

Práctica 8: Ramificación y poda

Diseño y Análisis de Algoritmos

Helena García Díaz

07 de mayo de 2023

1. Introducción	2
2. Metodología	3
2.1. Algoritmos	3
2.1.1. Constructivo voraz.	3
2.1.2. Búsqueda local.	3
2.1.3. GRASP (Greedy Randomized Adaptive Search).	4
2.1.4. Ramificación y poda.	4
2.2. Estructura de la solución	6
2.3. Código	6
3. Resultados	8
3.1.1. Constructivo voraz.	8
3.1.2. Búsqueda local.	8
3.1.3. GRASP (Greedy Randomized Adaptive Search).	9
3.1.4. Ramificación y poda. Partiendo del algoritmo Voraz.	10
3.1.4. Ramificación y poda. Partiendo de GRASP.	10
4. Discusión	11
5. Conclusiones	12
6. Bibliografía	12
Anexos	13
A. Tabla de tiempo mínimo de procesamiento para cada instancia. Algoritmo constructivo local.	13
B. Tabla de tiempo mínimo de procesamiento para cada instancia. Búsqueda local.	15
C. Tabla de tiempo mínimo de procesamiento para cada instancia. Algoritmo GRASP.	17
D. Tabla de tiempo mínimo de procesamiento para cada instancia. Algoritmo de Ramificación y poda. Partiendo del algoritmo Voraz.	27
E. Tabla de tiempo mínimo de procesamiento para cada instancia. Algoritmo de Ramificación y poda. Partiendo de GRASP con 10 iteraciones.	32
F. Tabla de tiempo mínimo de procesamiento para cada instancia. Algoritmo de Ramificación y poda. Partiendo de GRASP con 500 iteraciones.	38
G. Representaciones gráficas en función del número de puntos, del tiempo de CPU y valor de z. Comparación algoritmo voraz y constructivo local.	43

1. Introducción

Durante esta práctica se tratará el problema de la Máxima diversidad. Este problema consiste en encontrar el subconjunto de diversidad máxima de un conjunto dado de elementos (*cluster* de puntos).

Nuestro problema consistirá en maximizar las distancias entre los puntos que formen parte de la solución. De manera formal, se describe de la siguiente manera:

Sea dado un conjunto $S = \{s_1, s_2, \dots, s_n\}$ de n elementos, en el que cada elemento s_i es un vector $s_i = (s_{i1}, s_{i2}, \dots, s_{iK})$. Sea, asimismo, d_{ij} la distancia entre los elementos i y j . Si $m < n$ es el tamaño del subconjunto que se busca, el problema puede formularse como:

$$\text{Maximizar } z = \sum_{i=1}^{n-1} \sum_{j=i+1}^n d_{ij} x_i x_j$$

sujeto a:

$$\sum_{i=1}^n x_i = m \quad x_i \in \{0, 1\} \quad i = 1, 2, \dots, n$$

donde:

$x_i = 1$ si s_i pertenece a la solución

$x_i = 0$ si s_i no pertenece a la solución

La distancia d_{ij} se calculará como la distancia euclídea:

$$d_{ij} = d(s_i, s_j) = \sqrt{\sum_{r=1}^K (s_{ir} - s_{jr})^2}$$

2. Metodología

La inclusión de puntos en la solución se ha implementado con diferentes estrategias con el objetivo de estudiar y comparar su eficiencia y eficacia en la calidad de las soluciones.

Para ello se han implementado los siguientes algoritmos:

- Constructivo voraz.
- Búsqueda local.
- GRASP (*Greedy Randomized Adaptative Search*).
- Ramificación y poda.

2.1. Algoritmos

2.1.1. Constructivo voraz.

En el algoritmo constructivo voraz, se le pasa una instancia del problema, la cantidad de puntos que deberá tener la solución y el tamaño de la *LRC* (Lista restringida de candidatos). En este caso la *LRC* será de tamaño 1. El motivo de ello se explicará a continuación.

Primero, se calcula el centroide de la nube total de puntos disponibles (es decir, todos los de la instancia del problema).

A continuación, se añade a la solución el punto más alejado del centroide. Como nuestra *LRC* tiene tamaño 1, elegirá siempre el más alejado.

Ahora, mientras no se alcance el número de puntos en la solución, se calcula el centroide de los puntos de la solución y se elige, de entre los disponibles (los que aún no forman parte de la solución), el punto más alejado a dicho centroide.

De esta forma, se construye una solución inicial para nuestro problema.

2.1.2. Búsqueda local.

Para la búsqueda local, se le pasa una instancia del problema, el número total de puntos y el tamaño que debe tener la *LRC*, que en este caso será también 1.

A continuación, se ejecuta el algoritmo constructivo anteriormente mencionado y se calcula el valor de su función

objetivo. El motivo de ello es luego calcular los siguientes valores a partir de este inicial.

Luego se satura la búsqueda local con el movimiento de intercambio y se calcula el valor de la función objetivo a partir del inicial que calculamos.

Para realizar la búsqueda local, ya que la solución tiene un tamaño fijo, he realizado el movimiento de intercambio.

Primero calculo el mejor valor de la función objetivo hasta el momento. A continuación, si no queda entorno por generar, se para, en caso contrario, se busca el mejor de los vecinos con el movimiento de intercambio. Para la búsqueda de este mejor vecino, se exploran los vecinos de la solución actual, evaluándose parcialmente a partir del valor de z (función objetivo) original.

Una vez tenemos el mejor de los vecinos, se evalúa parcialmente la solución, y como estamos maximizando, si el valor nuevo es mejor que el que teníamos almacenado, se continúa explorando el entorno. Si no, se para.

De esta forma, al final tendremos un óptimo local.

2.1.3. GRASP (Greedy Randomized Adaptive Search).

GRASP es similar a como se ejecuta la búsqueda local, pero en lugar de elegir el punto más alejado al centroide, se tiene un tamaño variable de la LRC y se elige aleatoriamente entre uno de los puntos de la misma. En este caso, se ha probado con tamaños 2 y 3 para la LRC.

2.1.4. Ramificación y poda.

El objetivo de este algoritmo es encontrar la mejor solución posible mediante la exploración de todas las soluciones posibles, pero de manera más eficiente que una búsqueda exhaustiva.

En la ramificación y poda se parte de una solución inicial, que se habrá generado con el constructivo voraz o con GRASP, y se va expandiendo un árbol de posibles soluciones, de tal forma que se van generando nuevas soluciones que son mejores que las anteriores (en términos de la función objetivo del problema). Además, se utilizan cotas superiores e inferiores para acotar el espacio de búsqueda.

A partir de la solución inicial, se establece una cota inferior para el valor de z .

Con ese valor inicial, se crea el primer nodo en el árbol. Cada nodo contendrá una solución y su profundidad en el árbol. Además, utilizo un vector de *activeNodes_* que contiene los nodos que aún no han sido expandidos y que deben ser evaluados.

Teniendo en cuenta que se trata de un problema de máximo, se trabaja de la siguiente manera en el algoritmo:

1. Mientras haya nodos en el vector de *activeNodes_*, se selecciona el nodo con la mayor cota superior (mejor límite superior).
2. Se expande el nodo seleccionado, creando nuevos nodos que se agregan al vector de *activeNodes_*.
3. Se calcula la cota inferior del nodo seleccionado y se actualiza la mejor cota inferior encontrada hasta el momento.
4. Se realiza la poda, eliminando los nodos cuya cota superior sea menor o igual a la mejor cota inferior encontrada hasta el momento.
5. Se repiten los pasos anteriores hasta que no haya más nodos en el vector de *activeNodes_*.

Finalmente, se devuelve la mejor solución encontrada (la cota inferior).

A la hora de expandir el nodo actual, se hace mediante el método *generateLeaf()*. A este método se le pasa una instancia del problema, la cota inferior (instancia de *Solution<T>*), la profundidad del nodo actual y el tamaño que debe tener la solución.

Este método genera todos los nodos hoja (hijos) de un nodo padre. Cada hijo se genera avanzando por cada una de las ramas del árbol de soluciones, estableciendo elementos para obtener todas las posibles combinaciones.

En primer lugar, creo un vector de nodos hijos vacío llamado *children*. A continuación, se seleccionan los elementos correspondientes del nodo padre que ya han sido seleccionados en ramas anteriores, y se eliminan del conjunto de elementos iniciales *initialX* (inicializado al conjunto total de puntos de la instancia). Luego, se establece un eje de coordenadas, que representa el nuevo elemento que se agrega en esta rama

del árbol. El tamaño de la solución seleccionada se ajusta a la profundidad actual más uno, lo que significa que se agrega un elemento adicional.

A continuación, se genera un nuevo hijo para cada elemento restante en *initialX*. Se crea una nueva solución para este hijo usando la función *generateBestSolution()*, que selecciona los elementos de *initialX* que maximizan la función objetivo *z* de acuerdo con las restricciones establecidas en *selectedElement* y *m* (número de puntos en la solución). El nuevo hijo se crea como un nuevo nodo y se agrega al vector *children*.

Finalmente, se devuelve el vector de nodos hijos generado.

2.2. Estructura de la solución

Nuestra solución estará compuesta por los puntos obtenidos a partir de la instancia proporcionada.

Se ha empleado como base para esta práctica, la realizada anteriormente en la asignatura. Por ello, algunos nombres se han mantenido similares.

En el código, esta se verá representada por una clase *Solution<T>* que contará con un atributo *service_points_* constituido por un *cluster* donde cada *cluster* será un vector de vectores. De esta manera, se podrán almacenar puntos de diferentes dimensiones sin problemas.

En la misma clase *Solution*, se encuentran los métodos correspondientes para manejar la solución, como la evaluación de la función objetivo. Dicha función objetivo será la fórmula descrita inicialmente en la introducción. Es decir, la distancia euclídea entre los puntos que formen parte de la solución.

2.3. Código

En cuanto al código, se han desarrollado tres clases principales:

- *Problem*: Para almacenar toda la información sobre las instancias y trabajar con ellas. En ella se lee una instancia desde archivo, se almacena la nube de puntos y

la cantidad de los mismos. Así como diversos métodos para trabajar con los mismos.

- *Solution*: Para almacenar y trabajar con la solución, tal y como se ha mencionado anteriormente.
- *Algorithm*: Para implementar todos los algoritmos descritos como métodos. Se ha decidido trabajar de esta forma en lugar de a como se trabajó en la práctica anterior, debido a que algunos algoritmos son usados por otros en su implementación y de esta manera es más accesible.

Aparte de las clases principales, se ha implementado la clase *Node* para generar los nodos de la ramificación y poda.

Además, se han implementado dos clases para facilitar y manejar la realización de varias ejecuciones, pudiendo almacenar los resultados de las mismas en un archivo csv. Dichas clases son *Generador* e *Iteración*.

