Facultad de Ciencias Sociales y Jurídicas de Elche

Estadística Empresarial

2021/2022

Formulación y resolución de problemas cotidianos con optimización combinatoria

Sara Nian Sanabria Albert

Tutor:
Mercedes Landete Ruiz

Índice

1. Introducción

La optimización combinatoria es una rama de la optimización matemática que trata de hallar la solución óptima a problemas donde el número de opciones posibles es excesivamente grande. Este tipo de problemas tendrán una función objetivo que evalúe la bondad de las diferentes soluciones y un conjunto de acciones posibles para cada situación concreta. Lo que se pretende es minimizar o maximizar la función objetivo para tratar de encontrar la mejor solución a partir de la información de la que se disponga en cada caso.

Los problemas de optimización combinatoria se suelen llamar P o fáciles cuando se pueden resolver con un algoritmo en tiempo polinómico, y NP duros o difíciles cuando son más complejos y se requiere de mucho tiempo para resolverlos. La complejidad de cada situación suele estar relacionada con el tamaño de cada problema ya que el número de soluciones posibles crece exponencialmente a medida que este se va haciendo más grande.

En este tipo de problemas, las variables se llaman variables de decisión y son de tipo enteras. Existen problemas específicos dentro de esta rama, siendo estos algunos de los más utilizados:

- El viajante de comercio (Travelling Salesman Problem)
- El problema de la mochila
- El problema de la asignación

Hay dos formas de resolver estos problemas, implementando Modelos de Optimización o Algoritmos Heurísticos. Dependerá de las dimensiones del problema ya que habrá casos en los que no se podrá aplicar un modelo exacto por el tiempo computacional. En el caso de los exactos, se garantiza hallar una solución óptima; en cambio, al utilizar un algoritmo heurístico se encuentra una solución buena, sin embargo, esto no implica que sea la óptima.

La optimización combinatoria es muy útil y está presente en nuestro día a día. Estos problemas se pueden aplicar en diversidad de campos, aquí vemos diferentes ejemplos:

En logística se suele dar mucho este tipo de problemas. Cuando tenemos mercancía que se tiene que entregar a clientes y queremos que el recorrido que se haga sea el de menor distancia, por ejemplo.

También se utiliza mucho para la secuenciación de tareas. En empresas donde las actividades van en cadena y se busca que el tiempo de ejecución de estas sea el mínimo posible, teniendo en cuenta que a veces se tiene recursos limitados.

Se aplica mucho en industria. Está el problema de corte de materiales, que consiste en cortar tableros grandes en más pequeños, queriendo que se utilicen los menos posibles.

Para establecer horarios. Por ejemplo, en institutos o universidades es complicado ver qué horario asignar a cada grupo sin que se asignen mismas aulas, o que un mismo grupo no tenga clases diferentes a la misma hora. Para este tipo de problemas también se usa la optimización combinatoria.

El objetivo principal de este trabajo es dar a conocer la importancia que tiene la optimización combinatoria y cómo esta está presente en nuestra vida cotidiana. Para ello nos centramos en el estudio y planteamiento de diferentes problemas que resolveremos con la ayuda de programas como Microsoft Excel y Rstudio. Este objetivo principal se logrará a través de los siguientes objetivos parciales:

- Identificar una colección problemas relevantes donde la optimización combinatoria tiene un papel muy importante. Como hemos mencionado antes, en este trabajo se pretende destacar la importancia de la optimización combinatoria en problemas cotidianos, y por ello planteamos problemas como el del análisis de las paradas de bus.

- Construir un modelo matemático para cada uno de los problemas que se plantean. Ver cómo cambia la solución óptima cuando se realizan pequeñas modificaciones en el modelo.

- La resolución de estos problemas programándolos en Rstudio utilizando librerías actuales como lo es la Lpsolve.

2. Análisis de las paradas de autobús en Elche

En la ciudad de Elche actualmente encontramos 13 líneas de bus urbanas que serían las que van desde la A hasta la L, contando con las líneas K1 y K2, y 4 más que conectan las pedanías. La red de buses tiene un total de 169 km y cuenta con 324 paradas. A día de hoy, algo más del 90 % de la población de Elche tiene una parada a menos de 250 metros.

En la Figura 1, sacada de AUESA, se pueden observar los datos de la oferta de la red de bus urbana. Por cada línea encontramos la frecuencia de buses en minutos, las expediciones diarias, la longitud del trayecto, los vehículos de los que se dispone y la velocidad media de estos.

Línea	Intervalo de paso (min)	Expediciones diarias	Longitud	Vehículos	Velocidad comercial
A	13	64	8,0	3	12,7
B	8	105	10,3	6	11,3
C	10	84	4,8	3	8,6
D	9	93	10,6	6	11,7
E	11	76	11,3	4	13,9
F	12	70	9,7	4	11,0
G	13	64	8,7	3	12,4
H	9	93	9,8	5	9,7
I	12	70	9,1	4	10,6
J	14	60	6,3	2	12,1
K1	10	84	13,5	5	14,6
K2	10	84	13,6	5	14,6
L	20	42	8,0	2	11,0
R1	85	9	20,9	1*[3]	20,3
R2	45	18	12,3	1*[3]	16,9
R3	85	9	11,9	1*[3]	8,3
CN	*No se ha incluido en el análisis ya ue no se dispone de datos de demanda*				
Total		1.025	169	54	

Fuente: AUESA

Figura 1: Datos oferta red de buses Elche

Entre febrero y mayo de 2018 se realizó un estudio del plan de movilidad Metropolitano de Alicante-Elche donde se encuestaba a personas de todos los municipios que pertenecían a esta área metropolitana mediante entrevistas telefónicas. Y en febrero de 2020 se realizó otro, pero en este caso se recogía información mediante tarjetones con preguntas que 2 controladores repartían a los pasajeros que subían al bus.

