores:

El tamaño de la población del AGG será de 10 cromosomas. Las probabilidades de cruce y mutación serán 0,7 y 0,1 (por individuo) en ambos casos. Se detendrá la ejecución de la BL aplicada sobre un cromosoma bien cuando no se encuentre mejora en todo el entorno o bien cuando se hayan evaluado 400 vecinos distintos en la ejecución. El criterio de parada del AM consistirá en realizar 100000 evaluaciones de la función objetivo, incluidas por supuesto las de la BL. Debido al mayor tiempo de los AGs y AMs, se realizará una única ejecución sobre cada caso del problema.

AM-(10,1.0): Esta posibilidad consistirá en hibridar AGG uniforme ya que es el AG que mejores resultados obtiene como vemos en la tabla 5, y la Búsqueda Local de la práctica 1.

En este caso cada vez que ejecutemos 10 veces AGG uniforme la siguiente ejecución será a través de BL en la que aplicaremos a toda la población. Después de la BL sustituiremos la solución inicial de la población por la nueva generada.

AM-(10,0.1): A diferencia de AM-(10,1.0), en este caso escogeremos solo el 10% de la población para aplicarle la BL. Como en este problema el tamaño de la población es 10 cogeremos un cromosoma aleatorio de la población que es el equivalente al 10% sobre el total.

AM-(10,0.1mej): En esta posibilidad escogeremos el 10% de los mejores resultados que es equivalente a coger al mejor resultado debido a que el tamaño de la población es 10.

4 Experimentos y análisis de resultados

Después de experimentar con los 50 casos de datos ejecutados con los diferentes algoritmos descritos anteriormente ya podemos analizar resultados y llegar a una conclusión.

Si observamos la tabla de resultados ?? podemos ver que todos los algoritmos de esta práctica obtienen mejores resultado que la Búsqueda Local y Greedy de la práctica 1, exceptuando el modelo AG Estacionario donde el uniforme es solo mejor que greedy y el de posición es el peor de todos. En contra de esto último también podemos decir que estos algoritmos son mucho más lentos que BL o Greedy y se observa una gran diferencia en el tiempo medio.

Si tenemos en cuenta estos factores tendríamos que cuestionarnos si la diferencia de desviación entre BL y AM-(10, 1.0) merece la pena por consumir más tiempo en AM-(10,1.0) y obtener mejores resultados si no, nos quedaríamos con BL que requiere mucho menos tiempo de ejecución. Si tenemos que analizar los AGs vemos que el modelo generacional obtiene mejores resultados y la diferenciade tiempos no es muy grande por lo tanto nos quedaríamos con el Generacional y en cuanto al tipo de cruce en ambos modelos, Generacional y Estacionario, de manera uniforme es mucho mejor que por posición por lo que podemos observar.

A continuación voy a insertar imágenes con las tablas resultantes de aplicar cada algoritmo.

Algoritmo BL			
Caso	Coste medio obtenido	Desv	Tiempo
GKD-b_1_n25_m2	0,0000	0,00	3,92E-05
GKD-b_2_n25_m2	0,0000	0,00	4,78E-05
GKD-b_3_n25_m2	0,0000	0,00	4,31E-05
GKD-b_4_n25_m2	0,0000	0,00	4,22E-05
GKD-b_5_n25_m2	0,0000	0,00	4,44E-05
GKD-b_6_n25_m7	39,2021	67,56	4,23E-04
GKD-b_7_n25_m7	27,9700	49,59	3,45E-04
GKD-b_8_n25_m7	38,5928	56,57	3,38E-04
GKD-b_9_n25_m7	39,8396	57,16	3,24E-04
GKD-b_10_n25_m7	38,1981	39,09	5,02E-04
GKD-b_11_n50_m5	30,2544	93,63	6,44E-04
GKD-b_12_n50_m5	14,6676	85,54	4,27E-04
GKD-b_13_n50_m5	16,8537	85,98	4,51E-04
GKD-b_14_n50_m5	16,1220	89,68	4,62E-04
GKD-b_15_n50_m5	31,6256	90,98	4,61E-04
GKD-b_16_n50_m15	121,1840	64,73	3,13E-03
GKD-b_17_n50_m15	101,3580	52,54	3,49E-03
GKD-b_18_n50_m15	101,1020	57,27	3,17E-03
GKD-b_19_n50_m15	120,8550	61,60	4,52E-03
GKD-b_20_n50_m15	110,7570	56,92	3,36E-03
GKD-b_21_n100_m10	40,4112	65,77	3,48E-03
GKD-b_22_n100_m10	31,8462	57,09	4,04E-03
GKD-b_23_n100_m10	39,2590	60,91	3,77E-03
GKD-b_24_n100_m10	40,0255	78,41	2,46E-03
GKD-b_25_n100_m10	44,3333	61,20	2,82E-03
GKD-b_26_n100_m30	396,5660	57,45	2,44E-02
GKD-b_27_n100_m30	363,1230	65,00	2,43E-02
GKD-b_28_n100_m30	370,4030	71,28	2,30E-02
GKD-b_29_n100_m30	316,0250	56,51	2,96E-02
GKD-b_30_n100_m30	300,1200	57,52	1,98E-02
GKD-b_31_n125_m12	36,1022	67,47	1,28E-02
GKD-b_32_n125_m12	43,7893	57,09	7,16E-03
GKD-b_33_n125_m12	60,8093	69,53	5,92E-03
GKD-b_34_n125_m12	44,7439	56,44	7,22E-03
GKD-b_35_n125_m12	67,8362	73,30	6,77E-03
GKD-b_36_n125_m37	330,4210	52,96	6,11E-02
GKD-b_37_n125_m37	480,3190	58,59	4,41E-02
GKD-b_38_n125_m37	542,6080	65,36	4,06E-02
GKD-b_39_n125_m37	316,5290	46,74	5,84E-02
GKD-b_40_n125_m37	419,8180	57,55	7,46E-02
GKD-b_41_n150_m15	67,3402	65,33	1,32E-02
GKD-b_42_n150_m15	78,0787	65,69	1,73E-02
GKD-b_43_n150_m15	70,8865	62,26	1,20E-02
GKD-b_44_n150_m15	73,5100	64,72	1,71E-02
GKD-b_45_n150_m15	64,3543	56,84	1,23E-02
GKD-b_46_n150_m45	570,1420	60,05	9,49E-02
GKD-b_47_n150_m45	498,8750	54,18	8,79E-02
GKD-b_48_n150_m45	505,2170	55,12	6,22E-02
GKD-b_49_n150_m45	519,4510	56,41	9,21E-02
GKD-b_50_n150_m45	496,8310	49,91	8,97E-02

