

Heurísticas Greedy: Localización de Servicios

García Prado, Sergio
sergio@garciparedes.me

31 de marzo de 2017

Resumen

En este documento se describe el modelo matemático de los problema de cubrimiento de conjuntos (Set-Covering Problem), el problema de cubrimiento máximo (Max-Covering Problem) y el problema de la p -mediana (P-Median Problem). Además, se describen soluciones heurísticas basadas en métodos Greedy, Greedy Aleatorizado y Búsqueda Local para la resolución de los mismos, las cuales reducen drásticamente la complejidad temporal de la solución a costa de una determinada tasa de error. Además de la descripción de los problemas mediante el modelo matemático y la descripción de soluciones heurísticas se muestran los resultados a distintos problemas concretos.

1. INTRODUCCIÓN

[TODO]

6. SET-COVERING PROBLEM

El problema de *Set Covering* o *Cubrimiento de Conjuntos* consiste en la asignación de un conjunto de recursos n recursos x_j cuyo uso tiene un coste de c_j para cumplir m necesidades. Las necesidades que cubre cada recurso se representan a través de a_{ij} . La modelización matemática de este problema se muestra en la ecuación (1).

$$\begin{aligned} &\text{Minimizar} && \sum_{j=1}^n c_j x_j \\ &\text{sujeto a} && \sum_{j=1}^n a_{ij} x_j \geq 1, && i = 1, \dots, m \\ &&& x_j \in \{0, 1\}, && j = 1, \dots, n \end{aligned} \tag{1}$$

Ecuación 1: *Formulación de Set-Covering Problem.*

Debido al alto nivel de complejidad de la solución exacta para el problema de cubrimiento de conjuntos, se han propuesto soluciones heurísticas que mejoran el tiempo necesario para encontrar una solución. Sin embargo, dichas estrategias conllevan una penalización en cuanto optimalidad de resultados. Estas soluciones se convierten en algo necesario cuando se pretende resolver problemas de tamaño masivo, lo que conlleva un elevado tiempo de ejecución.

A pesar de no conseguir el resultado óptimo, las heurísticas Greedy presentan resultados aceptables en un gran número de ocasiones. En la figura 1 se muestra una *Heurística Greedy* para el problema de cubrimiento de conjuntos. La intuición en la cual se basa consiste en seleccionar en cada iteración del algoritmo el punto de servicio con menor coste despues marcar como cubiertas todas las demandas que cubre. Se realiza este proceso hasta que todas las demandas quedan cubiertas.

```
nfilcub := 0
while(nfilcub < m) do
  forall(j in servicios | fijada(j) = 0) do
    d(j) := 0
    forall(i in demandas) do
      if(cubierta(i) = 0 and a(i,j) = 1) then
        d(j) := d(j) + 1
      end-if
    end-do
    if(d(j) = 0) then
      fijada(j) := 1
      solu(j) := 0
    end-if
  end-do
  aux := -99
  forall(j in servicios | fijada(j) = 0) do
    if(d(j) > aux) then
      aux := d(j)
      jmax := j
    end-if
  end-do
  solu(jmax) := 1
  fijada(jmax) := 1
  forall(i in demandas) do
    if(cubierta(i) = 0 and a(i, jmax) = 1) then
      cubierta(i) := 1
      nfilcub := nfilcub +1
    end-if
  end-do
end-do
```

Figura 1: *Heurística Greedy para Set-Covering Problem*

Existe una optimización computacional que se puede realizar a la herurística greedy de la figura 1 basada en la eliminación de columnas no utilizadas, la cual se lleva a cabo al final de cada iteración. Dicho código se ilustra en la figura 2

Para resolver los problemas derivados del determinismo de la solución Greedy, existen técnicas basadas en el apoyo de la componente aleatoria en la selección del punto de servicio a utilizar. Esta técnica consiste en elegir aleatoriamente la solución entre las k mejores con un valor de k fijado a priori. Por último, se realiza dicho proceso durante un número N determinado de veces también fijado a priori, para después seleccionar la mejor solución de entre todas ellas. Nótese que esto no presenta mejoras en cuanto al tiempo de ejecución, sino que lo que trata de mejorar es la precisión del resultado. En la figura 3 se muestra el código para implementar dicha solución.

```
forall(i in demandas) do
  w(i):=0
  forall(j in servicios) do
    if(solu(j)=1 and a(i,j)=1)then
      w(i):=w(i)+1
    end-if
  end-do
end-do
forall(k in servicios | solu(k)=1) do
  elimin := 1
  forall(i in demandas) do
    if(w(i) < 1+a(i,k))then
      elimin :=0
    end-if
  end-do
  if(elimin=1) then
    solu(k) :=0
    forall(i in demandas)w(i):=w(i)-a(i,k)
  end-if
end-do
```

Figura 2: Eliminación de columnas no utilizadas

6.1 Ejercicio Nueva York

En este caso, se ha propuesto resolver el problema de *Cubrimiento de Conjuntos* sobre un conjunto de datos de entrada referidos a las distancias entre 30 distritos de la ciudad de Nueva York. Se considera que un distrito ha sido cubierto si la distancia a un punto de servicio es menor o igual que un determinado valor dc denominado distancia de cubrimiento. En este caso se ha resuelto para valores enteros comprendidos en el intervalo $[70, 120]$ mediante la estrategia *Greedy* y la de *Solución Óptima*. Los resultados obtenidos se muestran gráficamente en la figura 4 y de manera tabular en la tabla 1

6.2 Ejercicio *aint1*

En este caso, se ha propuesto resolver el problema de *Cubrimiento de Conjuntos* sobre un conjunto de datos de entrada de tamaño relativamente elevado, con $m = 356$ puntos de demanda y $n = 22$ puntos de servicio. Se considera que un distrito ha sido cubierto si la distancia a un punto de servicio es menor o igual que un determinado valor dc denominado distancia de cubrimiento. En este caso se ha resuelto para valores enteros comprendidos en el intervalo $[250, 400]$ mediante la estrategia *Greedy*, la estrategia *Greedy Aleatorizada* (con parámetros $k = 5$ y $n = 100$) y la de *Solución Óptima*. Los resultados obtenidos se muestran gráficamente en la figura 5 y de manera tabular en las tablas 2, 3 y 4.

6.3 Ejercicio *aint5*

En este caso, se ha propuesto resolver el problema de *Cubrimiento de Conjuntos* sobre un conjunto de datos de entrada de tamaño relativamente elevado, con $m = 328$ puntos de demanda y $n = 19$ puntos de servicio. Se considera que un distrito ha sido cubierto si la distancia a un punto de servicio es menor o igual que un determinado valor dc denominado distancia de cubrimiento. En este caso se ha resuelto para valores enteros comprendidos en el intervalo $[250, 400]$ mediante la estrategia *Greedy*, la estrategia *Greedy Aleatorizada* (con parámetros $k = 5$ y $n = 100$) y la de

```
forall(j in servicios) k_marca(j):=0
forall(k in k_iter) do
  aux := -99
  forall(j in servicios | (solu(j) = 0 and k_marca(j) = 0)) do
    if(aux < partial_sums(j)) then
      aux := partial_sums(j)
      jmax := j
    end-if
  end-do
  k_marca(jmax):=1
  k_list(k):=jmax
end-do
k_win := ceil(random*k_max)
selected := k_list(k_win)
```

Figura 3: Selección del punto de servicio de manera aleatoria

Solución Óptima. Los resultados obtenidos se muestran gráficamente en la figura 6 y de manera tabular en las tablas 5, 6 y 7.