Además, se ha implementado un programa sencillo en *python* para representar gráficamente el valor de la función objetivo y el tiempo de ejecución de algunos algoritmos con el objetivo de poder realizar comparaciones de manera más efectiva.

Por último, todo está manejado por un *makefile* que permite ejecutar los tests creados para controlar las fórmulas de referencia empleadas y ejecutar, compilando antes o no, el proyecto.

3. Resultados

Para hacer un estudio sobre los resultados obtenidos, se han tenido en cuenta principalmente el valor de la función objetivo y el tiempo de ejecución del algoritmo.

3.1.1. Constructivo voraz.

Si nos fijamos en una de las imágenes del apéndice H, construida a partir de los datos del apéndice A, podemos observar que el tiempo de ejecución no cambia de manera significativa en función del número de puntos que se introduzcan en la solución dentro del mismo problema.

Del mismo modo, dentro de cada problema se observa una relación directa entre el valor de la función objetivo y el número de puntos dentro de la misma. Esto era de esperar, ya que cuantos más puntos haya en la misma, aumentará el número de distancias a calcular entre los mismos, y con ello el valor de nuestra función objetivo.

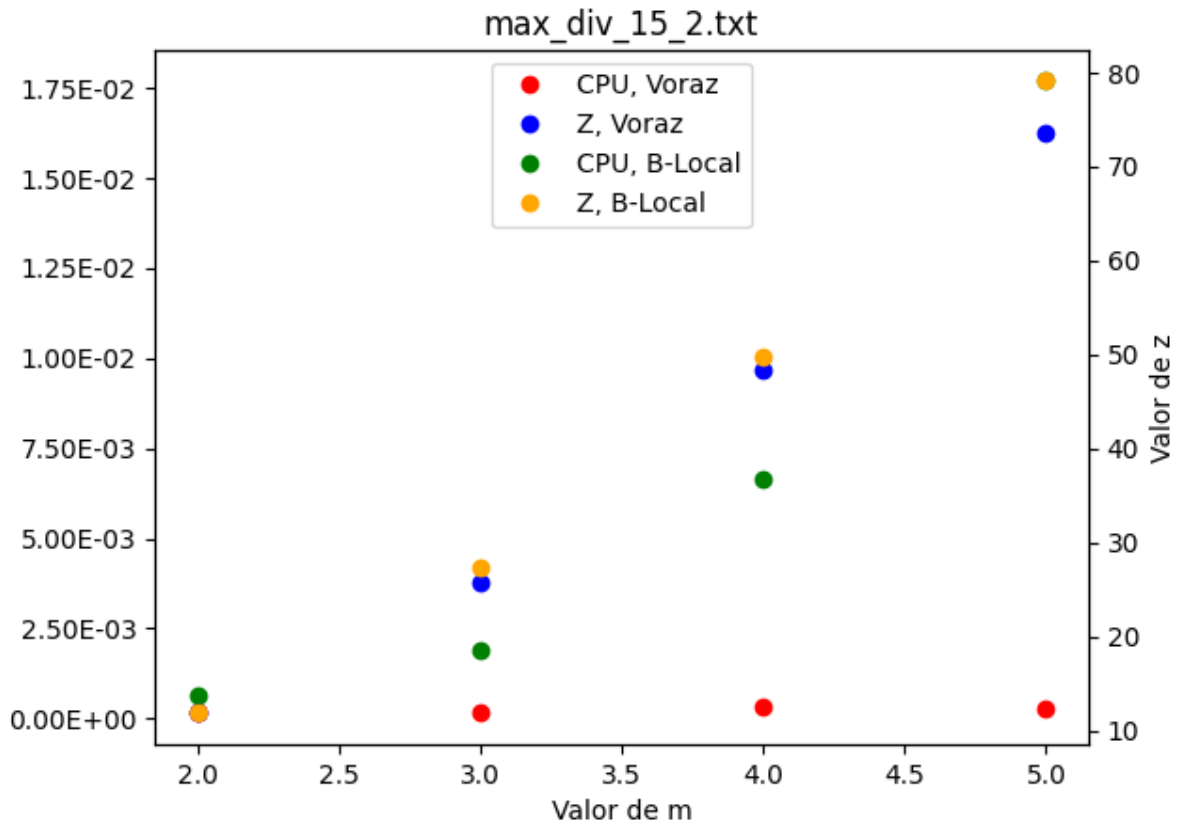
3.1.2. Búsqueda local.

Si estudiamos los datos obtenidos en el apéndice B, podemos realizar una comparación con los datos del constructivo voraz (apéndice A), obteniendo las gráficas del apéndice H.

Si observamos una de ellas, vemos que el tiempo de CPU (eje vertical izquierdo) del algoritmo Voraz es significativamente menor que el de la búsqueda local. Esto es lógico, ya que en el último se está generando un entorno y buscando el mejor valor de z de entre todos los vecinos. De igual manera, el tiempo de CPU aumenta de manera exponencial en función del número de puntos que se quieran introducir en la solución.

En cuanto al valor de la función objetivo, vemos que de igual manera aumenta con la cantidad de puntos que introducimos en la solución.

Si comparamos estos valores entre ambos algoritmos, se observa que cuanto mayor es el número de puntos en la solución, mejor es el valor de z obtenido con la búsqueda local. Esto también era de esperar, ya que se está buscando en el entorno de la solución, el vecino que tenga mayor valor para z .



3.1.3. GRASP (Greedy Randomized Adaptive Search).

En la tabla obtenida en el apéndice C, se han incluido los valores de z obtenidos con el algoritmo constructivo inicial, para comprobar que efectivamente se “parte” de soluciones iniciales diferentes. Al ser las instancias de tamaños pequeños, GRASP acaba saturando con la búsqueda local al valor máximo de la función objetivo.

Si nos fijamos en el tamaño de la LRC, en las primeras instancias no parece haber una diferencia notable en el tiempo de ejecución de una con respecto a otras. Sin embargo, cuando empleamos instancias más grandes (mayor número de puntos), se puede observar que en algunas de las mismas, cuando la lista tiene tamaño 3, aumenta ligeramente el tiempo de ejecución, aunque ambos tamaños siguen siendo del mismo orden de magnitud. En cuanto al valor de la función objetivo, tal y como comentaba anteriormente, se alcanza en ambos casos el mismo resultado.

Si comparamos los resultados con la búsqueda local, se puede observar que en la mayoría de instancias se obtienen los mismo

valores de la función objetivo. Sin embargo, en el caso de la instancia `"max_div_30_2.txt"` con un número de puntos en la solución de 5, se obtiene un resultado para la función objetivo de 81,20 mientras que en la búsqueda local se obtenía un valor de 80.91. Se trata del único caso para el que *GRASP* obtuvo un resultado para la función objetivo mejor que la simple búsqueda local.

3.1.4. Ramificación y poda. Partiendo del algoritmo Voraz.

En este caso, se han incluido en la tabla (apéndice D) las dos estrategias seguidas a la hora de expandir el árbol.

Si nos fijamos en la calidad de las soluciones, en ambos casos se obtiene siempre el mismo resultado para la función objetivo. Sin embargo, si nos fijamos en el tiempo de ejecución, no parecen haber diferencias significativas entre ambos algoritmos, pero cuando las hay, la estrategia de profundidad obtiene tiempos mayores de ejecución.

En cuanto a la cantidad de nodos generados, tal y como era de esperar, se puede observar una tendencia a que esta cantidad sea mayor con la estrategia de profundidad (*Deeper* en la tabla).

Si comparamos los resultados con el algoritmo *GRASP*, se siguen obteniendo los mismos resultados en cuanto al valor de la función objetivo. Sin embargo, los tiempo de ejecución han aumentado ligeramente. Del orden de 10^{-4} a 10^{-3} por ejemplo.

3.1.4. Ramificación y poda. Partiendo de GRASP.

En este caso, se han generado dos tablas (apéndices E y F). En la primera, se ha partido de la mejor iteración con *GRASP*, con 10 iteraciones. En la segunda, con 500. En ambas, se ha empleado un tamaño de *LRC* de 3.

Si observamos los valores de la función objetivo, una vez más, se vuelven a obtener los mismos resultados descritos en los apartados anteriores. Si observamos los tiempos de ejecución, siguen siendo muy similares.

En cuanto al número de nodos expandidos, en ambos casos se suele obtener el mismo resultado, salvo para contadas instancias donde se parte de 10 iteraciones, como `"max_div_20_2.txt"` y `"max_div_15_2"` con 5 puntos en la

solución, donde se expanden un mayor número de nodos en la estrategia de expandir el menor z (*SmallestZ*). Esto puede deberse a cómo se poda el árbol de nodos.

Si comparamos los resultados con el algoritmo anterior, donde se parte del constructivo voraz, se ve una diferencia notable en el número de nodos expandidos, siendo el actual siempre menor o igual que el anterior.

En cuanto a los tiempos de ejecución, si no son similares son, generalmente, ligeramente inferiores.

4. Discusión

Una vez estudiados todos los resultados de los algoritmos, comparando a su vez el mismo con el anterior, se puede determinar que, para el tamaño de instancias con el que trabajamos, se alcanzan los mismos resultados para la función objetivo iterando *GRASP* cierto número de veces y eligiendo la mejor iteración en términos de z y tiempo de ejecución. En cuanto al número de iteraciones de *GRASP*, la diferencia entre 10 y 20 iteraciones no es significativamente notable, por lo que con 10 de las mismas sería suficiente.

Si tuviera que elegir entre *Branch and Bound*, partiendo de *GRASP* con 10 o 500 iteraciones, para el tamaño de instancias con el que hemos estado trabajando, no existen una diferencia significativa más allá de un aumento en el tiempo de ejecución, no del algoritmo en sí, sino de la fase previa de realizar las iteraciones. Dado la diferencia posterior en calidad y tiempo de ejecución, no merece la pena esperar tanto tiempo.

5. Conclusiones

En conclusión, el problema de la máxima diversidad ha sido objeto de estudio por parte de la comunidad científica durante mucho tiempo. Los investigadores han trabajado en la implementación de diferentes algoritmos para resolverlo, cada uno con sus propias características y enfoques. Sin embargo, no todos los algoritmos son igualmente efectivos y eficientes en la solución del problema.

En este sentido, se ha comprobado que el algoritmo *GRASP* ejecutado cierto número de veces es el más adecuado para resolver el problema de la máxima diversidad con las instancias facilitadas en esta práctica. *GRASP* ha demostrado ser ligeramente superior a otros algoritmos en términos de relación entre calidad de las soluciones obtenidas y eficiencia computacional.

Además, es importante destacar que el algoritmo *GRASP* no solo se ha aplicado al problema de la máxima diversidad, sino que ha demostrado ser una técnica efectiva en la solución de otros problemas de optimización combinatoria. Por lo tanto, se puede considerar que *GRASP* es una herramienta valiosa y ampliamente utilizada en la investigación de la optimización combinatoria.

