Informe Práctica 8

MDP solved using Branch and Bound, Greedy and Grasp procedures

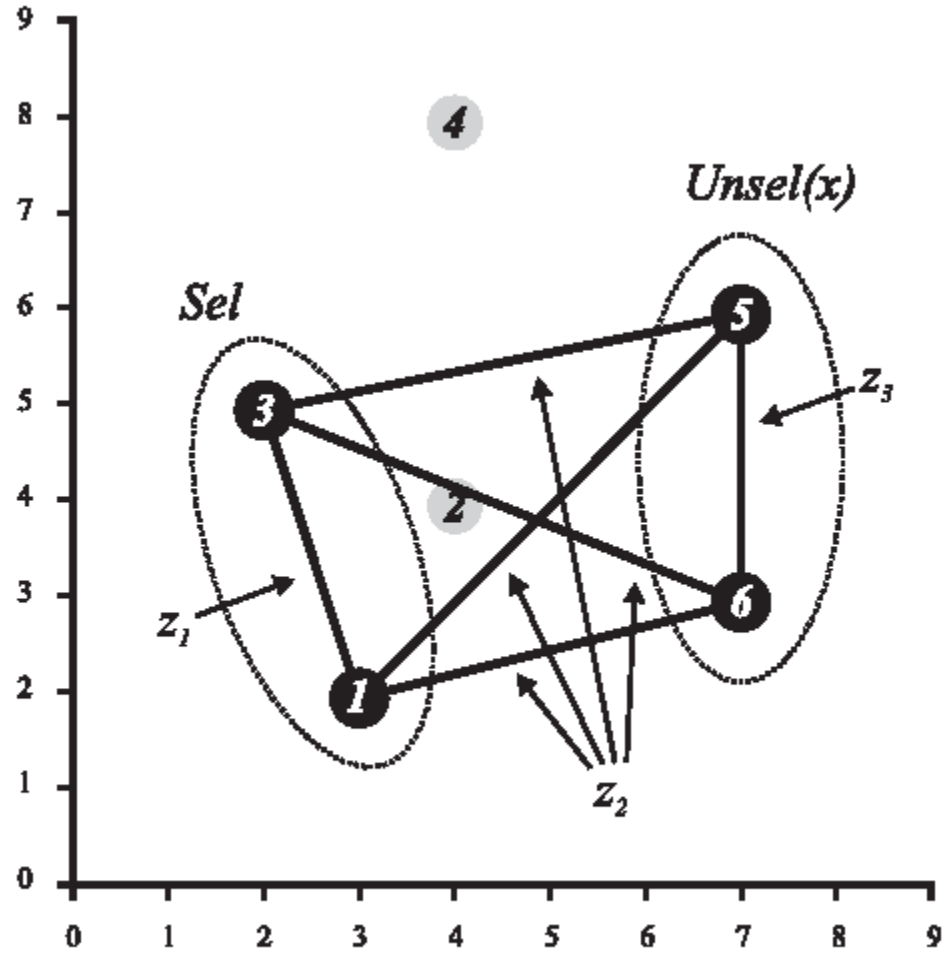

Figure 2. Decomposition of the solution value

Juan García Santos

20/05/2022

3º Ingeniería Informática

DAA

1. Una descripción de la arquitectura en la que se ha hecho la ejecución.

La ejecución de los algoritmos se ha hecho en un equipo con windows 11 como SO, 32 gb de memoria ram a 2600 mhz, un procesador intel core i7-9700K a frecuencia base y una tarjeta gráfica nvidia gtx 1080 de 8 gb.

2. Descripción de la estructura de datos que almacena las soluciones, la estructura de entorno usada en la búsqueda local, cómo se construye la RCL en GRASP y cómo se obtienen las cotas del algoritmo de ramificación y poda.

- **Clase Problema:** Esta clase representa nuestro problema y por tanto almacena la siguiente información:
    - Un entero que contiene el número de elementos disponibles
    - Un entero que contiene el tamaño de un elemento
    - Matriz de doubles que contiene en cada fila un elemento, pues un elemento se codifica como una lista de doubles.
    - Una matriz de floats donde se almacena la distancia entre el elemento de la posición i con el elemento de la posición j, de esta forma el acceso al coste será tan sencillo y eficiente como añadir matriz[i][j].
- **Clase Solución:** Esta clase representa una solución al problema. Para ello contiene los siguientes atributos:
    - Un entero que contiene el número de elementos disponibles
    - Un entero que contiene el tamaño de un elemento
    - Un flotante que almacena el valor de la solución (diversidad)
    - Un entero con el número de nodos expandidos (solo para el Branch and Bound)
    - Matriz de doubles que contiene en cada fila un elemento, pues un elemento se codifica como una lista de doubles. Esta matriz almacena los elementos contenidos en la solución.
- **Clase Partial Solution:** almacena la misma información que una solución, pero como se usan de elemento intermedio para diversas operaciones se requiere que sean ligeras. Por ello, codifican la información en listas de enteros, que almacenan los índices de los elementos que contiene esa solución parcial, en lugar de los propios elementos.
- **Estructura de Entorno:** La estructura de entorno empleada para la búsqueda local en

este problema es la de intercambiar un elemento de la solución por otro que no estuviera ya dentro. Se re calcula el coste y se selecciona aquella solución que dé mejores resultados, hallando así un óptimo local. Este proceso se realiza mientras haya mejora.