En el primer estudio se vio que, dependiendo del motivo del desplazamiento, las horas donde más se utilizaba el bus, variaba. Si el motivo era el trabajo o los estudios, el pico de los desplazamientos se daba entre las 7 y las 9, que justamente coincide con el inicio de la jornada de trabajo y las clases. Sin embargo, si el motivo era personal (compras, ocio, médico. . .), el pico se daba entre las 9 y las 12.

En la Figura 2 vemos el porcentaje de los desplazamientos por horas cuando no se tiene en cuenta el motivo. Observamos que las franjas con porcentajes más altos siguen siendo las de la mañana entre las 7 y las 9, seguidas de las del mediodía entre las 14 y 15 horas.

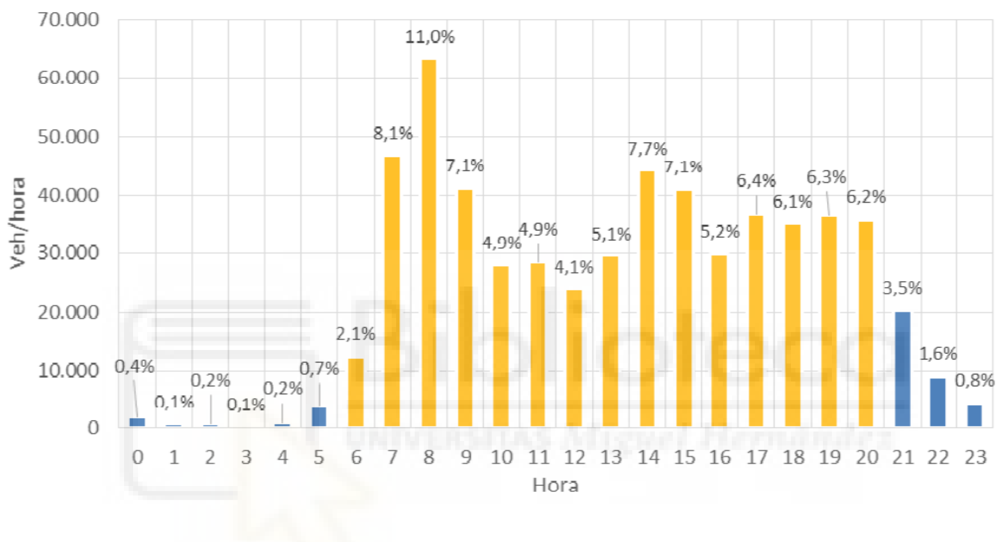

Figura 2: Porcentajes de desplazamientos por horas

Solo centrándonos en las líneas urbanas, las dos más utilizadas son la B con unos 6.653 pasajeros diarios y la D con 5.838. De media, el tiempo de los desplazamientos se calcula en unos 20,7 minutos.

2.1. Ruta de menor distancia

En este primer apartado se plantea un problema donde se analizan las paradas de autobús en Elche, donde el objetivo es minimizar la suma de las distancias que los usuarios recorren para acceder a las paradas. Para ello hemos cogido la línea C, una línea con un recorrido corto, pero con alta demanda.

A continuación, se ve en el mapa el recorrido de la línea con las 41 posibles paradas en azul y en amarillo las paradas que actualmente están en uso en la ciudad de Elche.

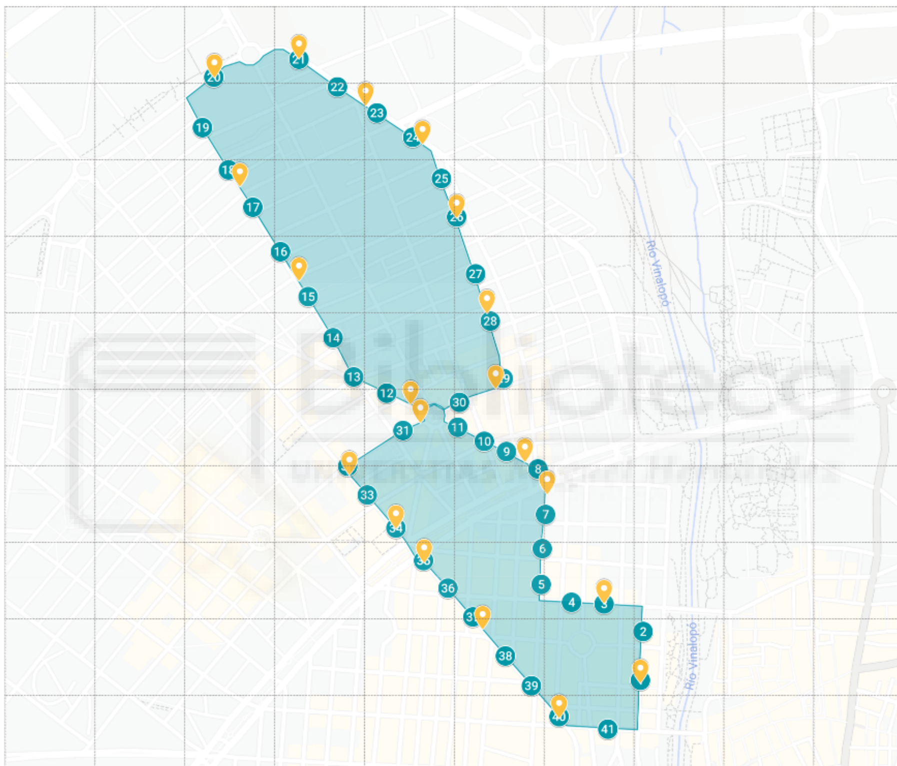

Figura 3: Mapa Urbano Línea C

Se considera relevante solamente la distancia desde el domicilio hasta la parada de salida.

En este caso se trabaja con dos familias de variables, las x's y las y's, ambas variables binarias donde:

$$x_{ij} = \begin{cases} 1 & \text{si el cliente } j \text{ va a la parada } i \\ 0 & \text{en caso contrario} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{si la parada } i \text{ se abre} \\ 0 & \text{en caso contrario} \end{cases}$$

Por otro lado, tenemos el parámetro c_{ij} que hace referencia al coste de la demanda. El parámetro d_{ij} que nos indica la distancia a la que está la persona de cada una de las paradas. Y por último p, número de paradas que se desean abrir.