Es importante destacar que se han evaluado diferentes enfoques y se han comparado sus resultados, lo que proporciona una evaluación completa y exhaustiva de las estrategias utilizadas para resolver el problema de la máxima diversidad en este contexto.

6. Bibliografía

- [1] Guión de la práctica e instancias:
<https://campusingenieriaytecnologia2223.ull.es/mod/folder/view.php?id=28667>
- [2] Apuntes sobre la Ramificación y Poda:
https://campusingenieriaytecnologia2223.ull.es/pluginfile.php/34313/mod_resource/content/1/Ramificaci%C3%B3nPoda.pdf
- [3] Artículo sobre Ramificación y Poda en el problema de la máxima diversidad:
https://campusingenieriaytecnologia2223.ull.es/pluginfile.php/34285/mod_resource/content/1/Branch_and_Bound_MDP.pdf

Anexos

A. Tabla de tiempo mínimo de procesamiento para cada instancia. Algoritmo constructivo local.

Problema	n	K	m	z	S	CPU
max_div_15_3.txt	15	3	2	13,27	(9.88 9.88 6.26) (0.30 0.92 4.23)	2,41E-04
max_div_15_3.txt	15	3	3	30,32	(9.88 9.88 6.26) (0.30 0.92 4.23) (1.71 1.95 9.22)	1,69E-04
max_div_15_3.txt	15	3	4	59,76	(9.88 9.88 6.26) (0.30 0.92 4.23) (1.71 1.95 9.22) (6.65 9.45 1.02)	2,30E-04
max_div_15_3.txt	15	3	5	94,75	(9.88 9.88 6.26) (0.30 0.92 4.23) (1.71 1.95 9.22) (6.65 9.45 1.02) (9.47 3.56 1.83)	3,32E-04
max_div_30_2.txt	30	2	2	11,66	(9.84 8.96) (2.07 0.27)	2,95E-04
max_div_30_2.txt	30	2	3	28,94	(9.84 8.96) (2.07 0.27) (1.91 9.60)	3,33E-04
max_div_30_2.txt	30	2	4	52,77	(9.84 8.96) (2.07 0.27) (1.91 9.60) (8.00 1.53)	5,25E-04
max_div_30_2.txt	30	2	5	80,91	(9.84 8.96) (2.07 0.27) (1.91 9.60) (8.00 1.53) (0.65 3.26)	6,06E-04
max_div_20_3.txt	20	3	2	11,80	(9.81 2.05 1.83) (0.65 3.76 9.07)	2,63E-04
max_div_20_3.txt	20	3	3	30,87	(9.81 2.05 1.83) (0.65 3.76 9.07) (8.87 9.56 5.34)	3,53E-04
max_div_20_3.txt	20	3	4	56,53	(9.81 2.05 1.83) (0.65 3.76 9.07) (8.87 9.56 5.34) (0.83 7.06 3.34)	4,42E-04
max_div_20_3.txt	20	3	5	92,83	(9.81 2.05 1.83) (0.65 3.76 9.07) (8.87 9.56 5.34) (0.83 7.06 3.34) (9.60 2.02 9.00)	4,05E-04

max_div_20_2.txt	20	2	2	8,51	(0.63 5.96) (8.67 3.17)	1,91E-04
max_div_20_2.txt	20	2	3	22,00	(0.63 5.96) (8.67 3.17) (8.57 8.36)	2,60E-04
max_div_20_2.txt	20	2	4	39,57	(0.63 5.96) (8.67 3.17) (8.57 8.36) (1.16 4.47)	4,05E-04
max_div_20_2.txt	20	2	5	61,24	(0.63 5.96) (8.67 3.17) (8.57 8.36) (1.16 4.47) (8.78 7.43)	4,95E-04
max_div_15_2.txt	15	2	2	11,86	(8.65 9.98) (0.58 1.29)	1,50E-04
max_div_15_2.txt	15	2	3	25,73	(8.65 9.98) (0.58 1.29) (9.96 8.17)	1,86E-04
max_div_15_2.txt	15	2	4	48,41	(8.65 9.98) (0.58 1.29) (9.96 8.17) (1.59 1.57)	3,39E-04
max_div_15_2.txt	15	2	5	73,56	(8.65 9.98) (0.58 1.29) (9.96 8.17) (1.59 1.57) (8.41 9.98)	3,01E-04
max_div_30_3.txt	30	3	2	13,07	(8.06 9.59 0.27) (6.71 0.35 9.42)	2,55E-04
max_div_30_3.txt	30	3	3	33,84	(8.06 9.59 0.27) (6.71 0.35 9.42) (1.53 8.09 9.56)	3,65E-04
max_div_30_3.txt	30	3	4	63,52	(8.06 9.59 0.27) (6.71 0.35 9.42) (1.53 8.09 9.56) (6.75 0.07 1.80)	7,20E-04
max_div_30_3.txt	30	3	5	99,51	(8.06 9.59 0.27) (6.71 0.35 9.42) (1.53 8.09 9.56) (6.75 0.07 1.80) (7.54 8.93 9.66)	6,76E-04

B. Tabla de tiempo mínimo de procesamiento para cada instancia. Búsqueda local.

Problema	n	K	m	z	S	CPU
max_div_15_3.txt	15	3	2	13,27	(9,88 9,88 6,26) (0,30 0,92 4,23)	8,80E-04
max_div_15_3.txt	15	3	3	31,87	(9,88 9,88 6,26) (5,17 1,39 0,91) (1,71 1,95 9,22)	1,92E-03
max_div_15_3.txt	15	3	4	59,76	(9,88 9,88 6,26) (0,30 0,92 4,23) (1,71 1,95 9,22) (6,65 9,45 1,02)	2,01E-03
max_div_15_3.txt	15	3	5	96,09	(9,88 9,88 6,26) (0,30 0,92 4,23) (1,71 1,95 9,22) (1,15 9,21 3,11) (9,47 3,56 1,83)	5,60E-03
max_div_30_2.txt	30	2	2	11,66	(9,84 8,96) (2,07 0,27)	1,07E-03
max_div_30_2.txt	30	2	3	28,94	(9,84 8,96) (2,07 0,27) (1,91 9,60)	2,26E-03
max_div_30_2.txt	30	2	4	52,77	(9,84 8,96) (2,07 0,27) (1,91 9,60) (8,00 1,53)	4,15E-03
max_div_30_2.txt	30	2	5	80,91	(9,84 8,96) (2,07 0,27) (1,91 9,60) (8,00 1,53) (0,65 3,26)	5,92E-03
max_div_20_3.txt	20	3	2	11,80	(9,81 2,05 1,83) (0,65 3,76 9,07)	1,00E-03
max_div_20_3.txt	20	3	3	30,87	(9,81 2,05 1,83) (0,65 3,76 9,07) (8,87 9,56 5,34)	1,31E-03
max_div_20_3.txt	20	3	4	56,69	(9,81 2,05 1,83) (0,65 3,76 9,07) (9,60 2,02 9,00) (0,83 7,06 3,34)	3,88E-03
max_div_20_3.txt	20	3	5	92,83	(9,81 2,05 1,83) (0,65 3,76 9,07) (8,87 9,56 5,34) (0,83 7,06 3,34) (9,60 2,02 9,00)	3,55E-03

max_div_20_2.txt	20	2	2	8,51	(0,63 5,96) (8,67 3,17)	7,13E-04
max_div_20_2.txt	20	2	3	22,00	(0,63 5,96) (8,67 3,17) (8,57 8,36)	1,55E-03
max_div_20_2.txt	20	2	4	40,00	(3,47 9,43) (8,67 3,17) (8,57 8,36) (1,16 4,47)	4,58E-03
max_div_20_2.txt	20	2	5	63,65	(0,63 5,96) (8,67 3,17) (8,57 8,36) (1,97 3,50) (3,47 9,43)	1,27E-02
max_div_15_2.txt	15	2	2	11,86	(8,65 9,98) (0,58 1,29)	6,23E-04
max_div_15_2.txt	15	2	3	27,37	(8,65 9,98) (0,58 1,29) (9,11 3,23)	1,88E-03
max_div_15_2.txt	15	2	4	49,83	(8,65 9,98) (0,58 1,29) (9,11 3,23) (0,16 4,62)	6,67E-03
max_div_15_2.txt	15	2	5	79,13	(8,65 9,98) (0,16 4,62) (9,11 3,23) (0,58 1,29) (9,96 8,17)	1,77E-02
max_div_30_3.txt	30	3	2	13,07	(8,06 9,59 0,27) (6,71 0,35 9,42)	1,17E-03
max_div_30_3.txt	30	3	3	34,29	(8,06 9,59 0,27) (8,32 0,47 8,98) (1,53 8,09 9,56)	4,34E-03
max_div_30_3.txt	30	3	4	63,70	(8,06 9,59 0,27) (8,32 0,47 8,98) (1,53 8,09 9,56) (6,75 0,07 1,80)	6,30E-03
max_div_30_3.txt	30	3	5	99,59	(8,06 9,59 0,27) (8,32 0,47 8,98) (1,53 8,09 9,56) (6,75 0,07 1,80) (7,54 8,93 9,66)	1,26E-02