- **RCL:** La lista restringida de candidatos se construye en la fase constructiva del Grasp, de tal forma que en cada iteración se genera una lista de los "x" mejores candidatos. De esa lista se escoge un candidato al azar como elemento a introducir en la solución que se está construyendo. Este proceso se realiza hasta que la solución posea el tamaño deseado
- **Cálculo de las cotas:** En el Branch and Bound para maximizar se emplean dos cotas, una inferior y otra superior. La inferior se obtiene como el valor de diversidad de una solución inicial dada y se actualiza cada vez que exploremos una solución parcial de tamaño completo si la diversidad de esa solución completa es mayor que la cota inferior actual. La cota superior se calcula para cada solución parcial siguiendo la fórmula explicada en este artículo. En él se proponen 3 formas de calcular la cota superior:
    - La primera consiste en añadirle a la diversidad de la solución parcial el valor de la máxima diversidad que podría aportar un elemento nuevo (que no esté ya en la solución) multiplicado por el número de elementos que faltan para que la solución parcial tenga el tamaño máximo permitido.
    - La segunda consiste en añadirle a la diversidad de la solución parcial el valor de la máxima diversidad entre todos los elementos candidatos (estos son los elementos del problema que no se encuentran en la solución parcial) multiplicado por el número de elementos que faltan para que la solución parcial tenga el tamaño máximo permitido.
    - La tercera estrategia (que es la empleada en este caso), consiste en aunar las dos ideas previas para obtener un valor máximo que se multiplica por un "peso", calculado a partir del número de elementos que faltan por añadir, y se suma a la diversidad de la solución parcial.

3. Conclusiones extraídas.

Dado que las instancias del problema dada son lo suficientemente pequeñas, el algoritmo greedy suele encontrar el valor óptimo. Es por esto, y por el número tan bajo de iteraciones que se le dan, que el algoritmo grasp puede llegar a encontrar soluciones iguales o incluso peores al greedy, salvo cuando el greedy no encuentra el valor óptimo en cuyo caso es mucho más probable que el grasp mejore la solución dada por el algoritmo Greedy.

El algoritmo branch and Bound siempre encuentra la solución óptima pero se aprecia que se suelen generar menos nodos si se parte de la solución greedy, además de que si se emplea la

expansión por profundidad se observa que genera un número aún menor de nodos. Resulta evidente que el tiempo de cómputo aumenta más si aumenta el tamaño del problema que si aumenta el tamaño máximo de solución permitido, así como puede resultar anti-intuitivo que las soluciones a los problemas con elementos de mayor tamaño sean más rápidas y generen menos nodos que sus contrapartes con elementos de menor tamaño.

4. Tablas:

1. Greedy:

Problema	n elements	K size	m max elements	z diversity	Solution	Time
max_div_15_2.txt	15	2	2	11,859215736389916	8,65 9,98 // 0,58 1,29	0
max_div_15_2.txt	15	2	3	25,726199865341187	8,65 9,98 // 0,58 1,29 // 9,96 8,17	0
max_div_15_2.txt	15	2	4	48,413930416107718	8,65 9,98 // 0,58 1,29 // 9,96 8,17 // 1,59 1,57	0
max_div_15_2.txt	15	2	5	73,561898931860926	8,65 9,98 // 0,58 1,29 // 9,96 8,17 // 1,59 1,57 // 8,41 9,98	0
max_div_15_3.txt	15	3	2	13,273240089416504	9,88 9,88 6,26 // 0,3 0,92 4,23	0
max_div_15_3.txt	15	3	3	30,324088096618652	9,88 9,88 6,26 // 0,3 0,92 4,23 // 1,71 1,95 9,22	0
max_div_15_3.txt	15	3	4	59,763757705688848	9,88 9,88 6,26 // 0,3 0,92 4,23 // 1,71 1,95 9,22 // 6,65 9,45 1,02	0
max_div_15_3.txt	15	3	5	94,748718261718750	9,88 9,88 6,26 // 0,3 0,92 4,23 // 1,71 1,95 9,22 // 6,65 9,45 1,02 // 9,47 3,56 1,83	0

max_div_20_2.txt	20	2	2	8,510329246520996	0,63 5,96 // 8,67 3,17	0
max_div_20_2.txt	20	2	3	21,996085643768831	0,63 5,96 // 8,67 3,17 // 8,57 8,36	0
max_div_20_2.txt	20	2	4	39,568228960037723	0,63 5,96 // 8,67 3,17 // 8,57 8,36 // 1,16 4,47	0
max_div_20_2.txt	20	2	5	61,239289999000818	0,63 5,96 // 8,67 3,17 // 8,57 8,36 // 1,16 4,47 // 8,78 7,43	0
max_div_20_3.txt	20	3	2	11,800309181213379	9,81 2,05 1,83 // 0,65 3,76 9,07	0
max_div_20_3.txt	20	3	3	30,872657775878906	9,81 2,05 1,83 // 0,65 3,76 9,07 // 8,87 9,56 5,34	0
max_div_20_3.txt	20	3	4	56,534726619720046	9,81 2,05 1,83 // 0,65 3,76 9,07 // 8,87 9,56 5,34 // 0,83 7,06 3,34	0
max_div_20_3.txt	20	3	5	92,829743385314944	9,81 2,05 1,83 // 0,65 3,76 9,07 // 8,87 9,56 5,34 // 0,83 7,06 3,34 // 9,6 2,02 9	0
max_div_30_2.txt	30	2	2	11,657143592834473	9,84 8,96 // 2,07 0,27	0
max_div_30_2.txt	30	2	3	28,944299697875977	9,84 8,96 // 2,07 0,27 // 1,91 9,6	0
max_div_30_2.txt	30	2	4	52,771171569824422	9,84 8,96 // 2,07 0,27 // 1,91 9,6 // 8 1,53	0
max_div_30_2.txt	30	2	5	80,910241127014416	9,84 8,96 // 2,07 0,27 // 1,91 9,6 // 8 1,53 // 0,65 3,26	0

max_div_30_3.txt	30	3	2	13,073737144470215	8,06 9,59 0,27 // 6,71 0,35 9,42	0
max_div_30_3.txt	30	3	3	33,842259407044346	8,06 9,59 0,27 // 6,71 0,35 9,42 // 1,53 8,09 9,56	0
max_div_30_3.txt	30	3	4	63,518414974211265	8,06 9,59 0,27 // 6,71 0,35 9,42 // 1,53 8,09 9,56 // 6,75 0,07 1,8	0
max_div_30_3.txt	30	3	5	99,508837699890014	8,06 9,59 0,27 // 6,71 0,35 9,42 // 1,53 8,09 9,56 // 6,75 0,07 1,8 // 7,54 8,93 9,66	0