En concreto, se plantea un problema con 41 paradas de bus posibles y 100 individuos. No nos referimos a personas cuando hablamos individuos sino a las casillas que encontramos en el mapa al poner una cuadrícula. Cuando hacemos referencia a un individuo es realmente la suma de todas las personas que hay en una cuadrícula. El índice i indica las diferentes paradas y el índice j, los clientes.

Empezamos planteando un modelo donde la demanda es igual para todas las paradas (c_{ij}). Para el cálculo de la distancia de los individuos a cada una de las paradas (d_{ij}), hemos calculado las posiciones de los individuos, cada centro de cuadrado sería una persona diferente, teniendo un total de 100 clientes. Estos datos se muestran en la siguiente matriz rectangular (Figura 4) donde cada fila es un individuo y cada columna una parada. Estas no son distancias reales por kilómetros o metros, para hacerlo más sencillo hemos considerado una escala diferente. Para esta escala nos hemos basado en la cuadrícula dibujada en el mapa donde cada cuadrado tiene una distancia de 1 u.d.

	1	2	3	4	5	6	7	8	9	10	11	12
1	6,6370174	6,7957707	6,3192167	6,0440053	5,7412978	6,0018164	6,306346	6,4777928	6,2201286	6,192334	6,0582588	5,8026632
2	5,6435804	5,6435804	5,3602705	5,0921508	4,8127435	5,1070344	5,4378304	5,6587808	5,4488531	5,4447222	5,3481305	5,2221547
3	4,652956	4,652956	4,4195588	4,1629317	3,9195025	4,2593192	4,6227697	4,9073211	4,7634021	4,7900939	4,74368	4,782353
4	3,6674242	3,6674242	3,5117659	3,2756679	3,0923292	3,4931075	3,8948684	4,2593192	4,2059482	4,2714166	4,2898135	4,5244779
5	2,6925824	2,6925824	2,6706741	2,4758837	2,4005208	2,8743347	3,312099	3,7685276	3,8327536	3,9427148	4,0376354	4,4800558
6	1,7464249	1,7464249	1,9830532	1,8788294	1,9906029	2,514319	2,9614186	3,5016853	3,7	3,852921	4,0252329	4,6552014
7	0,9219544	0,9219544	1,6530275	1,7117243	2,0402206	2,5262225	2,9274562	3,5102422	3,8327536	4,0180841	4,2547033	5,027017
8	0,8062258	0,8062258	1,8794946	2,0808652	2,5223997	2,9054776	3,2202484	3,7923344	4,2059482	4,4096485	4,6906823	5,5561587
9	1,5652476	1,5652476	2,5164459	2,7802878	3,25	3,5357885	3,764306	4,2943917	4,7634021	4,9744346	5,2822817	6,2024914
10	2,5	2,5	3,3365401	3,6235342	4,0942032	4,3083407	4,4687806	4,9499293	5,4488531	5,6608303	5,983519	6,9333181
11	6,6068147	6,6068147	6,1345334	5,8420887	5,482928	5,6640798	5,9135438	6,0018164	5,6824291	5,6253889	5,45	5,0725635
12	5,60803	5,60803	5,1412547	4,8507731	4,5013887	4,7055074	4,9769469	5,1070344	4,8259714	4,7900939	4,647849	4,3966919
13	4,6097722	4,6097722	4,1512046	3,8639358	3,5302266	3,7685276	4,0706265	4,2593192	4,0360872	4,0305087	3,9373214	3,8640523
14	3,6124784	3,6124784	3,1674122	2,8861739	2,5811819	2,8743347	3,2202484	3,4931075	3,3600595	3,3977934	3,3767588	3,5399011
15	2,6172505	2,6172505	2,1982948	1,9313208	1,6918924	2,0788939	2,4839485	2,8743347	2,879236	2,9740545	3,05	3,4829442
16	1,6278821	1,6278821	1,2776932	1,0630146	1,0307764	1,5433081	1,9924859	2,514319	2,7	2,8539446	3,0335623	3,7055229
17	0,6708204	0,6708204	0,6576473	0,728011	1,1236103	1,562626	1,9416488	2,5262225	2,879236	3,0732719	3,3320414	4,1630398
18	0,5	0,5	1,1101802	1,3892444	1,8607794	2,1217446	2,3600847	2,9054776	3,3600595	3,570014	3,8733061	4,7886219
19	1,4317821	1,4317821	2,0081086	2,3086793	2,7681221	2,9260554	3,0610456	3,5357885	4,0360872	4,2479407	4,5719252	5,5254774
20	2,4186773	2,4186773	2,9719522	3,2756679	3,7232378	3,8238462	3,8948684	4,3083407	4,8259714	5,0343818	5,3667961	6,3348954
21	6,726812	6,726812	6,1100327	5,8077534	5,4002315	5,4901548	5,6718604	5,6640798	5,2810984	5,1908573	4,970161	4,4486964

Figura 4: Matriz de Distancias

En la matriz vemos solo las distancias del individuo 1 - 21 hasta la parada 1 - 12. Esta matriz la convertimos más adelante en un vector columna para poder resolver el problema en R.