**C. Tabla de tiempo mínimo de procesamiento para cada instancia.
Algoritmo GRASP.**

Problema	n	K	m	Iter	LRC	z0	z	S	CPU
max_div_15_3.txt	15	3	2	10	2	11,76	13,27	(0,30 0,92 4,23) (9,88 9,88 6,26)	8,40E-04
max_div_15_3.txt	15	3	2	10	3	11,76	13,27	(0,30 0,92 4,23) (9,88 9,88 6,26)	7,02E-04
max_div_15_3.txt	15	3	2	20	2	11,76	13,27	(9,88 9,88 6,26) (0,30 0,92 4,23)	6,61E-04
max_div_15_3.txt	15	3	2	20	3	9,98	13,27	(9,88 9,88 6,26) (0,30 0,92 4,23)	9,67E-04
max_div_15_3.txt	15	3	3	10	2	30,32	31,87	(1,71 1,95 9,22) (9,88 9,88 6,26) (5,17 1,39 0,91)	1,42E-03
max_div_15_3.txt	15	3	3	10	3	30,10	31,87	(1,71 1,95 9,22) (9,88 9,88 6,26) (5,17 1,39 0,91)	1,41E-03
max_div_15_3.txt	15	3	3	20	2	30,32	31,87	(9,88 9,88 6,26) (5,17 1,39 0,91) (1,71 1,95 9,22)	1,63E-03
max_div_15_3.txt	15	3	3	20	3	31,87	31,87	(1,71 1,95 9,22) (9,88 9,88 6,26) (5,17 1,39 0,91)	1,02E-03
max_div_15_3.txt	15	3	4	10	2	59,76	59,76	(1,71 1,95 9,22) (6,65 9,45 1,02) (9,88 9,88 6,26) (0,30 0,92 4,23)	3,42E-03

max_div_15_3.txt	15	3	4	10	3	58,40	59,76	(1,71 9,22) 9,88 (6,65 1,02) 0,92	1,95 (9,88 6,26) 9,45 (0,30 4,23)	2,50E-03
max_div_15_3.txt	15	3	4	20	2	55,81	59,76	(1,71 9,22) 9,45 (9,88 6,26) 0,92	1,95 (6,65 1,02) 9,88 (0,30 4,23)	2,18E-03
max_div_15_3.txt	15	3	4	20	3	58,40	59,76	(1,71 9,22) 9,45 (9,88 6,26) 0,92	1,95 (6,65 1,02) 9,88 (0,30 4,23)	2,15E-03
max_div_15_3.txt	15	3	5	10	2	94,75	96,09	(1,71 9,22) 9,21 (9,88 6,26) 0,92 (9,47 1,83)	1,95 (1,15 3,11) 9,88 (0,30 4,23) 3,56	4,33E-03
max_div_15_3.txt	15	3	5	10	3	93,98	96,09	(9,88 6,26) 1,95 (9,47 1,83) 0,92 (1,15 3,11)	9,88 (1,71 9,22) 3,56 (0,30 4,23) 9,21	5,25E-03
max_div_15_3.txt	15	3	5	20	2	88,10	96,09	(9,88 6,26) 0,92 (1,15 3,11) 1,95 (9,47 1,83)	9,88 (0,30 4,23) 9,21 (1,71 9,22) 3,56	5,24E-03
max_div_15_3.txt	15	3	5	20	3	87,88	96,09	(9,88 6,26) 3,56 (0,30 4,23) 9,21	9,88 (9,47 1,83) 0,92 (1,15 3,11)	5,34E-03

								(1,71 1,95 9,22)	
max_div_30_2.txt	30	2	2	10	2	9,98	11,66	(2,07 0,27) (9,84 8,96)	8,07E-04
max_div_30_2.txt	30	2	2	10	3	11,66	11,66	(9,84 8,96) (2,07 0,27)	6,40E-04
max_div_30_2.txt	30	2	2	20	2	11,66	11,66	(9,84 8,96) (2,07 0,27)	6,91E-04
max_div_30_2.txt	30	2	2	20	3	10,81	11,66	(9,84 8,96) (2,07 0,27)	8,31E-04
max_div_30_2.txt	30	2	3	10	2	23,37	28,94	(2,07 0,27) (1,91 9,60) (9,84 8,96)	1,87E-03
max_div_30_2.txt	30	2	3	10	3	24,61	28,94	(2,07 0,27) (9,84 8,96) (1,91 9,60)	2,83E-03
max_div_30_2.txt	30	2	3	20	2	28,94	28,94	(2,07 0,27) (9,84 8,96) (1,91 9,60)	1,02E-03
max_div_30_2.txt	30	2	3	20	3	28,02	28,94	(2,07 0,27) (9,84 8,96) (1,91 9,60)	1,63E-03
max_div_30_2.txt	30	2	4	10	2	48,63	52,77	(2,07 0,27) (9,84 8,96) (1,91 9,60) (8,00 1,53)	3,06E-03
max_div_30_2.txt	30	2	4	10	3	46,78	52,77	(9,84 8,96) (2,07 0,27) (1,91 9,60) (8,00 1,53)	5,21E-03
max_div_30_2.txt	30	2	4	20	2	49,18	52,77	(2,07 0,27) (9,84 8,96) (1,91 9,60) (8,00 1,53)	4,37E-03
max_div_30_2.txt	30	2	4	20	3	42,33	52,77	(2,07 0,27) (1,91 9,60) (9,84 8,96) (8,00 1,53)	6,76E-03
max_div_30_2.txt	30	2	5	10	2	76,30	81,20	(2,07 0,27) (9,84 8,96) (1,91 9,60) (9,84 8,96) (2,07 0,27)	9,69E-03

max_div_30_2.txt	30	2	5	10	3	71,57	81,20	(1,91 9,60) (2,07 0,27) (2,07 0,27) (9,84 8,96) (9,84 8,96)	9,98E-03
max_div_30_2.txt	30	2	5	20	2	76,07	81,20	(9,84 8,96) (2,07 0,27) (1,91 9,60) (2,07 0,27) (9,84 8,96)	1,28E-02
max_div_30_2.txt	30	2	5	20	3	71,10	81,20	(1,91 9,60) (9,84 8,96) (2,07 0,27) (9,84 8,96) (2,07 0,27)	1,37E-02
max_div_20_3.txt	20	3	2	10	2	10,87	11,80	(0,65 3,76 9,07) (9,81 2,05 1,83)	8,15E-04
max_div_20_3.txt	20	3	2	10	3	10,73	11,80	(0,65 3,76 9,07) (9,81 2,05 1,83)	7,79E-04
max_div_20_3.txt	20	3	2	20	2	11,80	11,80	(9,81 2,05 1,83) (0,65 3,76 9,07)	5,55E-04
max_div_20_3.txt	20	3	2	20	3	10,73	11,80	(0,65 3,76 9,07) (9,81 2,05 1,83)	7,86E-04
max_div_20_3.txt	20	3	3	10	2	30,00	30,87	(0,65 3,76 9,07) (9,81 2,05 1,83) (8,87 9,56 5,34)	1,86E-03
max_div_20_3.txt	20	3	3	10	3	29,79	30,87	(0,65 3,76 9,07) (9,81 2,05 1,83) (8,87 9,56 5,34)	1,73E-03
max_div_20_3.txt	20	3	3	20	2	30,87	30,87	(0,65 3,76 9,07) (9,81 2,05 1,83) (8,87 9,56 5,34)	9,92E-04
max_div_20_3.txt	20	3	3	20	3	26,71	30,87	(0,65 3,76 9,07) (8,87 9,56 5,34) (9,81 2,05 1,83)	2,66E-03

max_div_20_3.txt	20	3	4	10	2	54,71	56,69	(0,65 9,07) 2,05 (9,60 9,00) 7,06 3,34)	3,76 (9,81 1,83) 2,02 (0,83	3,62E-03
max_div_20_3.txt	20	3	4	10	3	55,12	56,69	(0,65 9,07) 2,05 (9,60 9,00) 7,06 3,34)	3,76 (9,81 1,83) 2,02 (0,83	3,45E-03
max_div_20_3.txt	20	3	4	20	2	56,53	56,69	(0,65 9,07) 2,05 (9,60 9,00) 7,06 3,34)	3,76 (9,81 1,83) 2,02 (0,83	3,40E-03
max_div_20_3.txt	20	3	4	20	3	56,53	56,69	(0,65 9,07) 2,05 (9,60 9,00) 7,06 3,34)	3,76 (9,81 1,83) 2,02 (0,83	3,11E-03
max_div_20_3.txt	20	3	5	10	2	85,48	92,83	(9,81 1,83) 3,76 (8,87 5,34) 2,02 (0,83 3,34)	2,05 (0,65 9,07) 9,56 (9,60 9,00) 7,06	9,71E-03
max_div_20_3.txt	20	3	5	10	3	82,70	92,83	(9,81 1,83) 7,06 (8,87 5,34) 3,76 (9,60 9,00)	2,05 (0,83 3,34) 9,56 (0,65 9,07) 2,02	8,20E-03
max_div_20_3.txt	20	3	5	20	2	88,75	92,83	(0,65 9,07) 9,56 (9,60 9,00) 7,06 (9,81 1,83)	3,76 (8,87 5,34) 2,02 (0,83 3,34) 2,05	8,14E-03

								(0,65 3,76 9,07) (8,87 9,56 5,34) (9,81 2,05 1,83) (0,83 7,06 3,34) (9,60 2,02 9,00)	
max_div_20_3.txt	20	3	5	20	3	84,15	92,83		8,63E-03
max_div_20_2.txt	20	2	2	10	2	8,29	8,51	(0,63 5,96) (8,67 3,17)	8,59E-04
max_div_20_2.txt	20	2	2	10	3	8,01	8,51	(8,67 3,17) (0,63 5,96)	8,53E-04
max_div_20_2.txt	20	2	2	20	2	8,51	8,51	(0,63 5,96) (8,67 3,17)	5,61E-04
max_div_20_2.txt	20	2	2	20	3	8,28	8,51	(0,63 5,96) (8,67 3,17)	9,86E-04
max_div_20_2.txt	20	2	3	10	2	18,04	22,00	(8,67 3,17) (8,57 8,36) (0,63 5,96)	2,51E-03
max_div_20_2.txt	20	2	3	10	3	21,13	22,00	(8,67 3,17) (0,63 5,96) (8,57 8,36)	1,82E-03
max_div_20_2.txt	20	2	3	20	2	18,04	22,00	(8,67 3,17) (8,57 8,36) (0,63 5,96)	2,55E-03
max_div_20_2.txt	20	2	3	20	3	20,41	22,00	(8,67 3,17) (0,63 5,96) (8,57 8,36)	2,10E-03
max_div_20_2.txt	20	2	4	10	2	39,70	40,00	(1,16 4,47) (8,67 3,17) (8,57 8,36) (3,47 9,43)	4,43E-03
max_div_20_2.txt	20	2	4	10	3	38,76	40,00	(8,67 3,17) (3,47 9,43) (1,16 4,47) (8,57 8,36)	5,15E-03
max_div_20_2.txt	20	2	4	20	2	39,70	40,00	(1,16 4,47) (8,67 3,17) (8,57 8,36) (3,47 9,43)	5,14E-03
max_div_20_2.txt	20	2	4	20	3	37,03	40,00	(8,67 3,17) (1,16 4,47) (3,47 9,43) (8,57 8,36)	3,76E-03