2. Grasp:

Problema	n elements	K size	m max elements	iters	LRC size	z diversity	Solution	Time
max_div_15_2.txt	15	2	2	10	2	11,859215736389116	8,65 9,98 // 0,58 1,29	1
max_div_15_2.txt	15	2	2	10	3	11,632660865783691	9,96 8,17 // 0,58 1,29	0
max_div_15_2.txt	15	2	2	20	2	10,980514526367188	8,65 9,98 // 1,59 1,57	0
max_div_15_2.txt	15	2	2	20	3	10,546440124511719	8,41 9,98 // 0,46 3,05	0

max_div_15_2.txt	15	2	3	10	2	23,887823820114136	8,65 9,98 // 1,59 1,57 // 0,58 1,29	0
max_div_15_2.txt	15	2	3	10	3	23,571097016334534	8,65 9,98 // 1,59 1,57 // 0,46 3,05	0
max_div_15_2.txt	15	2	3	20	2	23,796437487006187	8,65 9,98 // 0,58 1,29 // 8,41 9,98	0
max_div_15_2.txt	15	2	3	20	3	23,313058495521545	9,96 8,17 // 0,46 3,05 // 1,59 1,57	0
max_div_15_2.txt	15	2	4	10	2	48,413930416107118	8,65 9,98 // 1,59 1,57 // 0,58 1,29 // 9,96 8,17	0
max_div_15_2.txt	15	2	4	10	3	49,3307147	9,96 8,17 // 0,58 1,29 // 0,16 4,62 // 8,41 9,98	0
max_div_15_2.txt	15	2	4	20	2	48,413930416107118	8,65 9,98 // 1,59 1,57 // 9,96 8,17 // 0,58 1,29	0

max_div_15_2.txt	15	2	4	20	3	47,34906780719757	9,96 8,17 // 0,58 1,29 // 0,16 4,62 // 9,71 6,49	0
max_div_15_2.txt	15	2	5	10	2	73,561898931860922	8,65 9,98 // 0,58 1,29 // 9,96 8,17 // 1,59 1,57 // 8,41 9,98	0
max_div_15_2.txt	15	2	5	10	3	73,761098027229331	8,65 9,98 // 1,59 1,57 // 0,58 1,29 // 9,71 6,49 // 9,96 8,17	0
max_div_15_2.txt	15	2	5	20	2	73,21230769	8,41 9,98 // 0,58 1,29 // 9,96 8,17 // 0,46 3,05 // 1,59 1,57	0
max_div_15_2.txt	15	2	5	20	3	72,686980009078988	8,65 9,98 // 1,59 1,57 // 0,46 3,05 // 9,96 8,17 // 0,16 4,62	1

max_div_15_3.txt	15	3	2	10	2	11,764157295227055	9,88 9,88 6,26 // 1,71 1,95 9,22	0
max_div_15_3.txt	15	3	2	10	3	11,359106063842773	9,88 9,88 6,26 // 1,48 2,7 3,63	0
max_div_15_3.txt	15	3	2	20	2	11,764157295227055	9,88 9,88 6,26 // 1,71 1,95 9,22	0
max_div_15_3.txt	15	3	2	20	3	12,161150932312012	1,71 1,95 9,22 // 6,65 9,45 1,02	0
max_div_15_3.txt	15	3	3	10	2	30,095834732055664	1,71 1,95 9,22 // 6,65 9,45 1,02 // 9,88 9,88 6,26	0
max_div_15_3.txt	15	3	3	10	3	27,493535280227666	8,67 9,58 4,39 // 1,71 1,95 9,22 // 6,65 9,45 1,02	0
max_div_15_3.txt	15	3	3	20	2	30,551758766174316	9,88 9,88 6,26 // 0,3 0,92 4,23 // 6,65 9,45 1,02	0

max_div_15_3.txt	15	3	3	20	3	30,324088096618652	1,71 1,95 9,22 // 9,88 9,88 6,26 // 0,3 0,92 4,23	0
max_div_15_3.txt	15	3	4	10	2	55,003021001815796	9,88 9,88 6,26 // 0,3 0,92 4,23 // 6,65 9,45 1,02 // 2,15 2,48 6,87	0
max_div_15_3.txt	15	3	4	10	3	58,39569568634033	1,71 1,95 9,22 // 9,88 9,88 6,26 // 5,17 1,39 0,91 // 6,65 9,45 1,02	0
max_div_15_3.txt	15	3	4	20	2	56,01756930351257	1,71 1,95 9,22 // 9,88 9,88 6,26 // 0,3 0,92 4,23 // 8,67 9,58 4,39	0
max_div_15_3.txt	15	3	4	20	3	58,39569568634033	1,71 1,95 9,22 // 6,65 9,45 1,02 // 9,88 9,88 6,26 // 5,17 1,39 0,91	0

max_div_15_3.txt	15	3	5	10	2	91,596895933151255	9,88 9,88 6,26 // 0,3 0,92 4,23 // 1,71 1,95 9,22 // 8,67 9,58 4,39 // 5,17 1,39 0,91	0
max_div_15_3.txt	15	3	5	10	3	89,38842201	1,71 1,95 9,22 // 6,65 9,45 1,02 // 9,88 9,88 6,26 // 0,3 0,92 4,23 // 8,67 9,58 4,39	0
max_div_15_3.txt	15	3	5	20	2	94,748718261718755	1,71 1,95 9,22 // 6,65 9,45 1,02 // 9,47 3,56 1,83 // 9,88 9,88 6,26 // 0,3 0,92 4,23	0
max_div_15_3.txt	15	3	5	20	3	89,38842201	8,67 9,58 4,39 // 0,3 0,92 4,23 // 9,88 9,88 6,26 // 1,71 1,95 9,22 // 6,65 9,45 1,02	1