Algoritmo Greedy			
Caso	Coste medio obtenido	Desv	Tiempo
GKD-b_1_n25_m2	0,0000	0,00	8,02E-07
GKD-b_2_n25_m2	0,0000	0,00	8,97E-04
GKD-b_3_n25_m2	0,0000	0,00	9,90E-04
GKD-b_4_n25_m2	0,0000	0,00	6,93E-04
GKD-b_5_n25_m2	0,0000	0,00	6,00E-07
GKD-b_6_n25_m7	49,2781	74,19	9,59E-05
GKD-b_7_n25_m7	70,0438	79,87	8,20E-05
GKD-b_8_n25_m7	42,8261	60,86	8,19E-05
GKD-b_9_n25_m7	78,3335	78,21	8,14E-05
GKD-b_10_n25_m7	80,0830	70,95	7,71E-05
GKD-b_11_n50_m5	26,7260	92,79	8,42E-05
GKD-b_12_n50_m5	29,4615	92,80	8,66E-05
GKD-b_13_n50_m5	36,7743	93,58	1,00E-04
GKD-b_14_n50_m5	24,6666	93,26	8,13E-05
GKD-b_15_n50_m5	28,2984	89,92	8,24E-05
GKD-b_16_n50_m15	181,7840	76,49	4,50E-04
GKD-b_17_n50_m15	187,2190	74,30	4,82E-04
GKD-b_18_n50_m15	168,9950	74,44	4,86E-04
GKD-b_19_n50_m15	193,3450	76,00	6,10E-04
GKD-b_20_n50_m15	173,2890	72,47	4,51E-04
GKD-b_21_n100_m10	75,3892	81,65	4,89E-04
GKD-b_22_n100_m10	71,4620	80,88	5,04E-04
GKD-b_23_n100_m10	74,1665	79,31	5,43E-04
GKD-b_24_n100_m10	69,8125	87,62	5,60E-04
GKD-b_25_n100_m10	82,0937	79,05	5,84E-04
GKD-b_26_n100_m30	439,7110	61,63	2,96E-03
GKD-b_27_n100_m30	431,9330	70,57	3,83E-03
GKD-b_28_n100_m30	457,5030	76,75	2,72E-03
GKD-b_29_n100_m30	325,5970	57,78	3,14E-03
GKD-b_30_n100_m30	455,2380	72,00	2,68E-03
GKD-b_31_n125_m12	107,1910	89,04	1,25E-03
GKD-b_32_n125_m12	69,6034	73,01	1,21E-03
GKD-b_33_n125_m12	103,7390	82,14	9,14E-04
GKD-b_34_n125_m12	63,9608	69,53	8,60E-04
GKD-b_35_n125_m12	84,8231	78,65	9,63E-04
GKD-b_36_n125_m37	458,5780	66,11	5,30E-03
GKD-b_37_n125_m37	463,8730	57,12	5,73E-03
GKD-b_38_n125_m37	545,3780	65,53	5,20E-03
GKD-b_39_n125_m37	447,0590	62,29	5,58E-03
GKD-b_40_n125_m37	415,8630	57,15	7,40E-03
GKD-b_41_n150_m15	109,1780	78,62	1,39E-03
GKD-b_42_n150_m15	142,6770	81,22	2,04E-03
GKD-b_43_n150_m15	103,7920	74,22	1,78E-03
GKD-b_44_n150_m15	137,1490	81,09	1,80E-03
GKD-b_45_n150_m15	112,6590	75,35	2,03E-03
GKD-b_46_n150_m45	538,0100	57,67	1,12E-02
GKD-b_47_n150_m45	578,9720	60,52	8,91E-03
GKD-b_48_n150_m45	448,7150	49,47	8,22E-03
GKD-b_49_n150_m45	559,0420	59,50	8,48E-03
GKD-b_50_n150_m45	643,3480	61,32	1,05E-02