El modelo matemático que planteamos es el siguiente:

$$\text{mín} \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} x_{ij} \tag{1}$$

$$\text{s.t.} \sum_{i} y_i = p \tag{2}$$

$$x_{ij} \leq y_i; \forall i, \ \forall j \tag{3}$$

$$\sum_{i} x_{ij} = 1, \forall j \tag{4}$$

$$x_{ij} \in \{0,1\} \qquad j = 1, \ldots, n. \tag{5}$$

$$y_i \in \{0,1\} \qquad i = 1, \ldots, n. \tag{6}$$

Ahora pasamos a resolverlo en Rstudio con el uso de la librería Lpsolve (Figuras 5 y 6).

```
y = 41
x = 100

col = x*y + y
fil = 1 + x*y + x

A = matrix(0, nrow=fil, ncol=col)

A[1,4101:4141]=1

for (cont in 0:40){
  A[(2+x*cont):(101+x*cont), (1+x*cont):(x+x*cont)] = diag(1, x)
  A[(2+x*cont):(101+x*cont), 4101+cont] = rep(-1,x)
  A[4102:4201, (1+x*cont):(x+x*cont)] = diag(1,x)
}

b = c(20, rep(0, (x*y)), rep(1, x))

signos = c("==", rep("<=", (x*y)), rep("==", x))
```

Figura 5: Problema Minimizar Distancias

El programa nos devuelve un vector con 1's y 0's como el que se muestra en la Figura 7 con la solución del problema. Las últimas 41 posiciones de este vector nos indican las paradas que se deben abrir para minimizar la función objetivo. En este caso se abren un total de 20 paradas, tal y como nosotros habíamos fijado en el problema ($p = 20$). Las paradas que se abren son la 1, 2, 7, 8, 14, 15, 17, 19, 20, 21, 23, 25, 27, 29, 31, 32, 34, 36, 38 y 40.

```
dist <- read_excel("Col_Mat.xlsx", col_types = c("numeric"))
dim(dist)

dij <- c(rep(0,4100))
for (i in 1: 4100){
  dij[i] <- dist[i,1]
}

dij <- c(dij, rep(0,41))

View(dij)

of=lp('min', dij, A, signos, b, all.bin = TRUE)
of

solucion = of$solution
solucion
```

Figura 6: Problema Minimizar Distancias

La solución óptima nos da 176.03, por lo que la suma de las distancias que recorre cada uno de los individuos a su correspondiente parada es de 176.03 u.d. Ahora vemos a qué parada va cada pasajero en la Figura 9.

Para entender el mapa, necesitamos la información que se encuentra en la Figura 10. Cada parada está asociada a un color; en la tabla vemos la parada 1 que corresponde al color morado. En el mapa observamos que ese color está en los cuadrados de la esquina inferior derecha, es decir, que esos individuos irán a la parada 1. Otro ejemplo serían los cuadrados de la esquina superior derecha, sombreados de color naranja, que corresponden a la parada 25. Y así sería con el resto de paradas.

2.2. Ruta de menor distancia con diferente demanda

Ahora, pasamos a comprobar la relación que hay entre las paradas que se abren y la demanda de pasajeros de cada zona. Para ello, utilizamos el mismo modelo y cambiamos simplemente la función objetivo del problema. Añadimos la matriz de costes (c_{ij}) que aparece en la Figura 11 que multiplica a d_{ij}.

Simulamos un escenario donde las paradas a las que más personas acuden son la 1, 2, 3, 4, 18, 19 y 20, y por ello, en la matriz c_{ij} vemos que las distancias en estas paradas se multiplicarán por 60.

Volvemos a resolver el problema con este cambio en la función objetivo y nos sale que el resultado de la suma de las distancias que recorren todos los individuos a su parada es de 184.79 u.d. Al igual

```
> solucion
  [1] 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [33] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [65] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [97] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1
[129] 1 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
[161] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[193] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[225] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[257] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[289] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[321] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[353] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[385] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[417] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[449] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[481] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[513] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[545] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[577] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[609] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0
[641] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[673] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[705] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[737] 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0
[769] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[801] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[833] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[865] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Figura 7: Solución Problema Minimizar Distancias: Vector 1's y 0's