Figura 4: Resultados del problema de Set Covering aplicado a los datos de los distritos de la ciudad de Nueva York

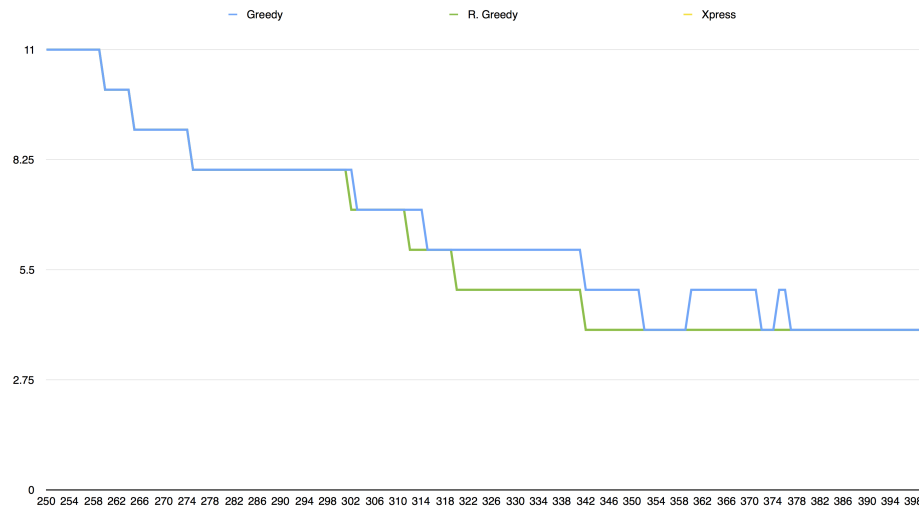


Figura 5: Resultados del problema de Set Covering aplicado a los datos aint1

DC	Greedy	Greedy Opened	Xpress	Xpress opened
70	9	1 4 5 7 9 12 13 16 18	8	3 4 5 9 12 13 26 27
71	8	1 5 7 9 13 14 16 18	7	1 3 5 9 13 14 27
72	8	1 5 7 9 13 14 16 18	7	1 3 5 9 13 14 27
73	8	1 5 7 9 13 14 16 18	7	1 3 5 9 13 14 27
74	7	1 5 7 13 14 16 18	7	1 2 3 5 13 14 27
75	7	1 5 7 13 14 16 18	6	5 6 13 14 26 27
76	7	1 5 7 13 16 18 23	6	5 6 13 23 26 27
77	7	1 5 7 13 16 18 23	6	5 6 13 23 26 30
78	6	1 5 7 13 14 27	6	5 6 13 23 26 27
79	6	1 5 7 13 14 27	6	5 6 13 23 26 27
80	6	1 5 7 13 14 27	6	5 6 13 23 26 27
81	6	1 5 7 13 14 27	6	5 6 13 23 26 27
82	6	1 5 7 13 14 27	6	5 6 13 23 26 27
83	6	1 5 7 13 14 27	6	5 6 13 23 26 27
84	6	1 5 7 13 14 27	6	5 7 13 23 26 27
85	6	1 5 7 13 14 27	6	5 6 13 23 26 27
86	7	1 5 7 13 16 18 23	6	5 6 13 23 26 27
87	7	1 5 7 13 16 18 23	6	5 10 13 16 23 26
88	7	1 5 7 13 16 18 23	6	5 7 13 23 26 27
89	7	1 5 7 13 16 18 23	6	5 7 13 23 26 27
90	7	1 3 5 13 14 16 24	6	1 5 7 13 23 27
91	6	1 5 13 14 24 27	6	1 2 5 13 23 27
92	6	1 5 7 13 14 16	5	1 5 13 23 27
93	6	1 5 7 13 14 16	5	1 5 13 23 27
94	6	1 5 10 13 16 23	5	1 5 13 23 27
95	6	1 5 10 12 13 16	5	1 5 12 13 27
96	6	1 5 10 12 13 16	5	1 5 12 13 27
97	6	1 5 10 12 13 16	5	1 5 12 13 27
98	6	1 5 10 12 13 16	5	1 5 12 13 27
99	6	1 5 10 12 13 16	5	1 5 12 13 27
100	6	1 5 10 12 13 16	5	5 12 13 20 27
101	6	1 2 3 12 13 29	5	2 12 13 20 27
102	6	1 2 7 12 13 16	5	2 12 13 27 28
103	6	1 2 7 12 13 16	5	2 12 13 27 28
104	6	1 2 7 12 13 16	5	12 13 17 20 27
105	5	2 7 12 13 16	4	2 12 13 27
106	5	2 7 12 13 16	4	2 12 25 27
107	4	4 13 17 29	4	2 12 25 27
108	4	4 13 17 29	4	2 10 12 13
109	4	4 13 17 29	4	2 10 12 13
110	4	4 13 17 29	4	2 10 12 13
111	4	2 4 13 29	4	2 12 13 27
112	4	2 3 12 13	4	2 12 13 27
113	4	2 3 12 13	4	2 12 13 27
114	4	2 3 12 13	4	2 12 13 27
115	4	2 3 12 13	4	2 12 13 27
116	4	2 3 12 13	4	2 12 13 27
117	4	2 7 12 13	4	2 12 13 27
118	4	2 7 12 13	4	2 12 13 18
119	4	2 7 12 13	4	2 12 13 18
120	4	2 7 12 13	4	2 8 22 30

Tabla 1: Resultados del problema de Set Covering aplicado a los datos de los distritos de la ciudad de Nueva York

dc	Greedy	Greedy Opened	R. Greedy	Xpress	Xpress Opened
250	11	1 2 5 8 9 11 13 14 15 17 18	11	11	1 2 3 8 9 11 13 14 15 17 18
251	11	1 2 5 8 9 11 13 14 15 17 18	11	11	1 2 3 8 9 11 13 14 15 17 18
252	11	1 2 5 8 9 11 14 15 16 17 18	11	11	1 2 3 8 9 11 14 15 16 17 18
253	11	1 2 5 8 9 11 14 15 16 17 18	11	11	1 2 3 8 9 11 14 15 16 17 18
254	11	1 2 5 8 9 11 14 15 16 17 18	11	11	1 2 3 8 9 11 14 15 16 17 18
255	11	1 2 5 8 9 11 14 15 16 17 18	11	11	1 2 3 8 9 11 14 15 16 17 18
256	11	1 2 5 8 9 11 14 15 16 17 18	11	11	1 2 3 8 9 11 14 15 16 17 18
257	11	1 2 8 9 11 14 15 16 17 18 22	11	11	1 2 3 8 9 11 14 15 16 17 18
258	11	1 2 8 9 11 14 15 16 17 18 22	11	11	1 2 3 8 9 11 14 15 16 17 18
259	11	1 2 8 9 11 14 15 16 17 18 22	11	11	1 2 3 8 9 11 14 15 16 17 18
260	10	1 2 8 9 11 14 16 17 18 22	10	10	1 2 3 8 9 11 14 16 17 18
261	10	1 2 8 9 11 14 16 17 18 22	10	10	1 2 3 8 9 11 14 16 17 18
262	10	1 2 8 9 11 14 16 17 18 22	10	10	1 2 3 8 9 11 14 16 17 18
263	10	1 2 8 9 11 14 16 17 18 22	10	10	1 2 3 8 9 11 14 16 17 18
264	10	1 2 8 9 11 14 16 17 18 22	10	10	1 2 3 8 9 11 14 16 17 18
265	9	1 2 3 8 9 11 14 17 18	9	9	1 2 3 8 9 11 14 17 18
266	9	1 2 3 8 9 11 14 17 18	9	9	1 2 3 8 9 11 14 17 18
267	9	1 2 3 8 9 11 14 17 18	9	9	1 2 3 8 9 11 14 17 18
268	9	1 2 3 8 9 11 14 17 18	9	9	1 2 7 8 9 11 14 17 18
269	9	1 2 3 8 9 11 14 17 18	9	9	1 2 7 8 9 11 14 17 18
270	9	1 2 3 8 9 11 14 17 18	9	9	1 2 4 8 9 11 14 17 18
271	9	1 2 3 8 9 11 14 17 18	9	9	1 2 4 8 9 11 14 17 18
272	9	1 2 3 8 9 11 14 17 18	9	9	1 2 4 8 9 11 14 17 18
273	9	1 2 3 9 11 14 16 17 18	9	9	1 2 4 9 11 14 16 17 18
274	9	1 2 3 9 11 14 16 17 18	9	9	1 2 4 9 11 14 16 17 18
275	8	1 2 7 9 11 14 17 18	8	8	1 2 5 9 11 14 17 18
276	8	1 2 3 9 11 14 17 18	8	8	1 2 5 9 11 14 17 18
277	8	1 2 3 9 11 14 17 18	8	8	1 2 5 9 11 14 17 18
278	8	1 2 3 9 11 14 17 18	8	8	1 2 5 9 11 14 17 18
279	8	1 2 3 9 11 14 17 18	8	8	1 2 5 9 11 14 17 18
280	8	1 2 3 9 11 14 17 18	8	8	1 2 5 9 11 14 17 18
281	8	1 2 3 9 11 14 17 18	8	8	1 2 5 9 11 14 17 18
282	8	1 2 3 9 11 14 17 18	8	8	1 2 7 9 14 17 18 19
283	8	1 2 3 9 11 14 17 18	8	8	1 2 7 9 14 17 18 19
284	8	1 2 3 9 11 14 17 18	8	8	1 2 7 9 14 17 18 19
285	8	1 2 3 9 11 14 17 18	8	8	1 3 9 12 14 17 18 19
286	8	1 2 3 9 11 14 17 18	8	8	1 3 9 12 14 17 18 19
287	8	1 2 3 9 11 14 17 18	8	8	1 9 12 14 17 18 19 20
288	8	1 2 3 9 11 14 17 18	8	8	1 9 12 14 17 18 19 20
289	8	1 2 3 9 11 14 17 18	8	8	1 9 12 14 17 18 19 20
290	8	1 2 3 9 11 14 17 18	8	8	1 9 12 14 17 18 19 20
291	8	2 3 9 11 14 16 17 18	8	8	9 10 12 14 17 18 19 20
292	8	1 2 9 11 14 17 18 22	8	8	1 9 11 12 14 17 18 20
293	8	1 2 9 11 14 17 18 20	8	8	1 9 11 12 14 17 18 20
294	8	1 2 9 11 14 17 18 20	8	8	1 9 11 12 14 17 18 20
295	8	1 2 9 11 14 17 18 22	8	8	1 9 11 12 14 17 18 20
296	8	1 2 9 11 14 17 18 22	8	8	1 2 9 11 14 17 18 20
297	8	1 2 9 11 14 17 18 22	8	8	1 9 11 12 14 17 18 20
298	8	1 2 9 11 14 17 18 22	8	8	9 11 12 14 16 17 18 20
299	8	1 2 9 11 14 17 18 22	8	8	2 9 11 14 16 17 18 20