?figurename? 1: BL y Greedy

Algoritmo AGG-uniforme			
Caso	Coste medio obtenido	Desv	Tiempo
GKD-b_1_n25_m2	0,0000	0,00	6,76E-01
GKD-b_2_n25_m2	0,0000	0,00	6,46E-01
GKD-b_3_n25_m2	0,0000	0,00	6,54E-01
GKD-b_4_n25_m2	0,0000	0,00	7,09E-01
GKD-b_5_n25_m2	0,0000	0,00	6,89E-01
GKD-b_6_n25_m7	54,5039	76,67	9,23E-01
GKD-b_7_n25_m7	41,1047	65,70	7,80E-01
GKD-b_8_n25_m7	64,9880	74,21	7,46E-01
GKD-b_9_n25_m7	14,2133	-16,73	7,69E-01
GKD-b_10_n25_m7	34,1075	31,79	7,52E-01
GKD-b_11_n50_m5	17,0401	88,70	1,09E+00
GKD-b_12_n50_m5	12,5345	83,08	1,09E+00
GKD-b_13_n50_m5	11,3063	79,11	1,07E+00
GKD-b_14_n50_m5	9,5625	82,61	1,12E+00
GKD-b_15_n50_m5	16,7551	82,97	1,08E+00
GKD-b_16_n50_m15	94,6539	54,84	1,39E+00
GKD-b_17_n50_m15	93,4909	48,54	1,37E+00
GKD-b_18_n50_m15	90,7068	52,38	1,37E+00
GKD-b_19_n50_m15	73,9304	37,22	1,35E+00
GKD-b_20_n50_m15	59,3310	19,58	1,34E+00
GKD-b_21_n100_m10	23,9682	42,29	1,41E+00
GKD-b_22_n100_m10	35,3732	61,37	1,41E+00
GKD-b_23_n100_m10	49,2249	68,83	1,37E+00
GKD-b_24_n100_m10	47,5930	81,84	1,42E+00
GKD-b_25_n100_m10	34,0100	49,43	1,43E+00
GKD-b_26_n100_m30	305,7660	44,82	2,31E+00
GKD-b_27_n100_m30	262,4210	51,57	2,28E+00
GKD-b_28_n100_m30	202,7700	47,54	2,30E+00
GKD-b_29_n100_m30	212,7200	35,38	2,27E+00
GKD-b_30_n100_m30	265,7300	52,03	2,33E+00
GKD-b_31_n125_m12	50,5394	76,76	1,85E+00
GKD-b_32_n125_m12	41,2745	54,48	1,83E+00
GKD-b_33_n125_m12	69,1217	73,19	1,86E+00
GKD-b_34_n125_m12	42,6161	54,27	1,83E+00
GKD-b_35_n125_m12	45,8652	60,51	1,84E+00
GKD-b_36_n125_m37	268,1870	42,04	3,16E+00
GKD-b_37_n125_m37	275,6550	27,85	3,17E+00
GKD-b_38_n125_m37	293,8450	36,03	3,22E+00
GKD-b_39_n125_m37	238,3900	29,28	3,21E+00
GKD-b_40_n125_m37	366,7540	51,41	3,32E+00
GKD-b_41_n150_m15	49,7672	53,09	1,83E+00
GKD-b_42_n150_m15	57,9964	53,81	1,81E+00
GKD-b_43_n150_m15	84,8848	68,48	1,86E+00
GKD-b_44_n150_m15	60,1815	56,90	1,85E+00
GKD-b_45_n150_m15	47,3205	41,31	1,84E+00
GKD-b_46_n150_m45	504,0200	54,81	3,79E+00
GKD-b_47_n150_m45	401,9290	43,12	4,19E+00
GKD-b_48_n150_m45	446,3140	49,20	3,78E+00
GKD-b_49_n150_m45	455,3210	50,27	3,78E+00
GKD-b_50_n150_m45	416,1880	40,21	3,93E+00