```
> solucion[4101:4141]
 [1] 1 1 0 0 0 0 1 1 0 0 0 0 0 1 1 0 1 0 1 1 1 0 1 0 1 0 1 0 1 0 1 1 0
[34] 1 0 1 0 1 0 1 0
```

Figura 8: Solución Problema Minimizar Distancias: Vector 1's y 0's con las últimas 41 posiciones

que antes, los colores que se muestran en la Figura 13 indican la parada a la que debe ir cada usuario. Si comparamos con el anterior ejemplo observamos que las paradas que se abren también son diferentes como aparece en la Figura 12.

A simple vista puede parecer que la solución no tenga sentido ya que hemos aumentado la demanda a paradas que finalmente no se abren, sin embargo, esto tiene su explicación. Se podría entender como que la distancia a estas paradas aumenta y el modelo, al minimizar, decide abrir otras diferentes con el objetivo dejar aquellas que queden a la menor distancia de la mayoría de usuarios, aún así, vemos que la suma de las distancias en este modelo es mayor.

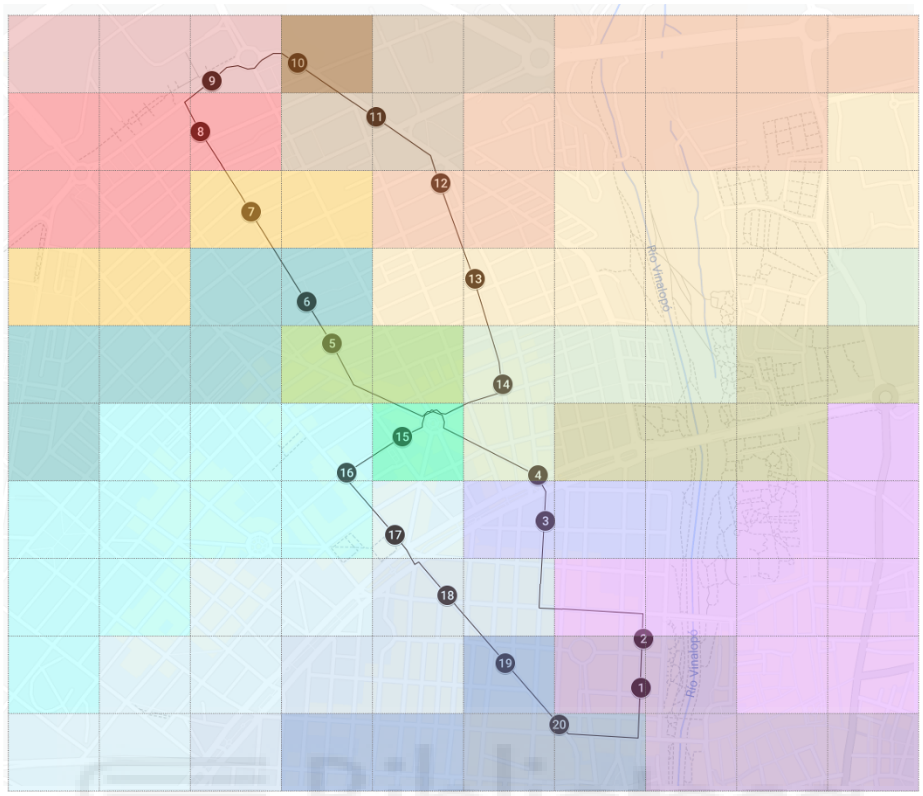

Figura 9: Mapa Paradas

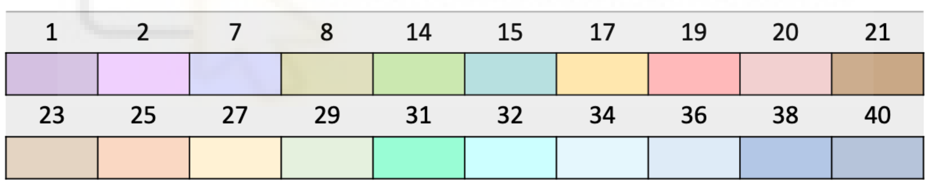

Figura 10: Colores paradas

2.3. Ruta de menor distancia con paradas reales

Por último, vamos a calcular la distancia que recorren los usuarios con las paradas que actualmente están abiertas en Elche. Al resolver este escenario, la solución óptima nos da 180.48 u.d. En la Figura 14 se ve cómo quedaría el mapa con las paradas reales de la línea C.

Comparamos este resultado con el del primer ejemplo resuelto y como observamos, a los individuos se les asigna paradas diferentes.

Teniendo estos tres escenarios, podríamos concluir que la mejor forma de distribuir las paradas de bus para minimizar la suma de las distancias sería la primera forma.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
2	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
3	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
4	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
5	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
6	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
7	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
8	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
9	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
10	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
11	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
12	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
13	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
14	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
15	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
16	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
17	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
18	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