Tabla 2: Resultados del problema de Set Covering aplicado a los datos aint1

dc	Greedy	Greedy Opened	R. Greedy	Xpress	Xpress Opened
300	8	1 2 7 9 11 14 17 18	8	8	2 7 9 11 14 16 17 18
301	8	1 2 7 9 11 14 17 18	8	8	2 7 9 11 14 16 17 18
302	8	1 2 3 6 11 13 17 18	7	7	2 7 9 11 16 17 18
303	7	1 2 3 9 11 17 18	7	7	2 5 9 11 16 17 18
304	7	1 2 3 9 11 17 18	7	7	2 9 11 13 16 17 18
305	7	1 2 3 9 11 17 18	7	7	2 9 11 13 16 17 18
306	7	1 2 3 9 11 17 18	7	7	2 9 11 13 16 17 18
307	7	1 2 3 9 11 17 18	7	7	2 9 11 13 16 17 18
308	7	1 2 3 9 11 17 18	7	7	2 9 11 13 16 17 18
309	7	1 2 3 9 11 17 18	7	7	2 9 11 13 16 17 18
310	7	1 2 3 9 11 17 18	7	7	9 11 12 13 16 17 18
311	7	1 2 3 9 11 17 18	7	7	2 9 11 13 16 17 18
312	7	1 2 3 9 11 17 18	6	6	2 11 13 16 17 18
313	7	1 2 3 9 11 17 18	6	6	2 11 13 16 17 18
314	7	1 2 3 9 11 17 18	6	6	2 11 13 16 17 18
315	6	2 6 11 13 17 18	6	6	2 6 9 14 17 21
316	6	2 6 11 13 17 18	6	6	2 9 11 16 17 18
317	6	2 6 11 13 17 18	6	6	2 6 11 13 17 18
318	6	2 6 11 13 17 18	6	6	2 6 11 13 17 18
319	6	2 6 9 11 17 18	6	6	2 9 11 16 17 18
320	6	2 6 9 11 17 18	5	5	2 11 16 17 18
321	6	2 6 9 11 17 18	5	5	2 11 16 17 18
322	6	2 6 9 11 17 18	5	5	2 11 16 17 18
323	6	2 6 9 11 17 18	5	5	2 11 16 17 18
324	6	2 6 9 11 17 18	5	5	2 11 16 17 18
325	6	2 3 6 9 17 18	5	5	2 11 16 17 18
326	6	2 3 6 9 17 18	5	5	2 11 16 17 18
327	6	2 6 9 11 17 18	5	5	2 11 16 17 18
328	6	2 6 9 11 17 18	5	5	11 12 16 17 18
329	6	2 6 9 11 17 18	5	5	11 12 16 17 18
330	6	2 6 9 11 17 18	5	5	11 12 16 17 18
331	6	2 6 9 11 17 18	5	5	2 11 16 17 18
332	6	2 6 9 11 17 18	5	5	2 11 16 17 18
333	6	2 6 9 11 17 18	5	5	2 11 16 17 18
334	6	2 6 9 11 17 18	5	5	2 11 16 17 18
335	6	2 6 9 11 17 18	5	5	2 11 16 17 18
336	6	2 6 9 11 17 18	5	5	2 11 16 17 18
337	6	2 6 9 11 17 18	5	5	2 11 16 17 18
338	6	2 6 9 11 17 18	5	5	2 11 16 17 18
339	6	2 6 9 11 17 18	5	5	11 12 16 17 18
340	6	2 6 9 11 17 18	5	5	11 12 16 17 18
341	6	2 6 9 11 17 18	5	5	11 12 16 17 18
342	5	3 6 9 17 18	4	4	11 16 17 18
343	5	3 6 9 17 18	4	4	11 16 17 18
344	5	3 6 9 17 18	4	4	11 16 17 18
345	5	3 6 9 17 18	4	4	11 16 17 18
346	5	3 6 9 17 18	4	4	11 16 17 18
347	5	3 6 9 17 18	4	4	11 16 17 18
348	5	3 6 9 17 18	4	4	11 16 17 18
349	5	3 6 9 17 18	4	4	11 16 17 18

Tabla 3: Resultados del problema de Set Covering aplicado a los datos aint1

dc	Greedy	Greedy Opened	R. Greedy	Xpress	Xpress Opened
350	5	1 11 13 17 18	4	4	11 16 17 18
351	5	1 11 13 17 18	4	4	11 16 17 18
352	4	1 11 17 18	4	4	11 16 17 18
353	4	1 11 17 18	4	4	11 16 17 18
354	4	1 11 17 18	4	4	11 16 17 18
355	4	1 11 17 18	4	4	11 16 17 18
356	4	1 11 17 18	4	4	6 11 17 18
357	4	1 11 17 18	4	4	6 11 17 18
358	4	1 11 17 18	4	4	6 11 17 18
359	4	6 11 17 18	4	4	6 11 17 18
360	5	6 8 11 18 20	4	4	6 11 17 18
361	5	6 8 11 18 20	4	4	6 11 17 18
362	5	6 8 11 18 20	4	4	6 11 17 18
363	5	6 8 11 18 20	4	4	6 11 17 18
364	5	6 8 11 18 20	4	4	6 11 17 18
365	5	6 8 11 18 20	4	4	6 11 17 18
366	5	6 8 11 18 20	4	4	6 11 17 18
367	5	6 8 11 18 20	4	4	6 11 17 18
368	5	6 8 11 18 20	4	4	6 11 17 18
369	5	6 8 11 18 20	4	4	6 11 17 18
370	5	6 8 11 18 20	4	4	6 11 17 18
371	5	6 8 11 18 20	4	4	6 11 17 18
372	4	1 8 11 18	4	4	6 11 17 18
373	4	1 8 11 18	4	4	6 11 17 18
374	4	1 8 11 18	4	4	6 11 17 18
375	5	6 8 11 18 20	4	4	6 11 17 18
376	5	6 8 11 18 20	4	4	6 11 17 18
377	4	6 8 11 18	4	4	6 11 17 18
378	4	6 8 11 18	4	4	6 11 17 18
379	4	6 8 11 18	4	4	6 11 17 18
380	4	1 8 11 18	4	4	6 11 17 18
381	4	1 8 11 18	4	4	6 11 17 18
382	4	1 8 11 18	4	4	6 11 17 18
383	4	1 8 11 18	4	4	6 11 17 18
384	4	1 8 11 18	4	4	6 11 17 18
385	4	1 8 11 18	4	4	6 11 17 18
386	4	1 8 11 18	4	4	6 11 17 18
387	4	1 8 11 18	4	4	6 11 17 18
388	4	1 8 11 18	4	4	6 11 17 18
389	4	1 8 11 18	4	4	6 11 17 18
390	4	1 8 11 18	4	4	6 11 17 18
391	4	1 8 11 18	4	4	6 11 17 18
392	4	1 8 11 18	4	4	1 11 17 18
393	4	1 8 11 18	4	4	1 11 17 18
394	4	1 8 11 18	4	4	1 11 17 18
395	4	1 8 11 18	4	4	1 11 17 18
396	4	1 8 11 18	4	4	1 11 17 18
397	4	1 8 11 18	4	4	1 11 17 18
398	4	1 8 11 18	4	4	1 11 17 18
399	4	1 8 11 18	4	4	1 8 11 18
400	4	1 8 11 18	4	4	1 8 11 18