Algoritmo AGG-posicion			
Caso	Coste medio obtenido	Desv	Tiempo
GKD-b_1_n25_m2	0,0000	0,00	6,38E-01
GKD-b_2_n25_m2	0,0000	0,00	5,76E-01
GKD-b_3_n25_m2	0,0000	0,00	6,07E-01
GKD-b_4_n25_m2	0,0000	0,00	5,79E-01
GKD-b_5_n25_m2	0,0000	0,00	6,68E-01
GKD-b_6_n25_m7	61,8657	79,44	6,88E-01
GKD-b_7_n25_m7	37,6516	62,55	6,61E-01
GKD-b_8_n25_m7	61,0909	72,56	6,96E-01
GKD-b_9_n25_m7	26,9646	36,70	6,88E-01
GKD-b_10_n25_m7	23,9225	2,75	7,03E-01
GKD-b_11_n50_m5	11,6162	83,42	9,77E-01
GKD-b_12_n50_m5	10,1656	79,14	9,69E-01
GKD-b_13_n50_m5	15,7910	85,04	9,56E-01
GKD-b_14_n50_m5	9,5625	82,61	9,59E-01
GKD-b_15_n50_m5	15,5980	81,71	9,65E-01
GKD-b_16_n50_m15	104,5050	59,10	1,22E+00
GKD-b_17_n50_m15	73,3621	34,42	1,23E+00
GKD-b_18_n50_m15	78,5105	44,98	1,18E+00
GKD-b_19_n50_m15	129,8120	64,25	1,23E+00
GKD-b_20_n50_m15	107,9950	55,82	1,20E+00
GKD-b_21_n100_m10	40,0387	65,45	1,24E+00
GKD-b_22_n100_m10	49,7180	72,52	1,24E+00
GKD-b_23_n100_m10	19,2800	20,41	1,24E+00
GKD-b_24_n100_m10	34,7906	75,16	1,20E+00
GKD-b_25_n100_m10	42,5700	59,59	1,26E+00
GKD-b_26_n100_m30	325,6640	48,19	2,03E+00
GKD-b_27_n100_m30	419,6070	69,71	2,04E+00
GKD-b_28_n100_m30	318,7440	66,63	2,05E+00
GKD-b_29_n100_m30	226,3410	39,27	2,24E+00
GKD-b_30_n100_m30	407,6330	68,73	2,14E+00
GKD-b_31_n125_m12	72,8798	83,88	1,76E+00
GKD-b_32_n125_m12	38,8723	51,66	1,66E+00
GKD-b_33_n125_m12	49,1584	62,30	1,72E+00
GKD-b_34_n125_m12	81,8112	76,18	1,80E+00
GKD-b_35_n125_m12	66,9783	72,96	1,79E+00
GKD-b_36_n125_m37	394,3170	60,58	2,89E+00
GKD-b_37_n125_m37	481,5900	58,70	2,86E+00
GKD-b_38_n125_m37	433,7130	56,66	2,81E+00
GKD-b_39_n125_m37	427,1110	60,53	2,94E+00
GKD-b_40_n125_m37	388,4730	54,13	2,92E+00
GKD-b_41_n150_m15	68,1090	65,72	1,68E+00
GKD-b_42_n150_m15	69,0668	61,21	1,66E+00
GKD-b_43_n150_m15	79,0274	66,15	1,72E+00
GKD-b_44_n150_m15	81,2807	68,09	1,62E+00
GKD-b_45_n150_m15	65,6174	57,67	1,58E+00
GKD-b_46_n150_m45	708,8040	67,87	3,38E+00
GKD-b_47_n150_m45	596,6300	61,68	3,27E+00
GKD-b_48_n150_m45	513,1570	55,81	3,40E+00
GKD-b_49_n150_m45	622,5010	63,63	3,44E+00
GKD-b_50_n150_m45	746,7460	66,67	3,49E+00