19	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
20	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
21	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
22	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
23	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
24	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
25	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
26	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
27	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
28	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
29	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60
30	60	60	60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	60	60	60

Figura 11: Matriz cij

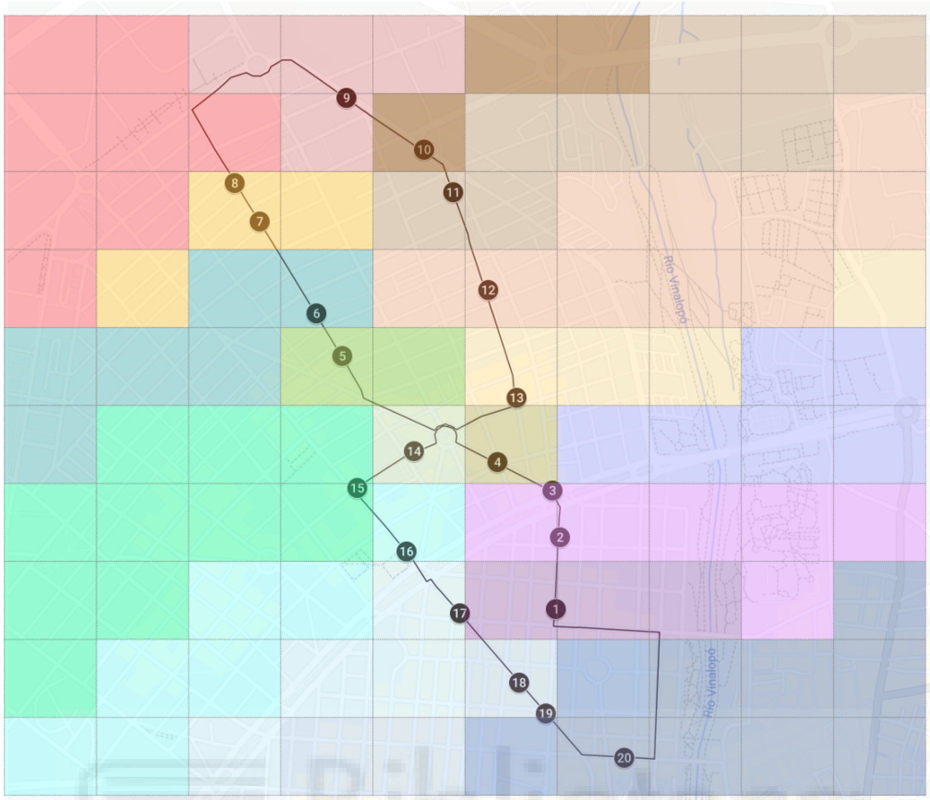

Figura 12: Mapa paradas con diferentes demandas

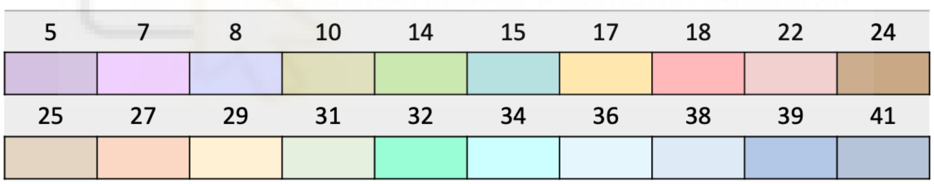

Figura 13: Colores paradas

2.4. Problema de maximizar número de parejas (clientes-paradas)

Continuamos con este otro apartado donde se plantea un modelo en el que se vuelven a analizar las paradas de bus en la ciudad de Elche. En este caso se desea maximizar el número de parejas (cliente, parada) útiles, es decir, solamente contarán las parejas cuyos clientes se encuentren a una distancia máxima de 800 m. de las paradas. Para ello, vamos a utilizar el recorrido que realiza la línea C.

Al igual que en el modelo anterior, se tienen 41 posibles paradas y 100 clientes. Se tiene en cuenta que el índice i indica las diferentes paradas y el índice j, los clientes. Y tenemos también el parámetro p que hace referencia al número de paradas que se desean abrir.

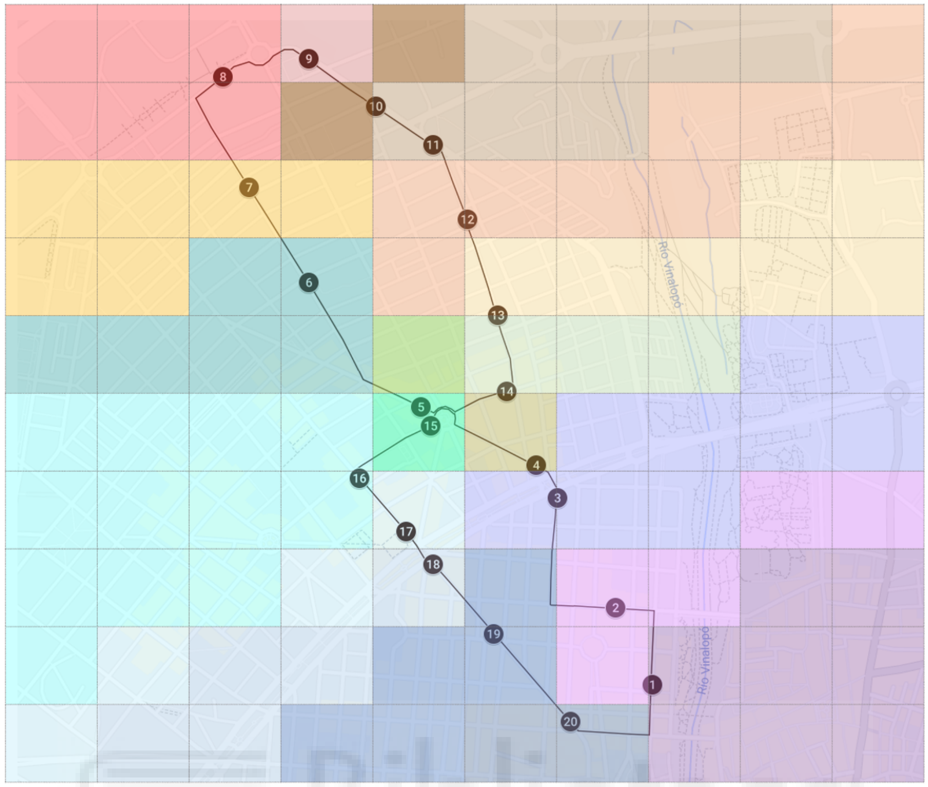

Figura 14: Mapa paradas reales línea C

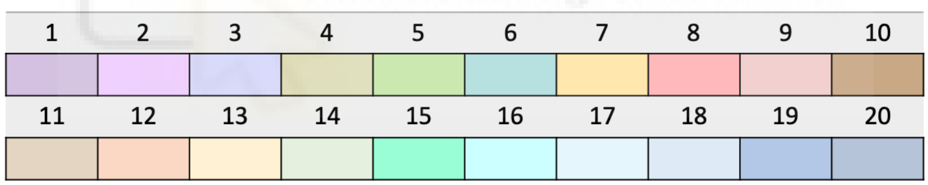

Figura 15: Colores paradas reales

También se trabajan dos familias de variables binarias, x e y:

$$x_{ij} = \begin{cases} 1 & \text{si el cliente } j \text{ va a la parada } i \\ 0 & \text{en caso contrario} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{si la parada } i \text{ se abre} \\ 0 & \text{en caso contrario} \end{cases}$$

En primer lugar planteamos el modelo matemático para resolver el problema.

$$\text{máx} \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} \tag{7}$$

$$\text{s.t.} \sum_{i} y_i = p \tag{8}$$

$$x_{ij} \leq y_j \qquad \forall i, j \tag{9}$$

$$x_{ij} = 0 \qquad \forall i, j : d_{ij} > 800 \tag{10}$$

$$x_{ij} \in \{0, 1\} \qquad j = 1, \ldots, n. \tag{11}$$

$$y_i \in \{0, 1\} \qquad i = 1, \ldots, n. \tag{12}$$

Con la librería Lpsolve del programa Rstudio, lo resolvemos utilizando la misma matriz de distancias calculada en el anterior apartado.