max_div_20_2.txt	20	2	5	10	2	60,45	63,65	(1,97 3,50) (8,57 8,36) (0,63 5,96) (3,47 9,43) (8,67 3,17)	8,18E-03
max_div_20_2.txt	20	2	5	10	3	62,20	63,65	(1,97 3,50) (8,57 8,36) (8,67 3,17) (3,47 9,43) (0,63 5,96)	8,87E-03
max_div_20_2.txt	20	2	5	20	2	59,21	63,65	(0,63 5,96) (8,67 3,17) (3,47 9,43) (1,97 3,50) (8,57 8,36)	9,64E-03
max_div_20_2.txt	20	2	5	20	3	62,20	63,65	(0,63 5,96) (8,67 3,17) (3,47 9,43) (8,57 8,36) (1,97 3,50)	7,22E-03
max_div_15_2.txt	15	2	2	10	2	10,83	11,86	(8,65 9,98) (0,58 1,29)	7,24E-04
max_div_15_2.txt	15	2	2	10	3	11,86	11,86	(8,65 9,98) (0,58 1,29)	3,98E-04
max_div_15_2.txt	15	2	2	20	2	11,86	11,86	(8,65 9,98) (0,58 1,29)	2,93E-04
max_div_15_2.txt	15	2	2	20	3	11,63	11,86	(8,65 9,98) (0,58 1,29)	5,58E-04
max_div_15_2.txt	15	2	3	10	2	23,89	27,37	(8,65 9,98) (9,11 3,23) (0,58 1,29)	1,29E-03
max_div_15_2.txt	15	2	3	10	3	23,87	27,37	(8,65 9,98) (0,58 1,29) (9,11 3,23)	1,39E-03
max_div_15_2.txt	15	2	3	20	2	25,71	27,37	(8,65 9,98) (0,58 1,29) (9,11 3,23)	1,70E-03
max_div_15_2.txt	15	2	3	20	3	25,93	27,37	(8,65 9,98) (0,58 1,29) (9,11 3,23)	1,57E-03
max_div_15_2.txt	15	2	4	10	2	46,65	49,83	(9,11 3,23) (0,58 1,29) (8,65 9,98) (0,16 4,62)	3,70E-03

max_div_15_2.txt	15	2	4	10	3	47,50	49,83	(9,11 3,23) (0,58 1,29) (8,65 9,98) (0,16 4,62)	3,54E-03
max_div_15_2.txt	15	2	4	20	2	46,84	49,83	(9,11 3,23) (0,58 1,29) (8,65 9,98) (0,16 4,62)	3,57E-03
max_div_15_2.txt	15	2	4	20	3	48,25	49,83	(9,11 3,23) (0,58 1,29) (8,65 9,98) (0,16 4,62)	3,67E-03
max_div_15_2.txt	15	2	5	10	2	73,56	79,13	(8,65 9,98) (9,11 3,23) (9,96 8,17) (0,58 1,29) (0,16 4,62)	5,18E-03
max_div_15_2.txt	15	2	5	10	3	78,01	79,13	(8,65 9,98) (0,16 4,62) (9,11 3,23) (0,58 1,29) (9,96 8,17)	4,21E-03
max_div_15_2.txt	15	2	5	20	2	73,56	79,13	(8,65 9,98) (0,16 4,62) (9,11 3,23) (0,58 1,29) (9,96 8,17)	8,34E-03
max_div_15_2.txt	15	2	5	20	3	75,13	79,13	(8,65 9,98) (0,58 1,29) (9,11 3,23) (0,16 4,62) (9,96 8,17)	4,43E-03
max_div_30_3.txt	30	3	2	10	2	12,61	13,07	(8,06 9,59) (0,27) (6,71) (0,35 9,42)	1,39E-03
max_div_30_3.txt	30	3	2	10	3	11,45	13,07	(6,71 0,35) (9,42) (8,06) (9,59 0,27)	1,11E-03
max_div_30_3.txt	30	3	2	20	2	12,61	13,07	(8,06 9,59) (0,27) (6,71) (0,35 9,42)	1,28E-03
max_div_30_3.txt	30	3	2	20	3	12,61	13,07	(8,06 9,59) (0,27) (6,71) (0,35 9,42)	1,25E-03

max_div_30_3.txt	30	3	3	10	2	30,85	34,29	(1,53 9,56) 0,47 (8,06 0,27)	8,09 (8,32 8,98) 9,59	4,06E-03
max_div_30_3.txt	30	3	3	10	3	29,15	34,29	(8,32 8,98) 8,09 (8,06 0,27)	0,47 (1,53 9,56) 9,59	5,13E-03
max_div_30_3.txt	30	3	3	20	2	31,51	34,29	(8,06 0,27) 0,47 (1,53 9,56)	9,59 (8,32 8,98) 8,09	4,01E-03
max_div_30_3.txt	30	3	3	20	3	31,03	34,29	(8,32 8,98) 8,09 (8,06 0,27)	0,47 (1,53 9,56) 9,59	4,24E-03
max_div_30_3.txt	30	3	4	10	2	63,70	63,70	(8,06 0,27) 0,47 (1,53 9,56) 0,07	9,59 (8,32 8,98) 8,09 (6,75 1,80)	3,26E-03
max_div_30_3.txt	30	3	4	10	3	62,93	63,70	(8,06 0,27) 0,47 (1,53 9,56) 0,07	9,59 (8,32 8,98) 8,09 (6,75 1,80)	7,44E-03
max_div_30_3.txt	30	3	4	20	2	63,52	63,70	(8,06 0,27) 0,47 (1,53 9,56) 0,07	9,59 (8,32 8,98) 8,09 (6,75 1,80)	5,69E-03
max_div_30_3.txt	30	3	4	20	3	62,75	63,70	(8,06 0,27) 0,47 (1,53 9,56) 0,07	9,59 (8,32 8,98) 8,09 (6,75 1,80)	5,65E-03

max_div_30_3.txt	30	3	5	10	2	98,74	99,59	(8,06 0,27) 0,47 (1,53 9,56) 0,07 (7,54 9,66)	9,59 (8,32 8,98) 8,09 (6,75 1,80) 8,93	9,28E-03
max_div_30_3.txt	30	3	5	10	3	95,13	99,59	(8,06 0,27) 0,47 (1,53 9,56) 0,07 (7,54 9,66)	9,59 (8,32 8,98) 8,09 (6,75 1,80) 8,93	1,81E-02
max_div_30_3.txt	30	3	5	20	2	98,24	99,59	(8,06 0,27) 0,47 (1,53 9,56) 0,07 (7,54 9,66)	9,59 (8,32 8,98) 8,09 (6,75 1,80) 8,93	1,32E-02
max_div_30_3.txt	30	3	5	20	3	95,10	99,59	(8,06 0,27) 0,47 (1,53 9,56) 0,07 (7,54 9,66)	9,59 (8,32 8,98) 8,09 (6,75 1,80) 8,93	1,54E-02

D. Tabla de tiempo mínimo de procesamiento para cada instancia. Algoritmo de Ramificación y poda. Partiendo del algoritmo Voraz.

Problema	Estrategia	n	K	m	z	S	CPU	Nodos Generados
max_div_15_3.txt	SmallestZ	15	3	2	13,27	(9,88 9,88 6,26) (0,30 0,92 4,23)	1,60E-03	30
max_div_15_3.txt	Deeper	15	3	2	13,27	(9,88 9,88 6,26) (0,30 0,92 4,23)	1,43E-03	30
max_div_15_3.txt	SmallestZ	15	3	3	31,87	(9,88 9,88 6,26) (5,17 1,39 0,91) (1,71 1,95 9,22)	4,62E-03	43
max_div_15_3.txt	Deeper	15	3	3	31,87	(9,88 9,88 6,26) (5,17 1,39 0,91) (1,71 1,95 9,22)	3,96E-03	43
max_div_15_3.txt	SmallestZ	15	3	4	59,76	(9,88 9,88 6,26) (0,30 0,92 4,23) (1,71 1,95 9,22) (6,65 9,45 1,02)	1,07E-02	55
max_div_15_3.txt	Deeper	15	3	4	59,76	(9,88 9,88 6,26) (0,30 0,92 4,23) (1,71 1,95 9,22) (6,65 9,45 1,02)	1,45E-02	94
max_div_15_3.txt	SmallestZ	15	3	5	96,09	(9,88 9,88 6,26) (0,30 0,92 4,23) (1,71 1,95 9,22) (1,15 9,21 3,11) (9,47 3,56 1,83)	1,85E-02	66
max_div_15_3.txt	Deeper	15	3	5	96,09	(9,88 9,88 6,26) (0,30 0,92 4,23) (1,71 1,95 9,22) (1,15 9,21 3,11) (9,47 3,56 1,83)	2,44E-02	102
max_div_15_3.txt	SmallestZ	15	3	6	138,16	(9,88 9,88 6,26) (0,30 0,92 4,23) (1,71 1,95 9,22) (9,88 9,88 6,26) (9,47 3,56 1,83) (1,15 9,21 3,11)	3,33E-02	76
max_div_15_3.txt	Deeper	15	3	6	138,16	(9,88 9,88 6,26) (0,30 0,92 4,23) (1,71 1,95 9,22) (9,88 9,88 6,26) (9,47 3,56 1,83) (1,15 9,21 3,11)	3,16E-02	76

max_div_30_2.txt	SmallestZ	30	2	2	11,66	(9,84 8,96) (2,07 0,27)	5,50E-03	60
max_div_30_2.txt	Deeper	30	2	2	11,66	(9,84 8,96) (2,07 0,27)	4,35E-03	60
max_div_30_2.txt	SmallestZ	30	2	3	28,94	(9,84 8,96) (2,07 0,27) (1,91 9,60)	1,76E-02	88
max_div_30_2.txt	Deeper	30	2	3	28,94	(9,84 8,96) (2,07 0,27) (1,91 9,60)	2,23E-02	145
max_div_30_2.txt	SmallestZ	30	2	4	52,77	(9,84 8,96) (2,07 0,27) (1,91 9,60) (8,00 1,53)	3,89E-02	115
max_div_30_2.txt	Deeper	30	2	4	52,77	(9,84 8,96) (2,07 0,27) (1,91 9,60) (8,00 1,53)	4,06E-02	115
max_div_30_2.txt	SmallestZ	30	2	5	80,91	(9,84 8,96) (2,07 0,27) (1,91 9,60) (8,00 1,53) (0,65 3,26)	3,31E-01	604
max_div_30_2.txt	Deeper	30	2	5	80,91	(9,84 8,96) (2,07 0,27) (1,91 9,60) (8,00 1,53) (0,65 3,26)	4,95E-01	1324
max_div_30_2.txt	SmallestZ	30	2	6	118,99	(9,84 8,96) (2,07 0,27) (1,91 9,60) (8,00 1,53) (0,65 3,26) (9,84 8,96)	3,57E-01	436
max_div_30_2.txt	Deeper	30	2	6	118,99	(9,84 8,96) (2,07 0,27) (1,91 9,60) (8,00 1,53) (0,65 3,26) (9,84 8,96)	3,56E-01	436
max_div_20_3.txt	SmallestZ	20	3	2	11,80	(9,81 2,05 1,83) (0,65 3,76 9,07)	2,20E-03	40
max_div_20_3.txt	Deeper	20	3	2	11,80	(9,81 2,05 1,83) (0,65 3,76 9,07)	2,01E-03	40
max_div_20_3.txt	SmallestZ	20	3	3	30,87	(9,81 2,05 1,83) (0,65 3,76 9,07) (8,87 9,56 5,34)	8,10E-03	58
max_div_20_3.txt	Deeper	20	3	3	30,87	(9,81 2,05 1,83) (0,65 3,76 9,07) (8,87 9,56 5,34)	6,94E-03	58
max_div_20_3.txt	SmallestZ	20	3	4	56,69	(9,81 2,05 1,83) (0,65 3,76 9,07) (9,60 2,02 9,00) (0,83 7,06 3,34)	1,78E-02	75