Tabla 4: Resultados del problema de Set Covering aplicado a los datos aint1

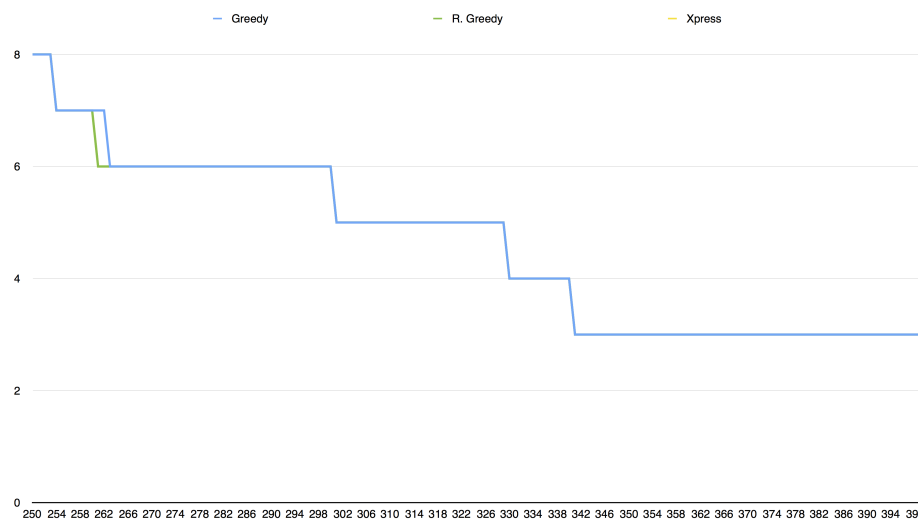


Figura 6: Resultados del problema de Set Covering aplicado a los datos aint5

dc	Greedy	Greedy Opened	R. Greedy	R. Greedy Opened	Xpress	Xpress Opened
250	8	1 2 4 5 6 9 13 17	8	1 2 4 5 6 9 13 17	8	1 2 4 5 6 9 13 17
251	8	1 2 4 5 6 9 13 17	8	1 2 4 5 6 7 13 17	8	1 2 4 5 6 7 13 17
252	8	1 2 4 5 6 9 13 17	8	1 2 4 5 6 9 13 17	8	1 2 4 5 6 7 13 17
253	8	1 2 4 5 6 9 13 17	8	1 2 4 5 6 9 13 17	8	1 2 4 5 6 9 13 17
254	7	1 2 5 6 7 13 17	7	1 2 6 9 13 15 17	7	1 2 5 6 9 13 17
255	7	1 2 5 6 7 13 17	7	1 2 6 9 13 15 17	7	1 2 5 6 9 13 17
256	7	1 2 5 6 7 13 17	7	1 2 6 9 13 15 17	7	1 2 5 6 9 13 17
257	7	1 2 5 6 7 13 17	7	1 2 5 6 7 13 17	7	1 2 5 6 9 13 17
258	7	1 2 5 6 7 13 17	7	1 2 6 9 13 15 17	7	1 2 5 6 9 13 17
259	7	1 2 5 6 7 13 17	7	1 2 5 6 9 13 17	7	1 2 5 6 9 13 17
260	7	1 2 3 5 6 13 17	7	1 2 5 6 9 13 17	7	1 2 5 6 9 13 17
261	7	1 2 3 5 6 13 17	6	1 2 6 9 13 17	6	1 2 6 9 13 17
262	7	1 2 3 5 6 13 17	6	1 2 6 9 13 17	6	1 2 6 9 13 17
263	6	1 2 3 6 13 17	6	1 2 6 9 13 17	6	1 2 3 6 13 17
264	6	1 2 3 6 13 17	6	1 2 3 6 13 17	6	1 2 3 6 13 17
265	6	1 2 3 6 13 17	6	1 2 6 9 13 17	6	1 2 3 6 13 17
266	6	1 2 3 6 13 17	6	1 2 6 9 13 17	6	1 2 3 6 13 17
267	6	1 2 3 6 13 17	6	1 2 3 6 13 17	6	1 2 3 6 13 17
268	6	1 2 3 6 13 17	6	1 2 3 6 13 17	6	1 2 3 6 13 17
269	6	1 2 3 6 13 17	6	1 2 6 9 13 17	6	1 2 3 6 13 17
270	6	1 2 3 6 13 17	6	1 2 3 6 13 17	6	1 2 3 6 13 17
271	6	1 2 3 6 8 13	6	1 2 6 9 13 17	6	1 2 3 6 13 17
272	6	1 2 3 6 8 13	6	1 2 6 9 11 13	6	1 2 3 6 8 13
273	6	1 2 3 6 8 13	6	1 2 3 6 8 13	6	1 2 3 6 8 13
274	6	1 2 3 6 8 13	6	1 2 3 6 8 13	6	1 2 3 6 8 13
275	6	1 2 3 6 8 13	6	1 2 6 9 11 13	6	1 2 3 6 8 13
276	6	1 2 3 6 8 13	6	1 2 3 6 8 13	6	1 2 3 6 8 13
277	6	1 2 3 6 8 13	6	1 2 3 6 8 13	6	1 2 3 6 8 13
278	6	1 2 3 6 8 13	6	1 2 6 9 13 17	6	1 2 3 6 8 13
279	6	1 2 3 6 8 13	6	1 2 3 6 8 13	6	1 2 3 6 8 13
280	6	1 2 3 6 8 13	6	1 2 3 6 8 13	6	1 2 3 6 8 13
281	6	1 2 3 6 8 13	6	1 2 3 6 8 13	6	1 2 3 6 8 13
282	6	1 2 3 4 8 13	6	1 2 3 6 8 13	6	1 2 3 4 8 13
283	6	1 2 3 4 8 13	6	1 2 4 9 11 13	6	1 2 4 5 11 13
284	6	1 2 3 4 8 13	6	1 2 6 9 11 13	6	1 2 4 5 11 13
285	6	1 2 3 4 8 13	6	1 2 3 6 8 13	6	1 2 3 4 8 13
286	6	1 2 3 4 8 13	6	1 2 3 6 11 13	6	1 2 3 4 8 13
287	6	1 2 3 4 8 13	6	1 2 3 6 8 13	6	1 2 3 4 8 13
288	6	1 2 3 4 13 14	6	1 2 3 6 8 13	6	1 2 3 4 8 13
289	6	1 2 3 4 13 14	6	1 2 3 6 13 14	6	1 2 3 4 8 13
290	6	1 2 3 4 13 14	6	1 2 3 4 13 14	6	1 2 3 4 8 13
291	6	1 2 3 4 13 14	6	1 2 3 6 13 17	6	1 2 3 4 8 13
292	6	1 2 3 4 13 14	6	1 2 3 6 13 14	6	1 2 3 4 11 13
293	6	1 2 3 4 13 14	6	1 2 3 6 8 13	6	1 2 3 4 11 13
294	6	1 2 3 4 13 14	6	1 2 3 6 11 13	6	1 2 3 4 11 13
295	6	1 2 3 4 13 14	6	1 2 6 9 11 13	6	1 2 3 4 11 13
296	6	1 2 3 4 13 14	6	1 2 3 6 13 14	6	1 2 3 4 11 13
297	6	1 2 3 4 13 14	6	1 2 3 6 13 14	6	1 2 3 4 11 13
298	6	1 2 3 4 13 14	6	1 2 3 6 13 14	6	1 2 3 4 11 13
299	6	1 2 3 4 13 14	6	1 2 3 6 13 14	6	1 2 3 4 11 13