max_div_20_2.txt	20	2	2	10	2	8,510329246520996	0,63 5,96 // 8,67 3,17	0
max_div_20_2.txt	20	2	2	10	3	8,294793128967285	0,63 5,96 // 8,57 8,36	0
max_div_20_2.txt	20	2	2	20	2	8,369002342224121	1,16 4,47 // 8,57 8,36	0
max_div_20_2.txt	20	2	2	20	3	8,281509399414062	0,63 5,96 // 8,78 7,43	0
max_div_20_2.txt	20	2	3	10	2	21,458415508270264	0,63 5,96 // 8,57 8,36 // 8,45 3,42	0
max_div_20_2.txt	20	2	3	10	3	19,845169544219967	0,63 5,96 // 8,78 7,43 // 8,14 3,75	0
max_div_20_2.txt	20	2	3	20	2	21,996085643768311	0,63 5,96 // 8,67 3,17 // 8,57 8,36	0
max_div_20_2.txt	20	2	3	20	3	20,724951267242433	1,16 4,47 // 7,75 8,48 // 8,67 3,17	0
max_div_20_2.txt	20	2	4	10	2	40,002256870269775	1,16 4,47 // 8,57 8,36 // 8,67 3,17	0

							// 3,47 9,43	
max_div_20_2.txt	20	2	4	10	3	35,25458037853241	8,67 3,17 // 1,53 6,79 // 3,47 9,43 // 8,45 3,42	0
max_div_20_2.txt	20	2	4	20	2	39,56822896003723	0,63 5,96 // 8,57 8,36 // 8,67 3,17 // 1,16 4,47	0
max_div_20_2.txt	20	2	4	20	3	38,431116342544556	0,63 5,96 // 8,67 3,17 // 8,78 7,43 // 1,16 4,47	0
max_div_20_2.txt	20	2	5	10	2	62,19887375831604	0,63 5,96 // 8,67 3,17 // 8,78 7,43 // 1,16 4,47 // 3,47 9,43	0
max_div_20_2.txt	20	2	5	10	3	60,44711840152740 5	1,16 4,47 // 8,57 8,36 // 8,67 3,17 // 3,47 9,43 // 8,45 3,42	0

max_div_20_2.txt	20	2	5	20	2	57,836700201034546	0,63 5,96 // 8,67 3,17 // 8,78 7,43 // 1,53 6,79 // 8,45 3,42	1
max_div_20_2.txt	20	2	5	20	3	56,170774430036545	1,16 4,47 // 7,75 8,48 // 8,14 3,75 // 1,53 6,79 // 8,45 3,42	1
max_div_20_3.txt	20	3	2	10	2	11,800309181213379	9,81 2,05 1,83 // 0,65 3,76 9,07	0
max_div_20_3.txt	20	3	2	10	3	11,800309181213379	9,81 2,05 1,83 // 0,65 3,76 9,07	0
max_div_20_3.txt	20	3	2	20	2	11,800309181213379	0,65 3,76 9,07 // 9,81 2,05 1,83	0
max_div_20_3.txt	20	3	2	20	3	9,914459228515625	9,6 2,02 9 // 4,84 9,69 4,9	0
max_div_20_3.txt	20	3	3	10	2	29,118255615234375	0,65 3,76 9,07 // 7,66 8,13 2,01 // 9,81 2,05	0

							1,83	
max_div_20_3.txt	20	3	3	10	3	30,004436492919922	9,81 2,05 1,83 // 0,65 3,76 9,07 // 6,92 9,54 4,14	0
max_div_20_3.txt	20	3	3	20	2	27,899738788604736	9,81 2,05 1,83 // 0,96 7,43 5,55 // 0,65 3,76 9,07	0
max_div_20_3.txt	20	3	3	20	3	30,269130706787111	0,65 3,76 9,07 // 8,87 9,56 5,34 // 8,02 0,12 3,56	0
max_div_20_3.txt	20	3	4	10	2	52,551854610443115	0,65 3,76 9,07 // 9,81 2,05 1,83 // 6,92 9,54 4,14 // 8,02 0,12 3,56	0
max_div_20_3.txt	20	3	4	10	3	53,16867375	9,6 2,02 9 // 0,83 7,06 3,34 // 9,81 2,05 1,83 // 0,96 7,43 5,55	0

max_div_20_3.txt	20	3	4	20	2	51,342039585113525	0,65 3,769,07 // 7,66 8,132,01 // 9,81 2,051,83 // 2,63 2,918,86	0
max_div_20_3.txt	20	3	4	20	3	53,16867375	9,6 2,02 9 // 0,96 7,43 5,55 // 9,81 2,05 1,83 // 0,83 7,06 3,34	0
max_div_20_3.txt	20	3	5	10	2	86,77310205	9,81 2,05 1,83 // 0,96 7,43 5,55 // 9,6 2,02 9 // 0,83 7,06 3,34 // 8,87 9,56 5,34	0
max_div_20_3.txt	20	3	5	10	3	90,63258934020996	9,6 2,02 9 // 0,96 7,43 5,55 // 9,81 2,05 1,83 // 0,65 3,76 9,07 // 8,87 9,56 5,34	0