max_div_20_3.txt	Deeper	20	3	4	56,69	(9,81 2,05 1,83) (0,65 3,76 9,07) (9,60 2,02 9,00) (0,83 7,06 3,34)	1,73E-02	75
max_div_20_3.txt	SmallestZ	20	3	5	92,83	(9,81 2,05 1,83) (0,65 3,76 9,07) (8,87 9,56 5,34) (0,83 7,06 3,34) (9,60 2,02 9,00)	1,11E-01	301
max_div_20_3.txt	Deeper	20	3	5	92,83	(9,81 2,05 1,83) (0,65 3,76 9,07) (8,87 9,56 5,34) (0,83 7,06 3,34) (9,60 2,02 9,00)	2,05E-01	786
max_div_20_3.txt	SmallestZ	20	3	6	131,49	(9,81 2,05 1,83) (0,65 3,76 9,07) (8,87 9,56 5,34) (0,83 7,06 3,34) (9,60 2,02 9,00) (8,02 0,12 3,56)	1,86E-01	323
max_div_20_3.txt	Deeper	20	3	6	131,49	(9,81 2,05 1,83) (0,65 3,76 9,07) (8,87 9,56 5,34) (0,83 7,06 3,34) (9,60 2,02 9,00) (8,02 0,12 3,56)	4,43E-01	951
max_div_20_2.txt	SmallestZ	20	2	2	8,51	(0,63 5,96) (8,67 3,17)	2,90E-03	40
max_div_20_2.txt	Deeper	20	2	2	8,51	(0,63 5,96) (8,67 3,17)	2,38E-03	40
max_div_20_2.txt	SmallestZ	20	2	3	22,00	(0,63 5,96) (8,67 3,17) (8,57 8,36)	7,76E-03	58
max_div_20_2.txt	Deeper	20	2	3	22,00	(0,63 5,96) (8,67 3,17) (8,57 8,36)	1,01E-02	95
max_div_20_2.txt	SmallestZ	20	2	4	40,00	(3,47 9,43) (8,67 3,17) (8,57 8,36) (1,16 4,47)	1,75E-02	75
max_div_20_2.txt	Deeper	20	2	4	40,00	(3,47 9,43) (8,67 3,17) (8,57 8,36) (1,16 4,47)	1,65E-02	75
max_div_20_2.txt	SmallestZ	20	2	5	63,65	(0,63 5,96) (8,67 3,17) (8,57 8,36) (1,97 3,50) (3,47 9,43)	3,45E-02	91
max_div_20_2.txt	Deeper	20	2	5	63,65	(0,63 5,96) (8,67 3,17) (8,57 8,36)	3,29E-02	91

						(1,97 3,50) (3,47 9,43)		
max_div_20_2.txt	SmallestZ	20	2	6	92,33	(0,63 5,96) (8,67 3,17) (8,57 8,36) (3,47 9,43) (8,67 3,17) (1,16 4,47)	6,37E-02	106
max_div_20_2.txt	Deeper	20	2	6	92,33	(0,63 5,96) (8,67 3,17) (8,57 8,36) (3,47 9,43) (8,67 3,17) (1,16 4,47)	6,10E-02	106
max_div_15_2.txt	SmallestZ	15	2	2	11,86	(8,65 9,98) (0,58 1,29)	1,32E-03	30
max_div_15_2.txt	Deeper	15	2	2	11,86	(8,65 9,98) (0,58 1,29)	1,19E-03	30
max_div_15_2.txt	SmallestZ	15	2	3	27,37	(8,65 9,98) (0,58 1,29) (9,11 3,23)	4,60E-03	43
max_div_15_2.txt	Deeper	15	2	3	27,37	(8,65 9,98) (0,58 1,29) (9,11 3,23)	3,89E-03	43
max_div_15_2.txt	SmallestZ	15	2	4	49,83	(8,65 9,98) (0,58 1,29) (9,11 3,23) (0,16 4,62)	9,91E-03	55
max_div_15_2.txt	Deeper	15	2	4	49,83	(8,65 9,98) (0,58 1,29) (9,11 3,23) (0,16 4,62)	1,43E-02	94
max_div_15_2.txt	SmallestZ	15	2	5	79,13	(8,65 9,98) (0,16 4,62) (9,11 3,23) (0,58 1,29) (9,96 8,17)	1,75E-02	66
max_div_15_2.txt	Deeper	15	2	5	79,13	(8,65 9,98) (0,16 4,62) (9,11 3,23) (0,58 1,29) (9,96 8,17)	1,70E-02	66
max_div_15_2.txt	SmallestZ	15	2	6	115,43	(8,65 9,98) (0,58 1,29) (9,11 3,23) (0,58 1,29) (9,96 8,17) (1,35 7,11)	3,45E-02	76
max_div_15_2.txt	Deeper	15	2	6	115,43	(8,65 9,98) (0,58 1,29) (9,11 3,23) (0,58 1,29) (9,96 8,17) (1,35 7,11)	3,39E-02	76
max_div_30_3.txt	SmallestZ	30	3	2	13,07	(8,06 9,59 0,27) (6,71 0,35 9,42)	5,89E-03	60
max_div_30_3.txt	Deeper	30	3	2	13,07	(8,06 9,59 0,27) (6,71 0,35 9,42)	4,80E-03	60

max_div_30_3.txt	SmallestZ	30	3	3	34,29	(8,06 9,59 0,27) (8,32 0,47 8,98) (1,53 8,09 9,56)	1,67E-02	88
max_div_30_3.txt	Deeper	30	3	3	34,29	(8,06 9,59 0,27) (8,32 0,47 8,98) (1,53 8,09 9,56)	1,85E-02	88
max_div_30_3.txt	SmallestZ	30	3	4	63,70	(8,06 9,59 0,27) (8,32 0,47 8,98) (1,53 8,09 9,56) (6,75 0,07 1,80)	4,47E-02	115
max_div_30_3.txt	Deeper	30	3	4	63,70	(8,06 9,59 0,27) (8,32 0,47 8,98) (1,53 8,09 9,56) (6,75 0,07 1,80)	4,00E-02	115
max_div_30_3.txt	SmallestZ	30	3	5	99,59	(8,06 9,59 0,27) (8,32 0,47 8,98) (1,53 8,09 9,56) (6,75 0,07 1,80) (7,54 8,93 9,66)	8,69E-02	141
max_div_30_3.txt	Deeper	30	3	5	99,59	(8,06 9,59 0,27) (8,32 0,47 8,98) (1,53 8,09 9,56) (6,75 0,07 1,80) (7,54 8,93 9,66)	7,79E-02	141
max_div_30_3.txt	SmallestZ	30	3	6	143,83	(8,06 9,59 0,27) (6,71 0,35 9,42) (1,53 8,09 9,56) (6,75 0,07 1,80) (9,99 4,40 9,76) (8,06 9,59 0,27)	1,50E-01	166
max_div_30_3.txt	Deeper	30	3	6	143,83	(8,06 9,59 0,27) (6,71 0,35 9,42) (1,53 8,09 9,56) (6,75 0,07 1,80) (9,99 4,40 9,76) (8,06 9,59 0,27)	1,54E-01	166

E. Tabla de tiempo mínimo de procesamiento para cada instancia. Algoritmo de Ramificación y poda. Partiendo de GRASP con 10 iteraciones.

Problema	Estrategia	n	K	m	z	S	CPU	Nodos Generados
max_div_15_3.txt	Smallest Z	15	3	2	13,27	(9,88 9,88 6,26) (0,30 0,92 4,23)	1,66E-03	30
max_div_15_3.txt	Deeper	15	3	2	13,27	(0,30 0,92 4,23) (9,88 9,88 6,26)	1,26E-03	30
max_div_15_3.txt	Smallest Z	15	3	3	31,87	(9,88 9,88 6,26) (1,71 1,95 9,22) (5,17 1,39 0,91)	4,71E-03	43
max_div_15_3.txt	Deeper	15	3	3	31,87	(9,88 9,88 6,26) (1,71 1,95 9,22) (5,17 1,39 0,91)	3,70E-03	43
max_div_15_3.txt	Smallest Z	15	3	4	59,76	(1,71 1,95 9,22) (9,88 9,88 6,26) (0,30 0,92 4,23) (6,65 9,45 1,02)	9,46E-03	55
max_div_15_3.txt	Deeper	15	3	4	59,76	(1,71 1,95 9,22) (6,65 9,45 1,02) (9,88 9,88 6,26) (0,30 0,92 4,23)	9,37E-03	55
max_div_15_3.txt	Smallest Z	15	3	5	96,09	(9,88 9,88 6,26) (0,30 0,92 4,23) (1,15 9,21 3,11) (1,71 1,95 9,22) (9,47 3,56 1,83)	2,01E-02	66
max_div_15_3.txt	Deeper	15	3	5	96,09	(9,88 9,88 6,26) (9,47 3,56 1,83) (0,30 0,92 4,23) (1,15 9,21 3,11) (1,71 1,95 9,22)	1,74E-02	66
max_div_15_3.txt	Smallest Z	15	3	6	138,16	(9,88 9,88 6,26) (1,15 9,21 3,11) (1,71 1,95 9,22) (9,88 9,88 6,26) (9,47 3,56 1,83) (0,30 0,92 4,23)	3,66E-02	76
max_div_15_3.txt	Deeper	15	3	6	138,16	(9,88 9,88 6,26) (0,30 0,92 4,23) (1,71 1,95 9,22) (9,88 9,88 6,26)	3,39E-02	76