Tabla 5: Resultados del problema de Set Covering aplicado a los datos aint5

dc	Greedy	Greedy Opened	R. Greedy	R. Greedy Opened	Xpress	Xpress Opened
300	6	1 2 3 4 13 14	6	1 2 3 6 8 13	6	1 2 3 4 11 13
301	5	1 2 3 6 14	5	1 2 3 6 17	5	1 2 3 6 11
302	5	1 2 3 6 14	5	1 2 3 6 17	5	1 2 3 6 11
303	5	1 2 3 6 14	5	1 2 3 6 8	5	1 2 3 6 11
304	5	1 2 3 6 14	5	1 2 3 6 11	5	1 2 3 6 11
305	5	1 2 3 6 14	5	1 2 3 6 8	5	1 2 3 6 11
306	5	1 2 3 6 14	5	1 2 3 6 14	5	1 2 3 6 11
307	5	1 2 3 6 14	5	1 2 3 6 11	5	1 2 3 6 11
308	5	1 2 3 6 14	5	1 2 3 6 14	5	1 2 3 6 11
309	5	1 2 3 6 14	5	1 2 3 6 14	5	1 2 3 6 11
310	5	1 2 3 6 14	5	1 2 3 6 8	5	1 2 3 6 11
311	5	1 2 3 6 14	5	1 2 3 6 11	5	1 2 3 6 11
312	5	1 2 3 6 12	5	1 2 3 6 12	5	1 2 3 6 11
313	5	1 2 3 6 12	5	1 2 3 6 14	5	1 2 3 6 11
314	5	1 2 3 6 12	5	1 2 3 6 11	5	1 2 3 6 11
315	5	1 2 3 6 12	5	1 2 3 6 14	5	1 2 3 6 11
316	5	1 2 3 6 12	5	1 2 3 6 12	5	1 2 3 6 11
317	5	1 2 3 6 12	5	1 2 3 6 12	5	1 2 3 6 11
318	5	1 2 3 6 12	5	1 2 3 6 14	5	1 2 3 6 11
319	5	1 2 3 6 8	5	1 2 3 6 12	5	1 2 3 6 11
320	5	1 2 3 6 8	5	1 2 5 6 11	5	1 2 3 6 11
321	5	1 3 6 8 16	5	1 5 6 11 16	5	1 3 6 11 16
322	5	1 3 6 8 16	5	1 2 5 6 11	5	1 3 6 11 16
323	5	1 3 6 8 16	5	1 6 9 16 18	5	1 3 6 11 16
324	5	1 3 6 8 16	5	1 3 6 8 16	5	1 3 6 11 16
325	5	1 2 3 6 14	5	1 3 6 14 16	5	1 3 6 11 16
326	5	1 3 6 8 19	5	1 6 7 11 19	5	1 3 6 11 19
327	5	1 3 6 8 19	5	1 6 9 11 16	5	1 3 6 11 19
328	5	1 3 6 8 19	5	1 3 6 14 19	5	1 3 6 11 19
329	5	1 3 6 8 19	5	1 3 6 8 19	5	1 3 6 11 19
330	4	1 3 6 19	4	1 3 6 19	4	1 3 6 19
331	4	1 3 6 19	4	1 3 6 19	4	1 3 6 19
332	4	1 3 4 19	4	1 3 4 19	4	1 3 6 19
333	4	1 3 4 19	4	1 3 6 19	4	1 3 6 19
334	4	1 3 4 19	4	1 3 4 19	4	1 3 6 19
335	4	1 3 4 19	4	1 3 4 19	4	1 3 6 19
336	4	1 3 4 19	4	1 3 4 19	4	1 3 6 19
337	4	1 3 4 19	4	1 3 4 19	4	1 3 6 19
338	4	1 3 4 19	4	1 3 6 19	4	1 3 6 19
339	4	1 3 4 19	4	1 3 6 19	4	1 3 6 19
340	4	1 3 4 19	4	1 3 6 19	4	1 3 6 19
341	3	3 4 19	3	3 4 19	3	3 4 19
342	3	3 4 19	3	3 4 19	3	3 4 19
343	3	3 4 19	3	3 4 19	3	3 4 19
344	3	3 4 19	3	3 4 19	3	3 4 19
345	3	3 4 14	3	4 13 14	3	1 13 14
346	3	3 4 14	3	4 13 14	3	1 13 14
347	3	3 4 14	3	1 13 14	3	1 13 14
348	3	3 4 14	3	3 4 14	3	1 13 14
349	3	3 4 14	3	4 13 14	3	3 4 14

Tabla 6: Resultados del problema de Set Covering aplicado a los datos aint5

dc	Greedy	Greedy Opened	R. Greedy	R. Greedy Opened	Xpress	Xpress Opened
350	3	3 4 10	3	3 4 10	3	3 4 14
351	3	3 4 10	3	1 13 19	3	3 4 14
352	3	3 4 10	3	1 13 19	3	3 4 14
353	3	3 4 10	3	3 4 14	3	3 4 14
354	3	3 4 10	3	3 4 19	3	3 4 14
355	3	3 4 10	3	3 4 19	3	3 4 14
356	3	3 4 10	3	4 13 19	3	3 4 14
357	3	3 4 10	3	3 4 10	3	3 4 14
358	3	3 4 10	3	4 12 13	3	3 4 14
359	3	3 4 14	3	4 13 14	3	3 4 14
360	3	3 4 10	3	1 12 13	3	3 4 14
361	3	3 4 10	3	3 4 12	3	3 4 14
362	3	3 4 14	3	3 4 14	3	3 4 14
363	3	3 4 14	3	4 12 13	3	3 4 14
364	3	3 4 14	3	1 10 13	3	1 13 14
365	3	3 4 14	3	3 4 10	3	1 13 14
366	3	3 4 14	3	4 12 13	3	1 13 14
367	3	3 4 14	3	3 4 16	3	1 13 14
368	3	3 4 14	3	1 12 13	3	1 13 14
369	3	3 4 14	3	3 4 14	3	1 13 14
370	3	3 4 14	3	1 13 19	3	1 13 14
371	3	3 4 14	3	4 13 14	3	1 13 14
372	3	3 4 14	3	4 10 13	3	1 13 14
373	3	3 4 12	3	1 12 13	3	1 13 14
374	3	3 4 12	3	1 13 14	3	1 13 14
375	3	3 4 12	3	4 10 13	3	1 13 14
376	3	3 4 12	3	4 10 13	3	1 13 14
377	3	3 4 12	3	3 4 19	3	1 13 14
378	3	3 4 8	3	4 8 13	3	1 8 13
379	3	3 4 8	3	4 13 16	3	1 8 13
380	3	3 4 8	3	4 10 18	3	1 8 13
381	3	3 4 8	3	1 12 13	3	1 8 13
382	3	3 4 8	3	1 10 13	3	1 8 13
383	3	3 4 8	3	3 4 14	3	1 8 13
384	3	3 4 8	3	3 4 8	3	1 8 13
385	3	3 4 8	3	7 8 13	3	3 4 8
386	3	3 4 8	3	1 8 13	3	3 4 8
387	3	3 4 8	3	1 12 13	3	3 4 8
388	3	3 4 8	3	3 4 12	3	3 4 8
389	3	3 4 8	3	7 12 13	3	3 4 8
390	3	3 4 8	3	1 8 13	3	3 4 8
391	3	3 4 8	3	4 12 13	3	3 4 8
392	3	3 4 8	3	4 12 13	3	3 4 8
393	3	3 4 8	3	4 12 13	3	3 4 8
394	3	3 4 8	3	7 13 16	3	3 4 8
395	3	4 8 9	3	4 8 9	3	3 4 8
396	3	4 8 9	3	1 8 13	3	3 4 8
397	3	4 8 9	3	4 8 9	3	7 13 14
398	3	4 8 9	3	4 13 16	3	7 13 14
399	3	4 8 9	3	7 13 14	3	7 13 14
400	3	4 8 9	3	7 8 13	3	7 13 14

Tabla 7: Resultados del problema de Set Covering aplicado a los datos aint5

7. MAX-COVERING PROBLEM

En esta sección se trata el *problema de cubrimiento máximo* o *max covering problem*. El problema consiste en lo siguiente: Sea m el número de puntos de demandas y n el de puntos de servicio. El objetivo se trata de maximizar el beneficio h_i obtenido de cubrir el i -ésimo punto de demanda. Para modelizar dicho cubrimiento se utiliza la variable binaria z_i . Para representar los puntos de servicio utilizados se utiliza la variable de tipo binario x_j . La motivación del problema consiste en encontrar el conjunto de variables x_j con cardinalidad máxima denominada por p y prefijada previamente, que maximice la ganancia debida al cubrimiento de los puntos de servicio z_i . El modelo formal se muestra en la ecuación (2).