?figurename? 2: AGG

Algoritmo AGE-uniforme			
Caso	Coste medio obtenido	Desv	Tiempo
GKD-b_1_n25_m2	0,0000	0,00	4,27E-01
GKD-b_2_n25_m2	0,0000	0,00	4,34E-01
GKD-b_3_n25_m2	0,0000	0,00	4,01E-01
GKD-b_4_n25_m2	0,0000	0,00	4,33E-01
GKD-b_5_n25_m2	0,0000	0,00	3,97E-01
GKD-b_6_n25_m7	73,1682	82,62	5,14E-01
GKD-b_7_n25_m7	43,7624	67,78	4,90E-01
GKD-b_8_n25_m7	77,5898	78,40	5,07E-01
GKD-b_9_n25_m7	40,4764	57,83	5,80E-01
GKD-b_10_n25_m7	38,5625	39,67	5,50E-01
GKD-b_11_n50_m5	14,0929	86,33	6,97E-01
GKD-b_12_n50_m5	13,4091	84,18	7,93E-01
GKD-b_13_n50_m5	14,8155	84,06	7,81E-01
GKD-b_14_n50_m5	21,3097	92,20	8,77E-01
GKD-b_15_n50_m5	18,2409	84,36	7,49E-01
GKD-b_16_n50_m15	142,1600	69,93	9,74E-01
GKD-b_17_n50_m15	98,4484	51,13	9,81E-01
GKD-b_18_n50_m15	79,9769	45,99	9,96E-01
GKD-b_19_n50_m15	130,6490	64,48	9,62E-01
GKD-b_20_n50_m15	82,0040	41,81	9,76E-01
GKD-b_21_n100_m10	49,9191	72,29	1,05E+00
GKD-b_22_n100_m10	42,2850	67,69	1,04E+00
GKD-b_23_n100_m10	56,6199	72,90	1,03E+00
GKD-b_24_n100_m10	52,9199	83,67	1,02E+00
GKD-b_25_n100_m10	21,3634	19,49	1,06E+00
GKD-b_26_n100_m30	328,2670	48,60	1,89E+00
GKD-b_27_n100_m30	303,2000	58,08	1,83E+00
GKD-b_28_n100_m30	331,7580	67,93	1,85E+00
GKD-b_29_n100_m30	320,6650	57,13	1,86E+00
GKD-b_30_n100_m30	309,4830	58,81	1,95E+00
GKD-b_31_n125_m12	84,5265	86,10	1,32E+00
GKD-b_32_n125_m12	75,3149	75,05	1,36E+00
GKD-b_33_n125_m12	58,2839	68,20	1,32E+00
GKD-b_34_n125_m12	65,6393	70,31	1,29E+00
GKD-b_35_n125_m12	56,8416	68,14	1,34E+00
GKD-b_36_n125_m37	323,5210	51,96	2,71E+00
GKD-b_37_n125_m37	397,1500	49,92	2,63E+00
GKD-b_38_n125_m37	524,5670	64,17	2,57E+00
GKD-b_39_n125_m37	443,6190	62,00	2,65E+00
GKD-b_40_n125_m37	385,6980	53,80	2,69E+00
GKD-b_41_n150_m15	152,5640	84,70	1,45E+00
GKD-b_42_n150_m15	130,1660	79,42	1,44E+00
GKD-b_43_n150_m15	85,2178	68,60	1,43E+00
GKD-b_44_n150_m15	84,4473	69,29	1,44E+00
GKD-b_45_n150_m15	81,1649	65,78	1,39E+00
GKD-b_46_n150_m45	666,3050	65,82	3,27E+00
GKD-b_47_n150_m45	511,3280	55,29	3,32E+00
GKD-b_48_n150_m45	647,5610	64,98	3,28E+00
GKD-b_49_n150_m45	557,3100	59,37	3,31E+00
GKD-b_50_n150_m45	534,8530	53,47	3,34E+00