max_div_20_3.txt	20	3	5	20	2	90,532837867736822	0,65 3,76 9,07 // 7,66 8,13 2,01 // 9,6 2,02 9 // 9,81 2,05 1,83 // 0,83 7,06 3,34	1
max_div_20_3.txt	20	3	5	20	3	85,796245813369745	9,6 2,02 9 // 0,83 7,06 3,34 // 9,81 2,05 1,83 // 0,96 7,43 5,55 // 0,65 3,76 9,07	1
max_div_30_2.txt	30	2	2	10	2	10,904696464538574	9,84 8,96 // 2,02 1,36	0
max_div_30_2.txt	30	2	2	10	3	10,904696464538574	9,84 8,96 // 2,02 1,36	0
max_div_30_2.txt	30	2	2	20	2	9,97562027	2,07 0,27 // 8,4 7,98	0
max_div_30_2.txt	30	2	2	20	3	10,110044479370117	1,91 9,6 // 8 1,53	0
max_div_30_2.txt	30	2	3	10	2	27,101214408874501	9,84 8,96 // 2,02 1,36 // 1,91 9,6	0

max_div_30_2.txt	30	2	3	10	3	27,594313144683838	2,07 0,27 // 9,84 8,96 // 2,93 9,25	0
max_div_30_2.txt	30	2	3	20	2	28,022974491119385	2,07 0,27 // 9,84 8,96 // 1,37 8,09	0
max_div_30_2.txt	30	2	3	20	3	24,760816097259 52	1,91 9,6 // 7,95 3 // 2,07 0,27	0
max_div_30_2.txt	30	2	4	10	2	44,564399123191 83	2,07 0,27 // 8,4 7,98 // 9,84 8,96 // 2,02 1,36	0
max_div_30_2.txt	30	2	4	10	3	51,066671848297 12	9,84 8,96 // 2,07 0,27 // 1,37 8,09 // 8 1,53	0
max_div_30_2.txt	30	2	4	20	2	49,180876493453 98	2,07 0,27 // 9,84 8,96 // 1,91 9,6 // 2,02 1,36	0
max_div_30_2.txt	30	2	4	20	3	50,549280643463 135	1,91 9,6 // 8 1,53 // 9,84 8,96 // 0,65 3,26	1

max_div_30_2.txt	30	2	5	10	2	70,10522329807281	2,07 0,27 // 8,4 7,98 // 9,84 8,96 // 0,65 3,26 // 2,02 1,36	0
max_div_30_2.txt	30	2	5	10	3	78,019540071487 43	9,84 8,96 // 2,02 1,36 // 0,65 3,26 // 1,91 9,6 // 8 1,53	0
max_div_30_2.txt	30	2	5	20	2	78,990164041519 17	2,07 0,27 // 9,84 8,96 // 1,91 9,6 // 2,02 1,36 // 8 1,53	1
max_div_30_2.txt	30	2	5	20	3	77,013294816017 15	9,84 8,96 // 0,65 3,26 // 2,07 0,27 // 8,4 7,98 // 1,91 9,6	1
max_div_30_3.txt	30	3	2	10	2	12,576001167297 363	1,53 8,09 9,56 // 8,39 1,48 1,35	6
max_div_30_3.txt	30	3	2	10	3	12,320162773132 324	6,75 0,07 1,8 // 1,53 8,09 9,56	0
max_div_30_3.txt	30	3	2	20	2	12,613726615905 762	8,06 9,59 0,27 //	0

							8,32 0,47 8,98	
max_div_30_3.txt	30	3	2	20	3	12,320162773132324	6,75 0,07 1,8 // 1,53 8,09 9,56	0
max_div_30_3.txt	30	3	3	10	2	34,290529251098663	8,06 9,59 0,27 // 8,32 0,47 8,98 // 1,53 8,09 9,56	0
max_div_30_3.txt	30	3	3	10	3	33,842259407043466	1,53 8,09 9,56 // 8,06 9,59 0,27 // 6,71 0,35 9,42	0
max_div_30_3.txt	30	3	3	20	2	31,505428314208984	8,06 9,59 0,27 // 6,71 0,35 9,42 // 0,92 3,37 5,61	0
max_div_30_3.txt	30	3	3	20	3	31,028532981872566	6,75 0,07 1,8 // 7,54 8,93 9,66 // 8,06 9,59 0,27	0
max_div_30_3.txt	30	3	4	10	2	62,751654148101811	1,53 8,09 9,56 // 8,39 1,48 1,35 // 8,06 9,59 0,27 // 8,32 0,47	0

							8,98	
max_div_30_3.txt	30	3	4	10	3	60,871976375579834	6,75 0,07 1,8 // 1,53 8,09 9,56 // 7,54 8,93 9,66 // 8,06 9,59 0,27	0
max_div_30_3.txt	30	3	4	20	2	61,482624053955008	1,53 8,09 9,56 // 8,39 1,48 1,35 // 9,99 4,4 9,76 // 8,06 9,59 0,27	1
max_div_30_3.txt	30	3	4	20	3	59,615173339843755	8,06 9,59 0,27 // 8,32 0,47 8,98 // 0,92 3,37 5,61 // 7,54 8,93 9,66	1
max_div_30_3.txt	30	3	5	10	2	94,212804317474377	1,53 8,09 9,56 // 8,39 1,48 1,35 // 9,99 4,4 9,76 // 3,26 6,63 1,22 // 8,06 9,59 0,27	1

max_div_30_3.txt	30	3	5	10	3	85,01454088	6,75 0,07 1,8 // 7,47 8,56 9,58 // 1,53 8,09 9,56 // 8,39 1,48 1,35 // 7,54 8,93 9,66	1
max_div_30_3.txt	30	3	5	20	2	97,964369773864675	8,06 9,59 0,27 // 8,32 0,47 8,98 // 1,53 8,09 9,56 // 8,39 1,48 1,35 // 7,54 8,93 9,66	1
max_div_30_3.txt	30	3	5	20	3	98,583616256713877	6,75 0,07 1,8 // 1,53 8,09 9,56 // 8,06 9,59 0,27 // 9,99 4,4 9,76 // 6,71 0,35 9,42	1