$$\begin{aligned}
 &\text{Maximizar} && \sum_{i=1}^m h_i z_i \\
 &\text{sujeto a} && \sum_{j \in N_i} x_j \geq z_i, \quad i = 1, \dots, m \\
 &&& \sum_{j=1}^n x_j \leq p, \\
 &&& x_j \in \{0, 1\}, \quad j = 1, \dots, n \\
 &&& z_i \in \{0, 1\}, \quad i = 1, \dots, m
 \end{aligned} \tag{2}$$

Ecuación 2: *Formulación de Max-Covering Problem.*

[TODO describir heurísticas]

[TODO describir heurísticas]

7.1 Ejercicio *aint1*

En este caso, se ha propuesto resolver el problema de *Cubrimiento Máximo* sobre un conjunto de datos de entrada de tamaño relativamente elevado, con $m = 356$ puntos de demanda y $n = 22$ puntos de servicio. Se considera que un distrito ha sido cubierto si la distancia a un punto de servicio es menor o igual que un determinado valor dc denominado distancia de cubrimiento. En este caso se ha resuelto para $dc = 200$ y un número de puntos de servicios restringido a $p = [1, 6]$, mediante la estrategia *Greedy*, la estrategia *Greedy Aleatorizada* (con parámetros $k = 5$ y $n = 100$) y la de *Solución Óptima*. Los resultados obtenidos se muestran gráficamente en la figura 9 y de manera tabular en las tablas 8.

p	Greedy	Greedy Opened	R. Greedy	R. Greedy Opened	Xpress	Xpress Opened
1	41.1322	18	41.1322	18	41.1322	18
2	53.2281	11 18	53.2281	11 18	53.2281	11 18
3	65.1815	6 11 18	65.1815	6 11 18	65.1815	6 11 18
4	75.0729	2 6 11 18	76.6026	1 2 19 22	76.6026	1 2 19 22
5	81.1139	2 6 8 11 18	82.6436	1 2 8 19 22	82.6436	1 2 8 19 22
6	84.665	2 6 8 9 11 18	88.3181	1 2 8 14 19 22	88.3181	1 2 8 14 19 22

Tabla 8: *Resultados del problema de Max Covering aplicado a los datos aint1*

```
nfilcub := 0
while(nfilcub < m) do
  forall(j in servicios | fijada(j) = 0) do
    d(j) := 0
    forall(i in demandas) do
      if(cubierta(i) = 0 and a(i,j) = 1) then
        d(j) := d(j) + 1
      end-if
    end-do
    if(d(j) = 0) then
      fijada(j) := 1
      solu(j) := 0
    end-if
  end-do
  aux := -99
  forall(j in servicios | fijada(j) = 0) do
    if(d(j) > aux) then
      aux := d(j)
      jmax := j
    end-if
  end-do
  solu(jmax) := 1
  fijada(jmax) := 1
  forall(i in demandas) do
    if(cubierta(i) = 0 and a(i, jmax) = 1) then
      cubierta(i) := 1
      nfilcub := nfilcub + 1
    end-if
  end-do
end-do
```

Figura 7: [TODO]

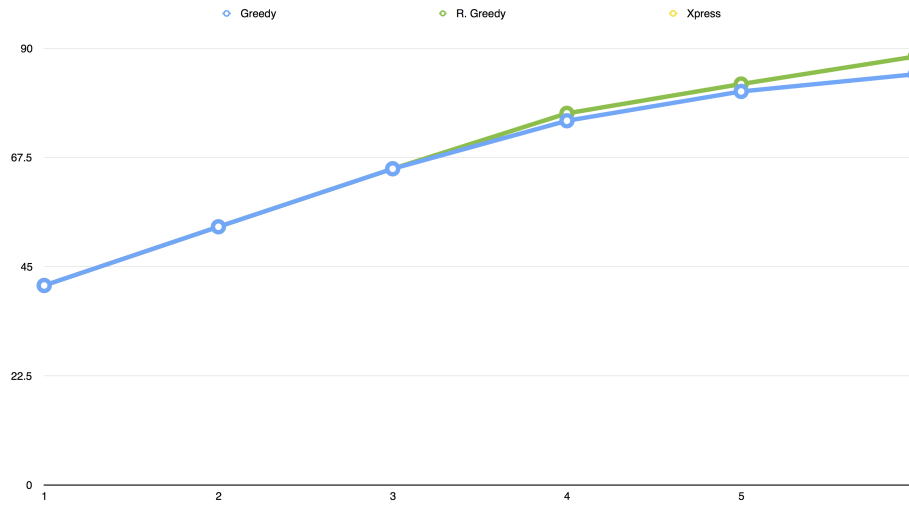
7.2 Ejercicio *aint5*

En este caso, se ha propuesto resolver el problema de *Cubrimiento Máximo* sobre un conjunto de datos de entrada de tamaño relativamente elevado, con $m = 328$ puntos de demanda y $n = 19$ puntos de servicio. Se considera que un distrito ha sido cubierto si la distancia a un punto de servicio es menor o igual que un determinado valor dc denominado distancia de cubrimiento. En este caso se ha resuelto para $dc = 200$ y un número de puntos de servicios restringido a $p = [1, 6]$, mediante la estrategia *Greedy*, la estrategia *Greedy Aleatorizada* (con parámetros $k = 5$ y $n = 100$) y la de *Solución Óptima*. Los resultados obtenidos se muestran gráficamente en la figura 10 y de manera tabular en las tablas 9.

```

forall(j in servicios) k_marca(j):=0
forall(k in k_iter) do
  aux := -99
  forall(j in servicios | (solu(j) = 0 and k_marca(j) = 0)) do
    if(aux < partial_sums(j)) then
      aux := partial_sums(j)
      jmax := j
    end-if
  end-do
  k_marca(jmax):=1
  k_list(k):=jmax
end-do
k_win := ceil(random*k_max)
selected := k_list(k_win)

```

Figura 8: *[TODO]***Figura 9:** Resultados del problema de Max Covering aplicado a los datos aint1

p	Greedy	Greedy Opened	R. Greedy	R. Greedy Opened	Xpress	Xpress Opened
1	51.9056	17	51.9056	17	51.9056	17
2	68.8971	17 18	68.8971	17 18	68.8971	17 18
3	77.9532	2 17 18	77.9532	2 17 18	77.9532	2 17 18
4	85.5581	2 4 17 18	85.5581	2 4 17 18	85.5581	2 4 17 18
5	90.7717	2 4 10 17 18	90.7717	2 4 10 17 18	90.7717	2 4 10 17 18
6	92.7893	2 4 9 10 17 18	93.5694	4 11 12 17 18 19	93.5694	4 11 12 17 18 19

Tabla 9: Resultados del problema de Max Covering aplicado a los datos aint5

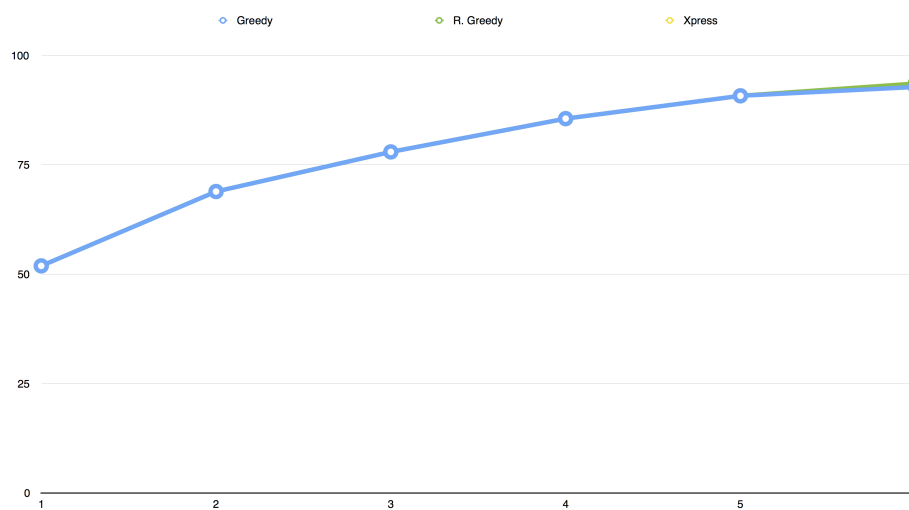


Figura 10: Resultados del problema de Max Covering aplicado a los datos aint5

8. P-MEDIAN PROBLEM

En el caso del problema de la p-mediana, modelizado matemáticamente en la ecuación 3, el objetivo es minimizar la distancia global de cada uno de los p puntos de servicio abiertos al conjunto global de puntos de demanda, de manera que los j puntos de servicio abierto mantengan la menor distancia en promedio a los i puntos de demanda.