Algoritmo AGE-posicion			
Caso	Coste medio obtenido	Desv	Tiempo
GKD-b_1_n25_m2	0,0000	0,00	3,80E-01
GKD-b_2_n25_m2	0,0000	0,00	3,53E-01
GKD-b_3_n25_m2	0,0000	0,00	3,60E-01
GKD-b_4_n25_m2	0,0000	0,00	3,74E-01
GKD-b_5_n25_m2	0,0000	0,00	3,48E-01
GKD-b_6_n25_m7	99,5208	87,22	4,49E-01
GKD-b_7_n25_m7	71,3809	80,25	4,77E-01
GKD-b_8_n25_m7	75,9061	77,92	4,39E-01
GKD-b_9_n25_m7	37,7153	54,74	4,58E-01
GKD-b_10_n25_m7	47,8523	51,38	4,51E-01
GKD-b_11_n50_m5	17,9416	89,26	5,99E-01
GKD-b_12_n50_m5	36,0571	94,12	6,01E-01
GKD-b_13_n50_m5	14,8155	84,06	6,13E-01
GKD-b_14_n50_m5	19,2407	91,36	5,89E-01
GKD-b_15_n50_m5	41,4018	93,11	6,11E-01
GKD-b_16_n50_m15	217,8350	80,38	8,20E-01
GKD-b_17_n50_m15	87,2153	44,84	8,52E-01
GKD-b_18_n50_m15	153,9490	71,94	8,82E-01
GKD-b_19_n50_m15	225,2810	79,40	8,48E-01
GKD-b_20_n50_m15	192,3130	75,19	8,32E-01
GKD-b_21_n100_m10	83,5127	83,44	8,97E-01
GKD-b_22_n100_m10	45,3676	69,88	8,85E-01
GKD-b_23_n100_m10	32,2318	52,39	8,78E-01
GKD-b_24_n100_m10	55,7410	84,50	8,69E-01
GKD-b_25_n100_m10	80,4014	78,61	8,83E-01
GKD-b_26_n100_m30	502,2960	66,41	1,61E+00
GKD-b_27_n100_m30	467,4670	72,81	1,61E+00
GKD-b_28_n100_m30	474,4320	77,58	1,65E+00
GKD-b_29_n100_m30	406,4120	66,18	1,65E+00
GKD-b_30_n100_m30	469,5630	72,85	1,65E+00
GKD-b_31_n125_m12	82,1933	85,71	1,12E+00
GKD-b_32_n125_m12	70,1176	73,20	1,12E+00
GKD-b_33_n125_m12	88,3272	79,02	1,15E+00
GKD-b_34_n125_m12	125,2890	84,45	1,15E+00
GKD-b_35_n125_m12	91,4667	80,20	1,15E+00
GKD-b_36_n125_m37	492,8560	68,46	2,30E+00
GKD-b_37_n125_m37	572,5820	65,26	2,30E+00
GKD-b_38_n125_m37	715,1280	73,72	2,33E+00
GKD-b_39_n125_m37	627,0690	73,11	2,30E+00
GKD-b_40_n125_m37	638,6160	72,10	2,30E+00
GKD-b_41_n150_m15	125,2020	81,35	1,24E+00
GKD-b_42_n150_m15	143,6540	81,35	1,25E+00
GKD-b_43_n150_m15	161,9570	83,48	1,28E+00
GKD-b_44_n150_m15	87,3783	70,32	1,23E+00
GKD-b_45_n150_m15	149,0230	81,36	1,25E+00
GKD-b_46_n150_m45	854,8290	73,36	2,83E+00
GKD-b_47_n150_m45	917,2400	75,08	2,91E+00
GKD-b_48_n150_m45	592,1080	61,71	2,73E+00
GKD-b_49_n150_m45	881,3330	74,31	2,82E+00
GKD-b_50_n150_m45	898,8340	72,31	2,85E+00

?figurename? 3: AGE

Algoritmo AM-(10,1,0)			
Caso	Coste medio obtenido	Dev	Tiempo
GKD-b 1 n25 m2	0,0000	0,00	3,83E-01
GKD-b 2 n25 m2	0,0000	0,00	3,87E-01
GKD-b 3 n25 m2	0,0000	0,00	3,61E-01
GKD-b 4 n25 m2	0,0000	0,00	3,81E-01
GKD-b 5 n25 m2	0,0000	0,00	3,49E-01
GKD-b 6 n25 m7	55,5851	77,12	2,53E-01
GKD-b 7 n25 m7	35,4433	60,22	2,79E-01
GKD-b 8 n25 m7	78,0604	78,53	2,58E-01
GKD-b 9 n25 m7	14,2133	-16,73	2,57E-01
GKD-b 10 n25 m7	47,5037	51,02	2,60E-01
GKD-b 11 n50 m5	8,1658	76,41	2,36E-01
GKD-b 12 n50 m5	9,3569	77,33	2,64E-01
GKD-b 13 n50 m5	7,9416	70,25	2,29E-01
GKD-b 14 n50 m5	8,1503	79,59	2,26E-01
GKD-b 15 n50 m5	14,3993	80,19	2,28E-01
GKD-b 16 n50 m15	84,4824	49,40	2,51E-01
GKD-b 17 n50 m15	75,1683	36,00	2,55E-01
GKD-b 18 n50 m15	116,1880	62,82	2,56E-01
GKD-b 19 n50 m15	83,3543	44,32	2,53E-01
GKD-b 20 n50 m15	59,4470	19,74	2,73E-01
GKD-b 21 n100 m1	41,4761	66,65	1,95E-01
GKD-b 22 n100 m1	26,6675	48,76	2,00E-01
GKD-b 23 n100 m1	35,2063	56,41	1,89E-01
GKD-b 24 n100 m1	44,6055	80,63	1,87E-01
GKD-b 25 n100 m1	40,8966	57,94	1,87E-01
GKD-b 26 n100 m3	277,3990	39,17	3,93E-01
GKD-b 27 n100 m3	243,0810	47,71	3,44E-01
GKD-b 28 n100 m3	292,1860	63,59	3,57E-01
GKD-b 29 n100 m3	375,8620	63,43	3,54E-01
GKD-b 30 n100 m3	259,4170	50,86	3,45E-01
GKD-b 31 n125 m1	35,1004	66,54	2,07E-01
GKD-b 32 n125 m1	44,5284	57,80	2,11E-01
GKD-b 33 n125 m1	60,7439	69,49	2,15E-01
GKD-b 34 n125 m1	40,8386	52,28	2,00E-01
GKD-b 35 n125 m1	46,8547	61,34	2,15E-01
GKD-b 36 n125 m3	261,9280	40,66	5,05E-01
GKD-b 37 n125 m3	314,1110	36,68	4,84E-01
GKD-b 38 n125 m3	276,6950	32,07	4,69E-01
GKD-b 39 n125 m3	245,5920	31,35	5,50E-01
GKD-b 40 n125 m3	387,5670	54,02	4,05E-01
GKD-b 41 n150 m1	55,1656	57,68	2,09E-01
GKD-b 42 n150 m1	59,4676	54,95	2,81E-01
GKD-b 43 n150 m1	55,2200	51,55	2,35E-01
GKD-b 44 n150 m1	70,6625	63,30	2,37E-01
GKD-b 45 n150 m1	36,2272	23,34	2,44E-01
GKD-b 46 n150 m4	625,2500	63,57	5,52E-01
GKD-b 47 n150 m4	359,6730	36,44	6,07E-01
GKD-b 48 n150 m4	439,4080	48,40	5,18E-01
GKD-b 49 n150 m4	570,6570	60,32	5,57E-01
GKD-b 50 n150 m4	418,8180	40,58	7,94E-01