						(9,47 3,56 1,83) (1,15 9,21 3,11)		
max_div_30_2 .txt	Smallest Z	30	2	2	11,66	(9,84 8,96) (2,07 0,27)	4,85E-03	60
max_div_30_2 .txt	Deeper	30	2	2	11,66	(9,84 8,96) (2,07 0,27)	4,12E-03	60
max_div_30_2 .txt	Smallest Z	30	2	3	28,94	(2,07 0,27) (9,84 8,96) (1,91 9,60)	1,52E-02	88
max_div_30_2 .txt	Deeper	30	2	3	28,94	(2,07 0,27) (1,91 9,60) (9,84 8,96)	1,35E-02	88
max_div_30_2 .txt	Smallest Z	30	2	4	52,77	(2,07 0,27) (1,91 9,60) (8,00 1,53) (9,84 8,96)	4,32E-02	115
max_div_30_2 .txt	Deeper	30	2	4	52,77	(9,84 8,96) (2,07 0,27) (1,91 9,60) (8,00 1,53)	3,48E-02	115
max_div_30_2 .txt	Smallest Z	30	2	5	81,20	(1,91 9,60) (2,07 0,27) (9,84 8,96) (2,07 0,27) (9,84 8,96)	7,36E-02	141
max_div_30_2 .txt	Deeper	30	2	5	81,20	(1,91 9,60) (9,84 8,96) (2,07 0,27) (9,84 8,96) (2,07 0,27)	5,89E-02	141
max_div_30_2 .txt	Smallest Z	30	2	6	118,99	(9,84 8,96) (0,65 3,26) (1,91 9,60) (2,07 0,27) (8,00 1,53) (9,84 8,96)	1,40E-01	166
max_div_30_2 .txt	Deeper	30	2	6	118,99	(9,84 8,96) (0,65 3,26) (9,84 8,96) (1,91 9,60) (2,07 0,27) (8,00 1,53)	1,44E-01	166
max_div_20_3 .txt	Smallest Z	20	3	2	11,80	(0,65 3,76 9,07) (9,81 2,05 1,83)	2,84E-03	40
max_div_20_3 .txt	Deeper	20	3	2	11,80	(0,65 3,76 9,07) (9,81 2,05 1,83)	2,32E-03	40

max_div_20_3 .txt	Smallest Z	20	3	3	30,87	(0,65 3,76 9,07) (8,87 9,56 5,34) (9,81 2,05 1,83)	7,74E-03	58
max_div_20_3 .txt	Deeper	20	3	3	30,87	(0,65 3,76 9,07) (9,81 2,05 1,83) (8,87 9,56 5,34)	6,24E-03	58
max_div_20_3 .txt	Smallest Z	20	3	4	56,69	(9,60 2,02 9,00) (0,65 3,76 9,07) (9,81 2,05 1,83) (0,83 7,06 3,34)	1,59E-02	75
max_div_20_3 .txt	Deeper	20	3	4	56,69	(0,65 3,76 9,07) (9,60 2,02 9,00) (9,81 2,05 1,83) (0,83 7,06 3,34)	1,50E-02	75
max_div_20_3 .txt	Smallest Z	20	3	5	92,83	(9,81 2,05 1,83) (0,83 7,06 3,34) (0,65 3,76 9,07) (8,87 9,56 5,34) (9,60 2,02 9,00)	3,05E-02	91
max_div_20_3 .txt	Deeper	20	3	5	92,83	(0,65 3,76 9,07) (8,87 9,56 5,34) (9,81 2,05 1,83) (0,83 7,06 3,34) (9,60 2,02 9,00)	3,63E-02	91
max_div_20_3 .txt	Smallest Z	20	3	6	131,49	(0,65 3,76 9,07) (8,02 0,12 3,56) (9,81 2,05 1,83) (0,83 7,06 3,34) (8,87 9,56 5,34) (9,60 2,02 9,00)	6,20E-02	106
max_div_20_3 .txt	Deeper	20	3	6	131,49	(0,65 3,76 9,07) (8,02 0,12 3,56) (9,60 2,02 9,00) (0,83 7,06 3,34) (9,81 2,05 1,83) (8,87 9,56 5,34)	6,50E-02	106
max_div_20_2 .txt	Smallest Z	20	2	2	8,51	(8,67 3,17) (0,63 5,96)	2,62E-03	40
max_div_20_2 .txt	Deeper	20	2	2	8,51	(8,67 3,17) (0,63 5,96)	2,11E-03	40
max_div_20_2 .txt	Smallest Z	20	2	3	22,00	(8,67 3,17) (8,57 8,36) (0,63 5,96)	7,34E-03	58
max_div_20_2 .txt	Deeper	20	2	3	22,00	(8,67 3,17) (8,57 8,36) (0,63 5,96)	6,68E-03	58

max_div_20_2 .txt	Smallest Z	20	2	4	40,00	(8,67 3,17) (1,16 4,47) (3,47 9,43) (8,57 8,36)	1,76E-02	75
max_div_20_2 .txt	Deeper	20	2	4	40,00	(8,67 3,17) (1,16 4,47) (8,57 8,36) (3,47 9,43)	1,56E-02	75
max_div_20_2 .txt	Smallest Z	20	2	5	63,65	(0,63 5,96) (8,57 8,36) (8,67 3,17) (1,97 3,50) (3,47 9,43)	4,94E-02	142
max_div_20_2 .txt	Deeper	20	2	5	63,65	(0,63 5,96) (8,67 3,17) (3,47 9,43) (1,97 3,50) (8,57 8,36)	3,18E-02	91
max_div_20_2 .txt	Smallest Z	20	2	6	92,33	(1,16 4,47) (3,47 9,43) (8,67 3,17) (0,63 5,96) (8,67 3,17) (8,57 8,36)	5,97E-02	106
max_div_20_2 .txt	Deeper	20	2	6	92,33	(1,16 4,47) (3,47 9,43) (8,67 3,17) (0,63 5,96) (8,57 8,36) (8,67 3,17)	5,83E-02	106
max_div_15_2 .txt	Smallest Z	15	2	2	11,86	(8,65 9,98) (0,58 1,29)	1,52E-03	30
max_div_15_2 .txt	Deeper	15	2	2	11,86	(8,65 9,98) (0,58 1,29)	1,33E-03	30
max_div_15_2 .txt	Smallest Z	15	2	3	27,37	(9,11 3,23) (0,58 1,29) (8,65 9,98)	4,68E-03	43
max_div_15_2 .txt	Deeper	15	2	3	27,37	(8,65 9,98) (0,58 1,29) (9,11 3,23)	3,59E-03	43
max_div_15_2 .txt	Smallest Z	15	2	4	49,83	(9,11 3,23) (0,58 1,29) (8,65 9,98) (0,16 4,62)	9,54E-03	55
max_div_15_2 .txt	Deeper	15	2	4	49,83	(9,11 3,23) (0,58 1,29)	8,27E-03	55

						(8,65 9,98) (0,16 4,62)		
max_div_15_2 .txt	Smallest Z	15	2	5	79,13	(8,65 9,98) (0,16 4,62) (0,58 1,29) (9,96 8,17) (9,11 3,23)	5,86E-02	260
max_div_15_2 .txt	Deeper	15	2	5	79,13	(8,65 9,98) (0,58 1,29) (9,11 3,23) (9,96 8,17) (0,16 4,62)	1,73E-02	66
max_div_15_2 .txt	Smallest Z	15	2	6	115,43	(8,65 9,98) (0,58 1,29) (9,11 3,23) (1,35 7,11) (0,58 1,29) (9,96 8,17)	3,45E-02	76
max_div_15_2 .txt	Deeper	15	2	6	115,43	(8,65 9,98) (1,35 7,11) (0,58 1,29) (9,96 8,17) (0,58 1,29) (9,11 3,23)	3,25E-02	76
max_div_30_3 .txt	Smallest Z	30	3	2	13,07	(6,71 0,35 9,42) (8,06 9,59 0,27)	5,64E-03	60
max_div_30_3 .txt	Deeper	30	3	2	13,07	(8,06 9,59 0,27) (6,71 0,35 9,42)	4,83E-03	60
max_div_30_3 .txt	Smallest Z	30	3	3	34,29	(1,53 8,09 9,56) (8,32 0,47 8,98) (8,06 9,59 0,27)	1,69E-02	88
max_div_30_3 .txt	Deeper	30	3	3	34,29	(8,06 9,59 0,27) (8,32 0,47 8,98) (1,53 8,09 9,56)	1,52E-02	88
max_div_30_3 .txt	Smallest Z	30	3	4	63,70	(8,06 9,59 0,27) (8,32 0,47 8,98) (1,53 8,09 9,56) (6,75 0,07 1,80)	4,11E-02	115
max_div_30_3 .txt	Deeper	30	3	4	63,70	(8,06 9,59 0,27) (8,32 0,47 8,98) (1,53 8,09 9,56) (6,75 0,07 1,80)	3,86E-02	115
max_div_30_3 .txt	Smallest Z	30	3	5	99,59	(8,06 9,59 0,27) (6,75 0,07 1,80) (1,53 8,09 9,56) (8,32 0,47 8,98) (7,54 8,93 9,66)	8,70E-02	141

max_div_30_3 .txt	Deeper	30	3	5	99,59	(8,06 9,59 0,27) (8,32 0,47 8,98) (1,53 8,09 9,56) (6,75 0,07 1,80) (7,54 8,93 9,66)	8,42E-02	141
max_div_30_3 .txt	Smallest Z	30	3	6	143,83	(6,75 0,07 1,80) (9,99 4,40 9,76) (8,06 9,59 0,27) (6,71 0,35 9,42) (1,53 8,09 9,56) (8,06 9,59 0,27)	1,41E-01	166
max_div_30_3 .txt	Deeper	30	3	6	143,83	(1,53 8,09 9,56) (6,75 0,07 1,80) (9,99 4,40 9,76) (8,06 9,59 0,27) (6,71 0,35 9,42) (8,06 9,59 0,27)	1,49E-01	166

F. Tabla de tiempo mínimo de procesamiento para cada instancia. Algoritmo de Ramificación y poda. Partiendo de GRASP con 500 iteraciones.