En este problema, al igual que en los anteriores, se utiliza un vector de demanda denominado h , que en la componente h_i almacena la demanda necesaria por el punto de demanda i . También existe una matriz de distancias de d , que en la posición d_{ij} recoge la distancia del punto de demanda i al punto de servicio j .

Para resolver este problema, además de las variables de decisión x_j utilizadas en casos anteriores, que representan que el punto de servicio j está activo, se añaden las variables y_{ij} , que representan que el punto de demanda i es servido por el punto de servicio j , lo que conlleva que en esta modelización cada punto de demanda sea servido únicamente por un único servicio.

$$\begin{aligned}
 &\text{Minimizar} && \sum_{i=1}^m \sum_{j=1}^n h_i d_{ij} y_{ij} \\
 &\text{sujeto a} && \sum_{j=1}^n y_{ij} = 1, && i = 1, \dots, m \\
 &&& y_{ij} \leq x_j, && i = 1, \dots, m, j = 1, \dots, n \\
 &&& \sum_{j=1}^n x_j = p, \\
 &&& x_j \in \{0, 1\}, && j = 1, \dots, n \\
 &&& y_{ij} \in \{0, 1\}, && i = 1, \dots, m, j = 1, \dots, n
 \end{aligned} \tag{3}$$

Ecuación 3: *Formulación de P-Median Problem.*

[TODO describir heurísticas]

[TODO describir heurísticas]

8.1 Ejercicio *coordenadas_15*

Este ejercicio tiene como novedad respecto de los anteriores la siguiente característica: En este caso los datos de entrada no se presentan a partir de la matriz de distancias, tal y como sucedía en el resto, sino que se suministran las coordenadas x e y de cada localización. Esto hace que el problema permita una mayor versatilidad en el sentido de calcular un mayor número de resultados, pero a la vez añade la complicación de requerir el cálculo de las distancias entre puntos.

Para la tarea de calcular las distancias se ha utilizado la *distancia euclídea* para espacios de 2 dimensiones (x, y) , que se define matemáticamente como $d(p, q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$. Por lo tanto, para la modelización del problema de la p-mediana, es necesario calcular dicha medida para todas las posibles combinaciones de localizaciones, de tal manera que la matriz d sea construida siguiendo la expresión $d_{ij} = d(l_i, l_j)$ donde l_i y l_j representan las coordenadas de las localizaciones i y j respectivamente.

```
n_cub := 0
while(n_cub < p) do
  best := MAX_INT
  ndx := n
  forall(i in nodes | fijada(i) = 0) do
    val := 0
    forall(j in nodes) do
      fijada(i) := 1
      temp_best := MAX_INT
      forall(k in nodes | fijada(k) = 1 and d(k,j) < temp_best) do
        temp_best := d(k,j)
      end-do
      fijada(i) := 0
      val := val + temp_best
    end-do
    if (val < best) then
      best := val
      ndx := i
    end-if
  end-do
  fijada(ndx) := 1
  solu(ndx) := 1
  n_cub := n_cub + 1
end-do
```

Figura 11: *[TODO]*

En esta sección se resuelve el problema de la *P Mediana* mediante la estrategia *Greedy*, la estrategia de *Búsqueda Local* y la de *Solución Óptima*. El conjunto de datos está compuesto por $n = m = 15$ poblaciones, para las cuales se pide resolver el problema para $p = [1, 10] \in N$. Dichos resultados se muestran gráficamente en la figura 13 y de manera tabular en la tabla 10. Además, se proporciona la solución gráfica en la figura 14.

8.2 Ejercicio *coordenadas_30*

En esta sección se resuelve el problema de la *P Mediana* mediante la estrategia *Greedy*, la estrategia de *Búsqueda Local* y la de *Solución Óptima*. El conjunto de datos está compuesto por $n = m = 30$ poblaciones, para las cuales se pide resolver el problema para $p = [1, 10] \in N$. Dichos resultados se muestran gráficamente en la figura 15 y de manera tabular en la tabla 11. Además, se proporciona la solución gráfica en la figura 16.

8.3 Ejercicio *coordenadas_100*

En esta sección se resuelve el problema de la *P Mediana* mediante la estrategia *Greedy*, la estrategia de *Búsqueda Local* y la de *Solución Óptima*. El conjunto de datos está compuesto por $n = m = 100$ poblaciones, para las cuales se pide resolver el problema para $p = [1, 10] \in N$. Dichos resultados se muestran gráficamente en la figura 17 y de manera tabular en la tabla 12. Además, se proporciona la solución gráfica en la figura 18.

```

final := 0
while(final = 0) do
  d_total_final := sum(i in nodes) min(j in nodes | solu(j) = 1) d(i,j)
  mejora_max := 0
  forall(j,k in nodes | solu(j) = 1 and solu(k) = 0) do
    solu(j) := 0
    solu(k) := 1
    d_total_parcial := sum(ii in nodes) min(jj in nodes | solu(jj) = 1) d(ii,jj)
    mejora := d_total_final - d_total_parcial
    if(mejora > mejora_max) then
      mejora_max := mejora
      j_max := j
      k_max := k
    end-if
    solu(j) := 1
    solu(k) := 0
  end-do
  if (mejora_max = 0) then
    final := 1
  else
    solu(j_max) := 0
    solu(k_max) := 1
  end-if
end-do

```

Figura 12: *[TODO]*

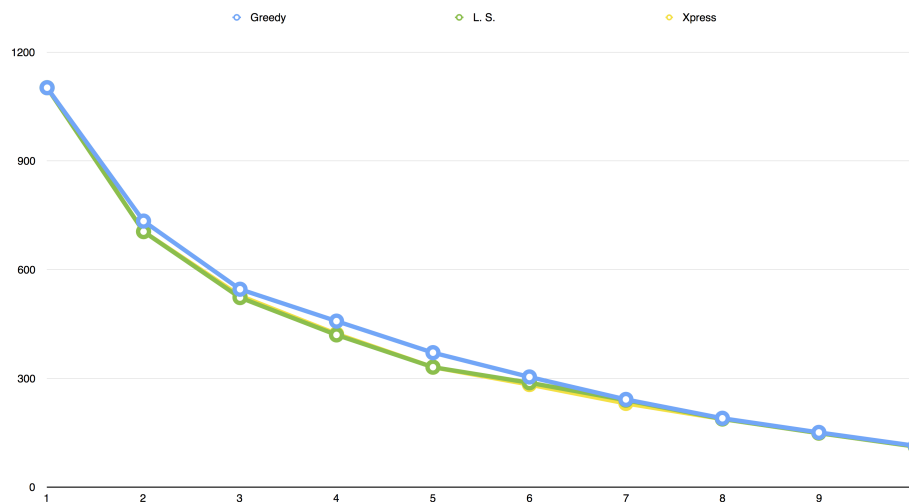


Figura 13: *Resultados del problema P-Median aplicado a los datos de 15 poblaciones*

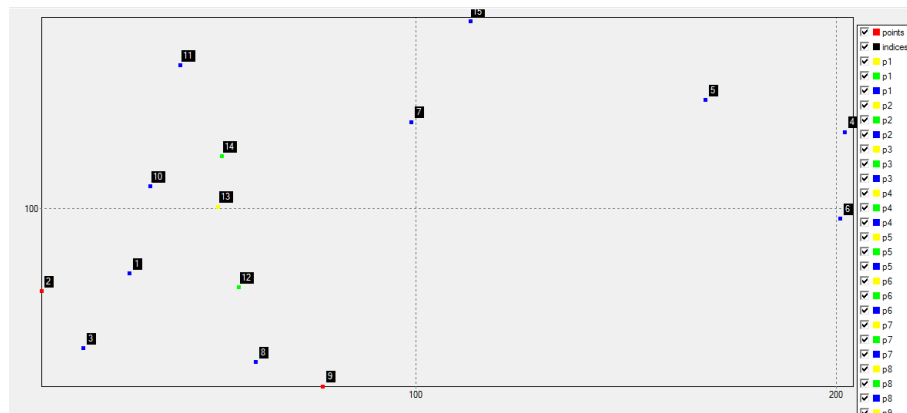


Figura 14: Representación gráfica del problema P-Median aplicado a los datos de 15 poblaciones

p	Greedy	Greedy Opened	L. S.	Xpress	Xpress Opened
1	1102	13	1102	1102	13
2	734	5 13	705	705	1 5
3	546	5 8 13	523	528	5 8 14
4	458	1 5 8 13	420	423	1 5 8 14
5	371	1 5 7 8 13	331	331	1 4 8 14 15
6	304	1 5 6 7 8 13	288	283	1 4 8 10 11 15
7	242	1 5 6 7 8 11 13	240	231	1 4 7 8 10 11 15
8	190	1 5 6 7 8 11 13 15	188	188	1 5 6 7 8 10 11 15
9	151	1 3 5 6 7 8 11 13 15	149	149	1 3 5 6 7 8 10 11 15
10	114	1 3 4 5 6 7 8 11 13 15	112	112	1 3 4 5 6 7 8 10 11 15