3. Branch and Bound Greedy en profundidad:

Problema	n elements	K size	m max elements	z diversity	Solution	Time	Nodes
max_div_15_2.txt	15	2	2	11,85921573638916	8,65 9,98 // 0,58 1,29	2	119

max_div_15_2.txt	15	2	3	27,372700214385986	9,11 3,23 // 0,58 1,29 // 8,65 9,98	0	413
max_div_15_2.txt	15	2	4	49,826781272888184	9,11 3,23 // 0,16 4,62 // 0,58 1,29 // 8,65 9,98	3	1559
max_div_15_2.txt	15	2	5	79,12953066825867	9,11 3,23 // 9,96 8,17 // 0,16 4,62 // 0,58 1,29 // 8,65 9,98	11	3808
max_div_15_3.txt	15	3	2	13,273240089416504	9,88 9,88 6,26 // 0,3 0,92 4,23	0	119
max_div_15_3.txt	15	3	3	31,868525505065918	1,71 1,95 9,22 // 5,17 1,39 0,91 // 9,88 9,88 6,26	0	469
max_div_15_3.txt	15	3	4	59,76375770568848	9,88 9,88 6,26 // 0,3 0,92 4,23 // 1,71 1,95 9,22 // 6,65 9,45 1,02	2	1232
max_div_15_3.txt	15	3	5	96,08583307266235	1,15 9,21 3,11 // 1,71 1,95 9,22 // 0,3 0,92 4,23 // 9,88 9,88 6,26 // 9,47 3,56 1,83	8	3038
max_div_20_2.txt	20	2	2	8,510329246520996	0,63 5,96 // 8,67 3,17	0	209
max_div_20_2.txt	20	2	3	21,99608564376831	0,63 5,96 // 8,67 3,17 // 8,57 8,36	1	580
max_div_20_2.txt	20	2	4	40,002256870269775	3,47 9,43 // 1,16 4,47 // 8,57 8,36 // 8,67 3,17	8	2978
max_div_20_2.txt	20	2	5	63,651679277420044	3,47 9,43 // 8,57 8,36 // 1,97 3,5 // 0,63 5,96 // 8,67 3,17	49	9616
max_div_20_3.txt	20	3	2	11,800309181213379	9,81 2,05 1,83 // 0,65 3,76 9,07	0	209

max_div_20_3.txt	20	3	3	30,872657775878906	9,81 2,05 1,83 // 0,65 3,76 9,07 // 8,87 9,56 5,34	1	507
max_div_20_3.txt	20	3	4	56,69031047821045	0,83 7,06 3,34 // 9,81 2,05 1,83 // 0,65 3,76 9,07 // 9,6 2,02 9	10	2508
max_div_20_3.txt	20	3	5	92,82974338531494	9,81 2,05 1,83 // 0,65 3,76 9,07 // 8,87 9,56 5,34 // 0,83 7,06 3,34 // 9,6 2,02 9	26	6477
max_div_30_2.txt	30	2	2	11,657143592834473	9,84 8,96 // 2,07 0,27	2	464
max_div_30_2.txt	30	2	3	28,944299697875977	9,84 8,96 // 2,07 0,27 // 1,91 9,6	10	2207
max_div_30_2.txt	30	2	4	52,77117156982422	9,84 8,96 // 2,07 0,27 // 1,91 9,6 // 8 1,53	107	12174
max_div_30_2.txt	30	2	5	80,91024112701416	9,84 8,96 // 2,07 0,27 // 1,91 9,6 // 8 1,53 // 0,65 3,26	868	70140
max_div_30_3.txt	30	3	2	13,073737144470215	8,06 9,59 0,27 // 6,71 0,35 9,42	2	464
max_div_30_3.txt	30	3	3	34,29052925109863	8,32 0,47 8,98 // 8,06 9,59 0,27 // 1,53 8,09 9,56	8	1831
max_div_30_3.txt	30	3	4	63,70196056365967	8,32 0,47 8,98 // 6,75 0,07 1,8 // 8,06 9,59 0,27 // 1,53 8,09 9,56	56	9533
max_div_30_3.txt	30	3	5	99,59204149246216	8,32 0,47 8,98 // 6,75 0,07 1,8 // 7,54 8,93 9,66 // 8,06 9,59 0,27 // 1,53 8,09 9,56	518	52363

4. Branch and Bound Greedy por cota más ajustada:

Problema	n elements	K size	m max elements	z diversity	Solution	Time	Nodes
max_div_15_2.txt	15	2	2	11,85921573638916	8,65 9,98 // 0,58 1,29	2	119
max_div_15_2.txt	15	2	3	27,37270021438598 6	9,11 3,23 // 0,58 1,29 // 8,65 9,98	0	456
max_div_15_2.txt	15	2	4	49,82678127288818 4	9,11 3,23 // 0,16 4,62 // 0,58 1,29 // 8,65 9,98	2	1615
max_div_15_2.txt	15	2	5	79,12953066825867	9,11 3,23 // 9,96 8,17 // 0,16 4,62 // 0,58 1,29 // 8,65 9,98	6	3904
max_div_15_3.txt	15	3	2	13,27324008941650 4	9,88 9,88 6,26 // 0,3 0,92 4,23	0	119
max_div_15_3.txt	15	3	3	31,86852550506591 8	1,71 1,95 9,22 // 5,17 1,39 0,91 // 9,88 9,88 6,26	0	484
max_div_15_3.txt	15	3	4	59,76375770568848	9,88 9,88 6,26 // 0,3 0,92 4,23 // 1,71 1,95 9,22 // 6,65 9,45 1,02	2	1363
max_div_15_3.txt	15	3	5	96,08583307266235	1,15 9,21 3,11 // 1,71 1,95 9,22 // 0,3 0,92 4,23 // 9,88 9,88 6,26 // 9,47 3,56 1,83	5	3322
max_div_20_2.txt	20	2	2	8,510329246520996	0,63 5,96 // 8,67 3,17	0	209
max_div_20_2.txt	20	2	3	21,99608564376831	0,63 5,96 // 8,67 3,17 // 8,57 8,36	1	695
max_div_20_2.txt	20	2	4	40,00225687026977 5	3,47 9,43 // 1,16 4,47 // 8,57 8,36 // 8,67 3,17	8	3536