Problema	Estrategia	n	K	m	z	S	CPU	Nodos Generados
max_div_15_3.txt	SmallestZ	15	3	2	13,27	(9,88 9,88 6,26) (0,30 0,92 4,23)	1,63E-03	30
max_div_15_3.txt	Deeper	15	3	2	13,27	(9,88 9,88 6,26) (0,30 0,92 4,23)	1,39E-03	30
max_div_15_3.txt	SmallestZ	15	3	3	31,87	(9,88 9,88 6,26) (5,17 1,39 0,91) (1,71 1,95 9,22)	4,51E-03	43
max_div_15_3.txt	Deeper	15	3	3	31,87	(9,88 9,88 6,26) (1,71 1,95 9,22) (5,17 1,39 0,91)	3,75E-03	43
max_div_15_3.txt	SmallestZ	15	3	4	59,76	(1,71 1,95 9,22) (6,65 9,45 1,02) (9,88 9,88 6,26) (0,30 0,92 4,23)	1,00E-02	55
max_div_15_3.txt	Deeper	15	3	4	59,76	(1,71 1,95 9,22) (6,65 9,45 1,02) (9,88 9,88 6,26) (0,30 0,92 4,23)	9,55E-03	55
max_div_15_3.txt	SmallestZ	15	3	5	96,09	(9,88 9,88 6,26) (9,47 3,56 1,83) (0,30 0,92 4,23) (1,15 9,21 3,11) (1,71 1,95 9,22)	1,63E-02	66
max_div_15_3.txt	Deeper	15	3	5	96,09	(9,88 9,88 6,26) (9,47 3,56 1,83) (0,30 0,92 4,23) (1,71 1,95 9,22) (1,15 9,21 3,11)	1,66E-02	66
max_div_15_3.txt	SmallestZ	15	3	6	138,16	(9,88 9,88 6,26) (1,71 1,95 9,22) (0,30 0,92 4,23) (9,88 9,88 6,26) (9,47 3,56 1,83) (1,15 9,21 3,11)	3,42E-02	76
max_div_15_3.txt	Deeper	15	3	6	138,16	(9,88 9,88 6,26) (1,15 9,21 3,11) (9,88 9,88 6,26) (1,71 1,95 9,22)	3,12E-02	76

						(9,47 3,56 1,83) (0,30 0,92 4,23)		
max_div_30_2.t xt	SmallestZ	30	2	2	11,66	(9,84 8,96) (2,07 0,27)	5,36E-03	60
max_div_30_2.t xt	Deeper	30	2	2	11,66	(9,84 8,96) (2,07 0,27)	4,19E-03	60
max_div_30_2.t xt	SmallestZ	30	2	3	28,94	(2,07 0,27) (1,91 9,60) (9,84 8,96)	1,66E-02	88
max_div_30_2.t xt	Deeper	30	2	3	28,94	(2,07 0,27) (9,84 8,96) (1,91 9,60)	1,51E-02	88
max_div_30_2.t xt	SmallestZ	30	2	4	52,77	(1,91 9,60) (8,00 1,53) (2,07 0,27) (9,84 8,96)	3,10E-02	115
max_div_30_2.t xt	Deeper	30	2	4	52,77	(9,84 8,96) (2,07 0,27) (1,91 9,60) (8,00 1,53)	3,31E-02	115
max_div_30_2.t xt	SmallestZ	30	2	5	81,20	(1,91 9,60) (2,07 0,27) (9,84 8,96) (9,84 8,96) (2,07 0,27)	7,32E-02	141
max_div_30_2.t xt	Deeper	30	2	5	81,20	(1,91 9,60) (2,07 0,27) (9,84 8,96) (2,07 0,27) (9,84 8,96)	7,89E-02	141
max_div_30_2.t xt	SmallestZ	30	2	6	118,99	(1,91 9,60) (8,00 1,53) (9,84 8,96) (0,65 3,26) (2,07 0,27) (9,84 8,96)	1,31E-01	166
max_div_30_2.t xt	Deeper	30	2	6	118,99	(9,84 8,96) (0,65 3,26) (2,07 0,27) (9,84 8,96) (1,91 9,60) (8,00 1,53)	1,43E-01	166
max_div_20_3.t xt	SmallestZ	20	3	2	11,80	(9,81 2,05 1,83) (0,65 3,76 9,07)	2,68E-03	40
max_div_20_3.t xt	Deeper	20	3	2	11,80	(9,81 2,05 1,83) (0,65 3,76 9,07)	2,29E-03	40
max_div_20_3.t xt	SmallestZ	20	3	3	30,87	(9,81 2,05 1,83) (0,65 3,76 9,07) (8,87 9,56 5,34)	6,59E-03	58
max_div_20_3.t xt	Deeper	20	3	3	30,87	(9,81 2,05 1,83) (8,87 9,56 5,34) (0,65 3,76 9,07)	7,39E-03	58
max_div_20_3.t xt	SmallestZ	20	3	4	56,69	(0,65 3,76 9,07) (9,81 2,05 1,83) (9,60 2,02 9,00) (0,83 7,06 3,34)	1,52E-02	75

max_div_20_3.txt	Deeper	20	3	4	56,69	(9,60 2,02 9,00) (0,65 3,76 9,07) (0,83 7,06 3,34) (9,81 2,05 1,83)	1,51E-02	75
max_div_20_3.txt	SmallestZ	20	3	5	92,83	(0,65 3,76 9,07) (8,87 9,56 5,34) (9,60 2,02 9,00) (0,83 7,06 3,34) (9,81 2,05 1,83)	3,64E-02	91
max_div_20_3.txt	Deeper	20	3	5	92,83	(9,60 2,02 9,00) (8,87 9,56 5,34) (0,83 7,06 3,34) (9,81 2,05 1,83) (0,65 3,76 9,07)	3,22E-02	91
max_div_20_3.txt	SmallestZ	20	3	6	131,49	(9,81 2,05 1,83) (0,65 3,76 9,07) (9,60 2,02 9,00) (8,87 9,56 5,34) (0,83 7,06 3,34) (8,02 0,12 3,56)	6,24E-02	106
max_div_20_3.txt	Deeper	20	3	6	131,49	(9,81 2,05 1,83) (0,83 7,06 3,34) (0,65 3,76 9,07) (9,60 2,02 9,00) (8,02 0,12 3,56) (8,87 9,56 5,34)	6,11E-02	106
max_div_20_2.txt	SmallestZ	20	2	2	8,51	(8,67 3,17) (0,63 5,96)	2,78E-03	40
max_div_20_2.txt	Deeper	20	2	2	8,51	(8,67 3,17) (0,63 5,96)	2,28E-03	40
max_div_20_2.txt	SmallestZ	20	2	3	22,00	(8,67 3,17) (0,63 5,96) (8,57 8,36)	9,51E-03	58
max_div_20_2.txt	Deeper	20	2	3	22,00	(8,67 3,17) (8,57 8,36) (0,63 5,96)	6,87E-03	58
max_div_20_2.txt	SmallestZ	20	2	4	40,00	(1,16 4,47) (8,67 3,17) (3,47 9,43) (8,57 8,36)	1,71E-02	75
max_div_20_2.txt	Deeper	20	2	4	40,00	(1,16 4,47) (8,67 3,17) (8,57 8,36) (3,47 9,43)	1,67E-02	75
max_div_20_2.txt	SmallestZ	20	2	5	63,65	(1,97 3,50) (3,47 9,43) (0,63 5,96) (8,57 8,36) (8,67 3,17)	3,18E-02	91
max_div_20_2.txt	Deeper	20	2	5	63,65	(1,97 3,50) (8,57 8,36) (0,63 5,96)	3,05E-02	91

						(8,67 3,17) (3,47 9,43)		
max_div_20_2.txt	SmallestZ	20	2	6	92,33	(8,67 3,17) (3,47 9,43) (1,16 4,47) (8,57 8,36) (0,63 5,96) (8,67 3,17)	5,78E-02	106
max_div_20_2.txt	Deeper	20	2	6	92,33	(8,67 3,17) (3,47 9,43) (1,16 4,47) (8,67 3,17) (0,63 5,96) (8,57 8,36)	5,75E-02	106
max_div_15_2.txt	SmallestZ	15	2	2	11,86	(8,65 9,98) (0,58 1,29)	1,61E-03	30
max_div_15_2.txt	Deeper	15	2	2	11,86	(8,65 9,98) (0,58 1,29)	1,21E-03	30
max_div_15_2.txt	SmallestZ	15	2	3	27,37	(9,11 3,23) (0,58 1,29) (8,65 9,98)	4,17E-03	43
max_div_15_2.txt	Deeper	15	2	3	27,37	(8,65 9,98) (9,11 3,23) (0,58 1,29)	3,69E-03	43
max_div_15_2.txt	SmallestZ	15	2	4	49,83	(9,11 3,23) (0,58 1,29) (8,65 9,98) (0,16 4,62)	9,06E-03	55
max_div_15_2.txt	Deeper	15	2	4	49,83	(9,11 3,23) (0,58 1,29) (8,65 9,98) (0,16 4,62)	8,64E-03	55
max_div_15_2.txt	SmallestZ	15	2	5	79,13	(8,65 9,98) (0,58 1,29) (9,11 3,23) (0,16 4,62) (9,96 8,17)	2,11E-02	66
max_div_15_2.txt	Deeper	15	2	5	79,13	(8,65 9,98) (0,16 4,62) (9,11 3,23) (0,58 1,29) (9,96 8,17)	1,62E-02	66
max_div_15_2.txt	SmallestZ	15	2	6	115,43	(8,65 9,98) (1,35 7,11) (0,58 1,29) (9,96 8,17) (0,58 1,29) (9,11 3,23)	3,38E-02	76
max_div_15_2.txt	Deeper	15	2	6	115,43	(8,65 9,98) (1,35 7,11) (0,58 1,29) (9,96 8,17) (0,58 1,29) (9,11 3,23)	3,40E-02	76
max_div_30_3.txt	SmallestZ	30	3	2	13,07	(8,06 9,59 0,27) (6,71 0,35 9,42)	6,12E-03	60
max_div_30_3.txt	Deeper	30	3	2	13,07	(6,71 0,35 9,42) (8,06 9,59 0,27)	4,42E-03	60

max_div_30_3.txt	SmallestZ	30	3	3	34,29	(1,53 8,09 9,56) (8,32 0,47 8,98) (8,06 9,59 0,27)	1,85E-02	88
max_div_30_3.txt	Deeper	30	3	3	34,29	(1,53 8,09 9,56) (8,06 9,59 0,27) (8,32 0,47 8,98)	1,34E-02	88
max_div_30_3.txt	SmallestZ	30	3	4	63,70	(8,06 9,59 0,27) (8,32 0,47 8,98) (1,53 8,09 9,56) (6,75 0,07 1,80)	4,15E-02	115
max_div_30_3.txt	Deeper	30	3	4	63,70	(8,06 9,59 0,27) (8,32 0,47 8,98) (1,53 8,09 9,56) (6,75 0,07 1,80)	3,53E-02	115
max_div_30_3.txt	SmallestZ	30	3	5	99,59	(8,06 9,59 0,27) (8,32 0,47 8,98) (1,53 8,09 9,56) (6,75 0,07 1,80) (7,54 8,93 9,66)	8,49E-02	141
max_div_30_3.txt	Deeper	30	3	5	99,59	(8,06 9,59 0,27) (8,32 0,47 8,98) (1,53 8,09 9,56) (6,75 0,07 1,80) (7,54 8,93 9,66)	8,36E-02	141
max_div_30_3.txt	SmallestZ	30	3	6	143,83	(1,53 8,09 9,56) (6,75 0,07 1,80) (9,99 4,40 9,76) (8,06 9,59 0,27) (6,71 0,35 9,42) (8,06 9,59 0,27)	1,54E-01	166
max_div_30_3.txt	Deeper	30	3	6	143,83	(1,53 8,09 9,56) (6,75 0,07 1,80) (9,99 4,40 9,76) (8,06 9,59 0,27) (6,71 0,35 9,42) (8,06 9,59 0,27)	1,34E-01	166

G. Representaciones gráficas en función del número de puntos, del tiempo de CPU y valor de z. Comparación algoritmo voraz y constructivo local.