Algoritmo AM-(10,0,1)			
Caso	Coste medio obtenido	Dev	Tiempo
GKD-b 1 n25 m2	0,0000	0,00	7,42E-01
GKD-b 2 n25 m2	0,0000	0,00	7,04E-01
GKD-b 3 n25 m2	0,0000	0,00	6,87E-01
GKD-b 4 n25 m2	0,0000	0,00	7,16E-01
GKD-b 5 n25 m2	0,0000	0,00	7,00E-01
GKD-b 6 n25 m7	112,8250	88,73	4,80E-01
GKD-b 7 n25 m7	35,4433	60,22	6,82E-01
GKD-b 8 n25 m7	42,6343	60,69	6,64E-01
GKD-b 9 n25 m7	37,9253	54,99	6,03E-01
GKD-b 10 n25 m7	34,1075	31,79	6,96E-01
GKD-b 11 n50 m5	23,9840	91,97	8,34E-01
GKD-b 12 n50 m5	10,5621	79,92	8,04E-01
GKD-b 13 n50 m5	19,4224	87,84	8,11E-01
GKD-b 14 n50 m5	15,7495	89,44	7,80E-01
GKD-b 15 n50 m5	25,4827	88,80	7,94E-01
GKD-b 16 n50 m15	111,3090	61,60	6,55E-01
GKD-b 17 n50 m15	96,4372	50,12	7,27E-01
GKD-b 18 n50 m15	116,1880	62,82	6,87E-01
GKD-b 19 n50 m15	84,7259	45,22	6,82E-01
GKD-b 20 n50 m15	120,3870	60,37	6,97E-01
GKD-b 21 n100 m1	42,1257	67,16	5,43E-01
GKD-b 22 n100 m1	42,6294	67,95	5,51E-01
GKD-b 23 n100 m1	42,7205	64,08	5,25E-01
GKD-b 24 n100 m1	43,7611	80,25	5,27E-01
GKD-b 25 n100 m1	30,9777	44,47	5,01E-01
GKD-b 26 n100 m3	311,0180	45,75	5,92E-01
GKD-b 27 n100 m3	435,6680	70,83	5,38E-01
GKD-b 28 n100 m3	439,5870	75,80	5,60E-01
GKD-b 29 n100 m3	282,7520	51,39	5,63E-01
GKD-b 30 n100 m3	245,3780	48,05	5,78E-01
GKD-b 31 n125 m1	31,8333	63,10	5,94E-01
GKD-b 32 n125 m1	40,4277	53,52	5,58E-01
GKD-b 33 n125 m1	45,5634	59,33	5,42E-01
GKD-b 34 n125 m1	49,2491	60,43	5,40E-01
GKD-b 35 n125 m1	36,3723	50,20	5,69E-01
GKD-b 36 n125 m3	368,9280	57,87	6,53E-01
GKD-b 37 n125 m3	362,5000	45,13	6,07E-01
GKD-b 38 n125 m3	506,6310	62,90	6,21E-01
GKD-b 39 n125 m3	308,7460	45,40	6,17E-01
GKD-b 40 n125 m3	366,7420	51,41	6,99E-01
GKD-b 41 n150 m1	60,5475	61,44	5,43E-01
GKD-b 42 n150 m1	62,7022	57,28	5,42E-01
GKD-b 43 n150 m1	57,5824	53,54	4,48E-01
GKD-b 44 n150 m1	69,1906	62,52	5,49E-01
GKD-b 45 n150 m1	49,0055	43,33	4,72E-01
GKD-b 46 n150 m4	476,5200	52,21	7,53E-01
GKD-b 47 n150 m4	486,3010	52,99	7,24E-01
GKD-b 48 n150 m4	569,9580	60,22	5,98E-01
GKD-b 49 n150 m4	471,8750	52,02	8,09E-01
GKD-b 50 n150 m4	565,8170	56,02	7,00E-01