```
x = 100
y = 41

col = x*y + y
fil = 1 + x*y + x*y

#funcion objetivo
f <- c(rep(1,x*y), rep(0,y))
f

#matriz A
A=matrix(0, nrow=fil, ncol=col)

View(A)

#primera
A[1,4101:4141]=1

#segunda
for (cont in 0:40){
  A[(2+x*cont):(101+x*cont), (1+x*cont):(x+x*cont)] = diag(1,x)
  A[(2+x*cont):(101+x*cont), 4101+cont] = rep(-1,x)
}
```

Figura 16: Problema Maximizar Número Clientes

Como resultado, el valor óptimo es de 993. Este es el número de parejas donde el cliente está a una distancia menor de 800 metros de una de las paradas abiertas. Las paradas que se abren son las que van de la 6 a la 15 y de la 26 a la 35. En la Figura 19 se muestra el mapa con estas paradas y sombreado de verde los pasajeros que estarían cubiertos.

Si reducimos esta distancia a 600 m., es decir, ahora suponemos que los clientes no están dispuestos a ir a una parada que tengan a más de 600 m, entonces las que se abrirían serían las siguientes:

```
#tercera
R = 3
dist <- read_excel("Col_Mat_2.xlsx", col_types = c("numeric"))

dij <- c(rep(0,4100))
for (i in 1: 4100){
  dij[i] <- dist[i,1]
}

dij <- as.numeric(dij)

for(i in 1:length(dij)){
  if (dij[i] < R){
    dij[i] = 0
  }
}

for (i in 1:length(dij)){
  if (dij[i] != 0){
    dij[i] = 1
  }
}

for (i in 1:length(dij)){
  A[4102+i-1, i] = dij[i]
}
```

Figura 17: Problema Maximizar Número Clientes