max_div_20_2.txt	20	2	5	63,65167927742004	3,47 9,43 // 8,57 8,36 // 1,97 3,5 // 0,63 5,96 // 8,67 3,17	30	11756
max_div_20_3.txt	20	3	2	11,80030918121337	9,81 2,05 1,83 // 0,65 3,76 9,07	0	209
max_div_20_3.txt	20	3	3	30,87265777587890	9,81 2,05 1,83 // 0,65 3,76 9,07 // 8,87 9,56 5,34	1	668
max_div_20_3.txt	20	3	4	56,69031047821045	0,83 7,06 3,34 // 9,81 2,05 1,83 // 0,65 3,76 9,07 // 9,6 2,02 9	7	3135
max_div_20_3.txt	20	3	5	92,82974338531494	9,81 2,05 1,83 // 0,65 3,76 9,07 // 8,87 9,56 5,34 // 0,83 7,06 3,34 // 9,6 2,02 9	24	8864
max_div_30_2.txt	30	2	2	11,65714359283447	9,84 8,96 // 2,07 0,27	2	464
max_div_30_2.txt	30	2	3	28,94429969787597	9,84 8,96 // 2,07 0,27 // 1,91 9,6	11	2464
max_div_30_2.txt	30	2	4	52,77117156982422	9,84 8,96 // 2,07 0,27 // 1,91 9,6 // 8 1,53	70	15023
max_div_30_2.txt	30	2	5	80,91024112701416	9,84 8,96 // 2,07 0,27 // 1,91 9,6 // 8 1,53 // 0,65 3,26	454	87145
max_div_30_3.txt	30	3	2	13,07373714447021	8,06 9,59 0,27 // 6,71 0,35 9,42	2	464
max_div_30_3.txt	30	3	3	34,29052925109863	8,32 0,47 8,98 // 8,06 9,59 0,27 // 1,53 8,09 9,56	12	2135

max_div_30_3.txt	30	3	4	63,70196056365967	8,32 0,47 8,98 // 6,75 0,07 1,8 // 8,06 9,59 0,27 // 1,53 8,09 9,56	61	12719
max_div_30_3.txt	30	3	5	99,59204149246216	8,32 0,47 8,98 // 6,75 0,07 1,8 // 7,54 8,93 9,66 // 8,06 9,59 0,27 // 1,53 8,09 9,56	362	71422

5. Branch and Bound Grasp en profundidad:

Problema	n elements	K size	m max elements	z diversity	Solution	Time	Nodes
max_div_15_2.txt	15	2	2	11,85921573638916	0,58 1,29 // 8,65 9,98	2	119
max_div_15_2.txt	15	2	3	27,37270021438598 6	9,11 3,23 // 0,58 1,29 // 8,65 9,98	0	413
max_div_15_2.txt	15	2	4	49,82678127288818 4	9,11 3,23 // 0,16 4,62 // 0,58 1,29 // 8,65 9,98	3	1559
max_div_15_2.txt	15	2	5	79,12953066825867	9,11 3,23 // 9,96 8,17 // 0,16 4,62 // 0,58 1,29 // 8,65 9,98	11	3808
max_div_15_3.txt	15	3	2	13,27324008941650 4	9,88 9,88 6,26 // 0,3 0,92 4,23	0	119
max_div_15_3.txt	15	3	3	31,86852550506591 8	1,71 1,95 9,22 // 9,88 9,88 6,26 // 5,17 1,39 0,91	0	408
max_div_15_3.txt	15	3	4	59,76375770568848	9,88 9,88 6,26 // 0,3 0,92 4,23 // 6,65 9,45 1,02 // 1,71 1,95 9,22	2	1232
max_div_15_3.txt	15	3	5	96,08583307266235	1,15 9,21 3,11 // 1,71 1,95 9,22 // 0,3 0,92 4,23 // 9,88 9,88 6,26 // 9,47 3,56 1,83	9	3195
max_div_20_2.txt	20	2	2	8,510329246520996	0,63 5,96 // 8,67 3,17	0	209
max_div_20_2.txt	20	2	3	21,99608564376831	8,67 3,17 // 0,63 5,96 // 8,57 8,36	1	580