Algoritmo AM-(10,0,1 mej)			
Caso	Coste medio obtenido	Dev	Tiempo
GKD-b 1 n25 m2	0,0000	0,00	7,65E-01
GKD-b 2 n25 m2	0,0000	0,00	6,95E-01
GKD-b 3 n25 m2	0,0000	0,00	6,78E-01
GKD-b 4 n25 m2	0,0000	0,00	7,09E-01
GKD-b 5 n25 m2	0,0000	0,00	6,93E-01
GKD-b 6 n25 m7	112,8250	88,73	4,66E-01
GKD-b 7 n25 m7	55,6852	74,68	6,09E-01
GKD-b 8 n25 m7	42,6343	60,69	6,53E-01
GKD-b 9 n25 m7	20,5190	16,81	6,78E-01
GKD-b 10 n25 m7	30,7658	24,38	6,65E-01
GKD-b 11 n50 m5	21,9155	91,21	8,11E-01
GKD-b 12 n50 m5	10,5621	79,92	7,81E-01
GKD-b 13 n50 m5	14,5863	83,80	7,79E-01
GKD-b 14 n50 m5	16,9528	90,19	7,81E-01
GKD-b 15 n50 m5	10,7082	73,36	7,77E-01
GKD-b 16 n50 m15	123,3350	65,34	6,96E-01
GKD-b 17 n50 m15	91,9723	47,69	7,20E-01
GKD-b 18 n50 m15	100,3890	56,97	6,87E-01
GKD-b 19 n50 m15	115,0240	59,65	6,74E-01
GKD-b 20 n50 m15	79,7455	40,17	6,60E-01
GKD-b 21 n100 m10	41,4761	66,65	5,10E-01
GKD-b 22 n100 m10	30,6850	55,47	5,33E-01
GKD-b 23 n100 m10	47,5107	67,70	5,65E-01
GKD-b 24 n100 m10	39,4483	78,10	5,23E-01
GKD-b 25 n100 m10	38,1137	54,87	5,08E-01
GKD-b 26 n100 m30	264,1570	36,13	5,95E-01
GKD-b 27 n100 m30	251,5230	49,47	5,76E-01
GKD-b 28 n100 m30	373,5660	71,52	5,60E-01
GKD-b 29 n100 m30	383,8200	64,19	5,59E-01
GKD-b 30 n100 m30	315,8340	59,64	5,67E-01
GKD-b 31 n125 m12	49,3887	76,22	6,83E-01
GKD-b 32 n125 m12	28,4336	33,92	5,48E-01
GKD-b 33 n125 m12	43,7645	57,66	5,24E-01
GKD-b 34 n125 m12	42,0501	53,65	5,98E-01
GKD-b 35 n125 m12	30,0198	39,67	5,40E-01
GKD-b 36 n125 m37	319,6590	51,37	6,49E-01
GKD-b 37 n125 m37	310,6150	35,97	6,37E-01
GKD-b 38 n125 m37	425,6010	55,83	7,81E-01
GKD-b 39 n125 m37	296,2900	43,10	8,35E-01
GKD-b 40 n125 m37	414,0070	56,96	6,77E-01
GKD-b 41 n150 m15	80,3944	70,96	5,82E-01
GKD-b 42 n150 m15	73,9238	63,76	5,07E-01
GKD-b 43 n150 m15	74,8369	64,25	4,68E-01
GKD-b 44 n150 m15	50,4611	48,60	4,26E-01
GKD-b 45 n150 m15	56,1173	50,51	4,61E-01
GKD-b 46 n150 m45	431,1580	47,18	7,40E-01
GKD-b 47 n150 m45	466,2930	50,97	8,37E-01
GKD-b 48 n150 m45	282,6040	19,77	8,45E-01
GKD-b 49 n150 m45	580,3050	60,98	6,75E-01
GKD-b 50 n150 m45	745,8810	66,64	5,82E-01

?figurename? 4: AM

Algoritmo	Desv	Tiempo
Gready	66,53792625876	0,002293429016
BL	56,71072367558	0,019550993792
AGG-uniforme	48,255379081	1,84162086
AGG-posicion	55,64489490996	1,64951558
AGE-uniforme	59,07472171518	1,43318966
AGE-posicion	67,79319419217	1,23076892
AM-(10, 1.0)	48,47510123059	0,32171192
AM-(10, 0.1)	54,62159382653	0,63516988
AM-(10, 0.1mej)	52,10578737499	0,64134536

?figurename? 5: Resultados