Tabla 10: Resultados del problema P-Median aplicado a los datos de 15 poblaciones

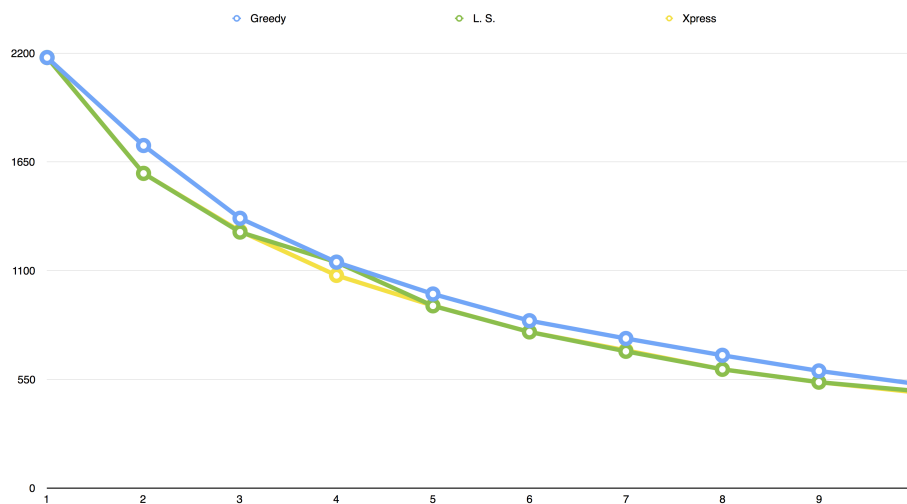


Figura 15: Resultados del problema P-Median aplicado a los datos de 30 poblaciones

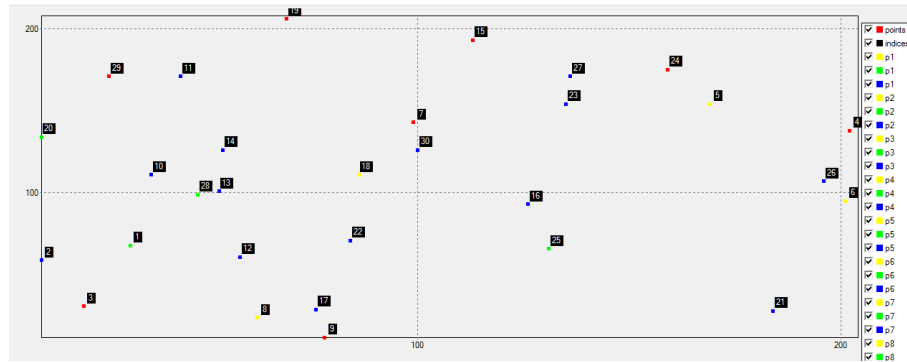


Figura 16: Representación gráfica del problema P-Median aplicado a los datos de 30 poblaciones

p	Greedy	Greedy Opened	L. S.	Xpress	Xpress Opened
1	2178	18	2178	2178	18
2	1733	5 18	1592	1592	23 28
3	1365	5 8 18	1295	1300	5 12 14
4	1143	5 8 11 18	1143	1076	12 14 26 27
5	982	5 6 8 11 18	922	923	8 11 13 23 26
6	847	5 6 8 11 18 28	790	790	8 11 26 27 28 30
7	757	5 6 8 11 18 27 28	692	697	1 10 11 17 26 27 30
8	672	5 6 8 11 18 25 27 28	601	601	1 10 11 17 25 26 27 30
9	593	1 5 6 8 11 18 25 27 28	536	536	1 10 11 16 17 21 26 27 30
10	528	1 5 6 8 11 18 21 25 27 28	492	485	2 10 11 16 17 21 22 26 27 30

Tabla 11: Resultados del problema P-Median aplicado a los datos de 30 poblaciones

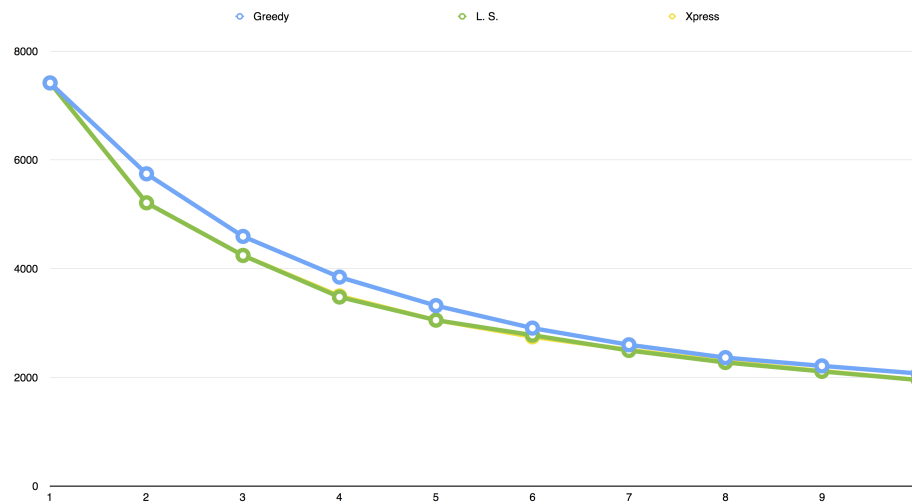


Figura 17: Resultados del problema P-Median aplicado a los datos de 100 poblaciones

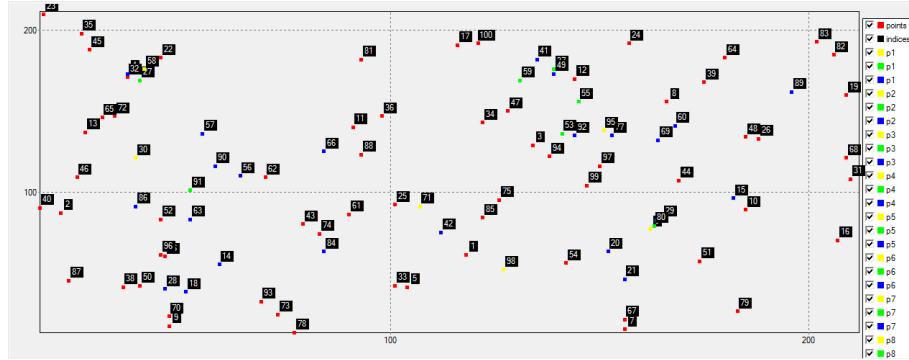


Figura 18: Representación gráfica del problema P-Median aplicado a los datos de 100 poblaciones

p	Greedy	Greedy Opened	L. S.	Xpress	Xpress Opened
1	7416	71	7416	7416	71
2	5744	71 95	5211	5211	63 77
3	4593	30 71 95	4243	4243	57 77 84
4	3845	28 30 71 95	3478	3500	6 14 55 57
5	3320	6 28 30 71 95	3053	3053	4 6 18 55 56
6	2907	6 28 30 58 71 95	2777	2745	4 18 20 49 56 69
7	2600	6 28 30 37 58 71 95	2495	2509	4 6 28 42 49 60 90
8	2364	6 28 30 37 58 71 89 95	2274	2287	4 6 28 41 42 89 90 92
9	2212	6 28 30 37 58 71 89 95 98	2110	2118	4 6 28 41 42 66 86 89 92
10	2072	6 28 30 37 58 66 71 89 95 98	1952	1952	4 15 21 28 41 42 66 86 89 92

Tabla 12: Resultados del problema P-Median aplicado a los datos de 100 poblaciones

REFERENCIAS

- [1] AGUADO, J. S. Modelos de Investigación Operativa, 2016/17.
- [2] GARCÍA PRADO, S. Mosel Examples. <https://github.com/garciparedes/mosel-examples>.