max_div_20_2.txt	20	2	4	40,00225687026977 5	3,47 9,43 // 1,16 4,47 // 8,57 8,36 // 8,67 3,17	8	3051
max_div_20_2.txt	20	2	5	63,65167927742004 4	3,47 9,43 // 8,57 8,36 // 1,97 3,5 // 0,63 5,96 // 8,67 3,17	44	9405
max_div_20_3.txt	20	3	2	11,80030918121337 9	0,65 3,76 9,07 // 9,81 2,05 1,83	0	209
max_div_20_3.txt	20	3	3	30,87265777587890 6	8,87 9,56 5,34 // 9,81 2,05 1,83 // 0,65 3,76 9,07	1	613
max_div_20_3.txt	20	3	4	56,69031047821045	0,83 7,06 3,34 // 9,81 2,05 1,83 // 0,65 3,76 9,07 // 9,6 2,02 9	9	2508
max_div_20_3.txt	20	3	5	92,82974338531494	0,83 7,06 3,34 // 8,87 9,56 5,34 // 9,81 2,05 1,83 // 0,65 3,76 9,07 // 9,6 2,02 9	31	7707
max_div_30_2.txt	30	2	2	11,65714359283447 3	2,07 0,27 // 9,84 8,96	2	464
max_div_30_2.txt	30	2	3	28,94429969787597 7	2,07 0,27 // 9,84 8,96 // 1,91 9,6	10	2207
max_div_30_2.txt	30	2	4	52,77117156982422	1,91 9,6 // 2,07 0,27 // 9,84 8,96 // 8 1,53	67	12174
max_div_30_2.txt	30	2	5	80,91024112701416	1,91 9,6 // 9,84 8,96 // 8 1,53 // 0,65 3,26 // 2,07 0,27	1086	73570
max_div_30_3.txt	30	3	2	13,07373714447021 5	6,71 0,35 9,42 // 8,06 9,59 0,27	2	464
max_div_30_3.txt	30	3	3	34,29052925109863	8,06 9,59 0,27 // 8,32 0,47 8,98 // 1,53 8,09 9,56	7	1707
max_div_30_3.txt	30	3	4	63,70196056365967	8,32 0,47 8,98 // 6,75 0,07 1,8 // 8,06 9,59 0,27 // 1,53 8,09 9,56	54	10154
max_div_30_3.txt	30	3	5	99,59204149246216	1,53 8,09 9,56 // 6,75 0,07 1,8 // 7,54 8,93 9,66 // 8,06 9,59 0,27 // 8,32 0,47 8,98	509	52082

6. Branch and Bound Grasp por cota más ajustada:

Problema	n elements	K size	m max elements	z diversity	Solution	Time	Nodes
max_div_15_2.txt	15	2	2	11,85921573638916	8,65 9,98 // 0,58 1,29	2	119
max_div_15_2.txt	15	2	3	27,372700214385986	9,11 3,23 // 0,58 1,29 // 8,65 9,98	0	456
max_div_15_2.txt	15	2	4	49,826781272888184	9,11 3,23 // 0,16 4,62 // 0,58 1,29 // 8,65 9,98	2	1606
max_div_15_2.txt	15	2	5	79,12953066825867	9,11 3,23 // 9,96 8,17 // 0,16 4,62 // 0,58 1,29 // 8,65 9,98	8	3904
max_div_15_3.txt	15	3	2	13,273240089416504	0,3 0,92 4,23 // 9,88 9,88 6,26	0	119
max_div_15_3.txt	15	3	3	31,868525505065918	9,88 9,88 6,26 // 1,71 1,95 9,22 // 5,17 1,39 0,91	0	439
max_div_15_3.txt	15	3	4	59,76375770568848	1,71 1,95 9,22 // 6,65 9,45 1,02 // 9,88 9,88 6,26 // 0,3 0,92 4,23	2	1363
max_div_15_3.txt	15	3	5	96,08583307266235	1,15 9,21 3,11 // 1,71 1,95 9,22 // 0,3 0,92 4,23 // 9,88 9,88 6,26 // 9,47 3,56 1,83	5	3534
max_div_20_2.txt	20	2	2	8,510329246520996	0,63 5,96 // 8,67 3,17	0	209
max_div_20_2.txt	20	2	3	21,99608564376831	0,63 5,96 // 8,67 3,17 // 8,57 8,36	1	695
max_div_20_2.txt	20	2	4	40,002256870269775	3,47 9,43 // 1,16 4,47 // 8,57 8,36 // 8,67 3,17	9	3532
max_div_20_2.txt	20	2	5	63,651679277420044	3,47 9,43 // 8,57 8,36 // 1,97 3,5 // 0,63 5,96 // 8,67 3,17	29	11540
max_div_20_3.txt	20	3	2	11,800309181213379	9,81 2,05 1,83 // 0,65 3,76 9,07	0	209
max_div_20_3.txt	20	3	3	30,872657775878906	0,65 3,76 9,07 // 8,87 9,56 5,34 // 9,81 2,05 1,83	1	668
max_div_20_3.txt	20	3	4	56,69031047821045	0,83 7,06 3,34 // 9,81 2,05 1,83 // 0,65 3,76 9,07 // 9,6 2,02 9	7	3297
max_div_20_3.txt	20	3	5	92,82974338531494	0,83 7,06 3,34 // 8,87 9,56 5,34 // 9,81 2,05 1,83 // 0,65 3,76 9,07 // 9,6 2,02 9	27	9639
max_div_30_2.txt	30	2	2	11,657143592834473	2,07 0,27 // 9,84 8,96	2	464

max_div_30_2.txt	30	2	3	28,944299697875977	1,91 9,6 // 9,84 8,96 // 2,07 0,27	11	2512
max_div_30_2.txt	30	2	4	52,77117156982422	2,07 0,27 // 9,84 8,96 // 1,91 9,6 // 8 1,53	70	15023
max_div_30_2.txt	30	2	5	80,91024112701416	1,91 9,6 // 9,84 8,96 // 8 1,53 // 0,65 3,26 // 2,07 0,27	466	89128
max_div_30_3.txt	30	3	2	13,073737144470215	8,06 9,59 0,27 // 6,71 0,35 9,42	2	464
max_div_30_3.txt	30	3	3	34,29052925109863	8,32 0,47 8,98 // 8,06 9,59 0,27 // 1,53 8,09 9,56	10	2299
max_div_30_3.txt	30	3	4	63,70196056365967	1,53 8,09 9,56 // 6,75 0,07 1,8 // 8,06 9,59 0,27 // 8,32 0,47 8,98	60	12616
max_div_30_3.txt	30	3	5	99,59204149246216	8,32 0,47 8,98 // 6,75 0,07 1,8 // 7,54 8,93 9,66 // 8,06 9,59 0,27 // 1,53 8,09 9,56	377	74094