```
#signos
signos=c("==", rep("<=", x*y), rep("==", x*y))

b=c(20, rep(0, x*y), rep(0, x*y))

bin = seq(1,col)

solucion=lp('max', f, A, signos, b, binary.vec=bin)
solucion

eo=solucion$solution
eo
```

Figura 18: Problema Maximizar Número Clientes

Al reducir la distancia habrá menos personas que queden cubiertas y por ello, vemos menos cuadrados sombreados en la Figura 20. El valor óptimo pasa a ser 576, por lo que reduciendo la distancia en 200 m., pasan a haber 417 parejas menos.

Continuando con el problema de maximizar el número de clientes, vamos a comprobar a cuántas parejas se les podría dar servicio si utilizamos las paradas reales de la línea C. Al igual que antes, vamos a plantear dos escenarios; el primero con la condición de que la distancia que cada persona deberá recorrer a la parada sea de máximo 800 m, y el segundo con un máximo de 600. Estos resultados se muestran en las Figuras 21 y 22.

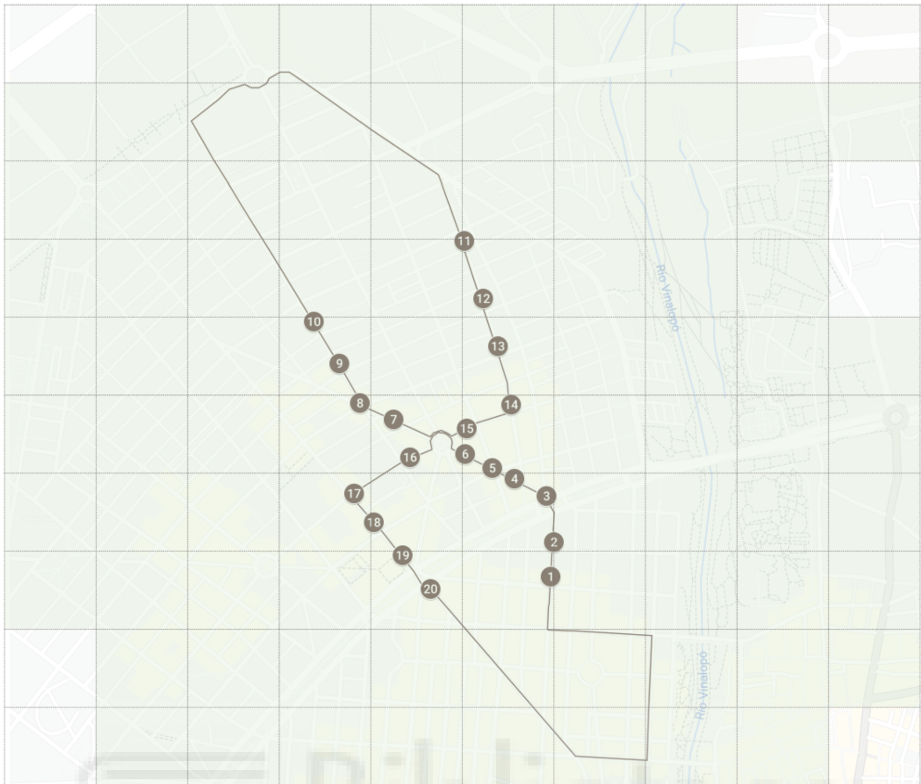

Figura 19: Mapa Paradas al maximizar número de clientes con distancia menor a 800 m.

3. Recorridos ecológicos

En este apartado dejamos a un lado los clientes y nos centramos en los autobuses. Vamos a plantear un modelo donde buscaremos el recorrido más económico para el combustible.

Para este modelo nos hemos basado en el tan conocido problema del viajante de comercio (Travelling Salesmam Problem, TSP). El TSP consiste en encontrar el recorrido de menor distancia para que el agente visite todas las ciudades. Este problema que a simple vista puede parecer sencillo se considera un problema NP-duro ya que, a medida que va creciendo el número de ciudades, el número de combinaciones va aumentando de forma factorial.

Para empezar a plantear el problema de la ruta más económica para el combustible, tenemos que ver qué variables son las que más afectan al consumo. Uno de los factores que más contribuye es la distancia que se recorre, por otro lado, el peso del bus y la velocidad a la que vaya. Otras variables de menor peso serían las cuestas del camino, o la aceleración. Se puede calcular una aproximación del consumo por $A * d_{ij} * w$ cuando la velocidad es constante, donde A es una constante, d_{ij} la distancia entre paradas y w, otra constante que hace referencia al peso del vehículo.

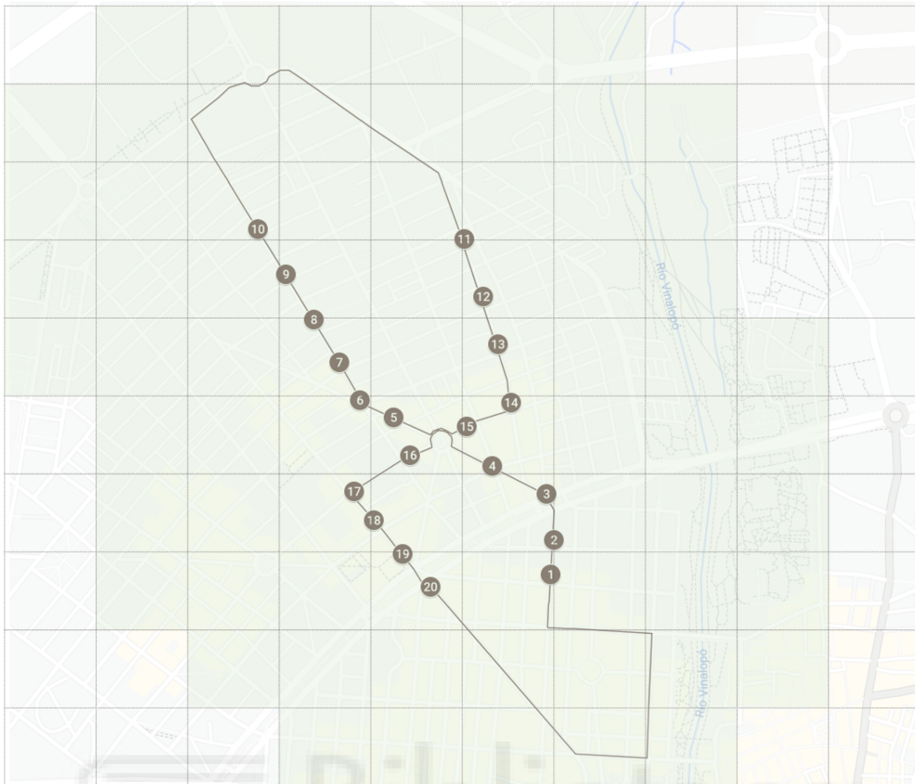

Figura 20: Mapa Paradas al maximizar número de clientes con distancia menor a 600 m.

Para la función objetivo tenemos varias opciones dependiendo de los datos de los que dispongamos. Las penalizaciones en lugar de ser únicamente d_{ij}, pasan a ser $A * d_{ij} * w$. Las distancias entre las paradas no cambian, sin embargo, el peso del vehículo varía en función de si el bus lleva a más o menos pasajeros. Por un lado, podríamos considerar un ejemplo con el peso del bus cuando este va vacío. De promedio, los buses suelen pesar entre 11.339,8 y 18.143,6 kg, si ponemos un peso intermedio, la función objetivo se calcularía de la siguiente forma:

$$\text{mín}\, A * d_{ij} * 15{,}000 \tag{13}$$

Por otro lado, podríamos calcularla dependiendo de la demanda en el nodo i (d_i) y la demanda en el nodo j (d_j) además de la distancia entre las paradas:

$$\text{mín}\ \ d_{ij} + d_i + d_j \tag{14}$$

El modelo matemático para resolver el problema de minimizar el consumo de combustible sería el siguiente:

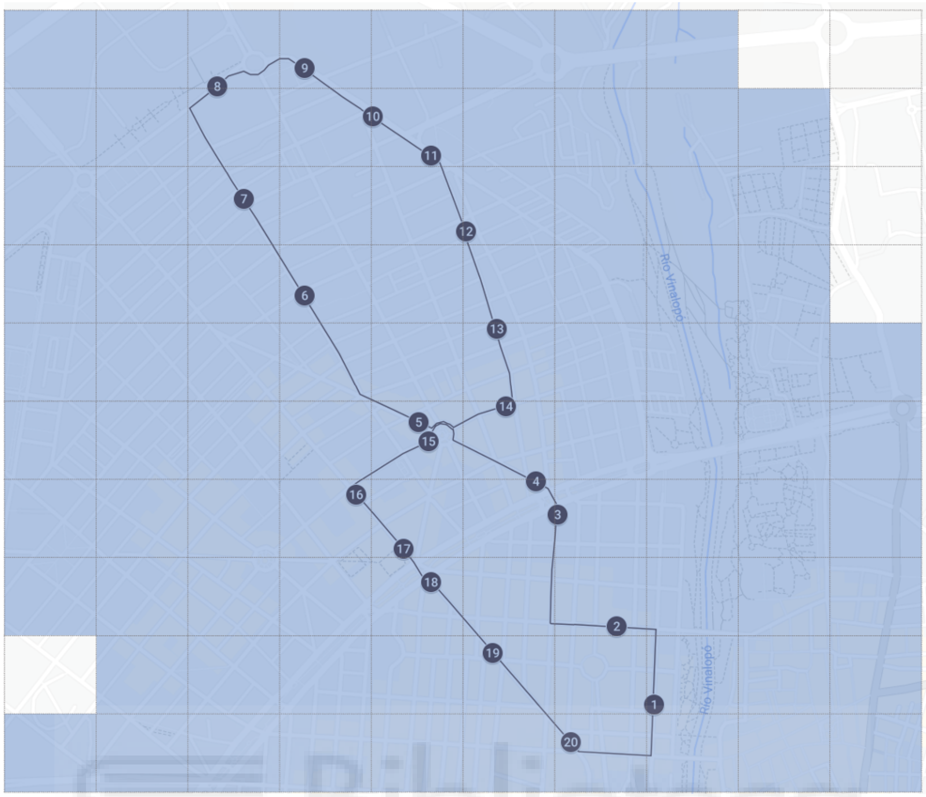

Figura 21: Mapa Paradas Reales al maximizar número de clientes con distancia menor a 800 m.

$$\text{mín} \sum_{i=0}^{n} \sum_{j \neq i; j=1}^{n} c_{ij} x_{ij} \tag{15}$$

$$\text{s.t.} \sum_{i=0, i \neq j}^{n} x_{ij} = 1 \; j = 1, \ldots, n \tag{16}$$

$$\sum_{j=0, j \neq i}^{n} x_{ij} = 1 \; i = 1, \ldots, n \tag{17}$$

$$x_{ii} = 0; \forall i \tag{18}$$

$$x_{ij} \in \{0, 1\} \qquad j = 1, \ldots, n \tag{19}$$

$$u_i + u_j + n x_{ij} \leq n - 1 \; i \leq i \neq j \leq n \tag{20}$$

En este caso, $c_{ij} = d_{ij} + d_i + d_j$ o $c_{ij} = A * d_{ij} * 15{,}000$. Para este modelo, únicamente tenemos una variable binaria, x_{ij}:

$$x_{ij} \ (i \neq j) = \begin{cases} 1 & \text{si el viajante realiza el trayecto desde } i \text{ hasta } j \\ 0 & \text{en caso contrario} \end{cases}$$

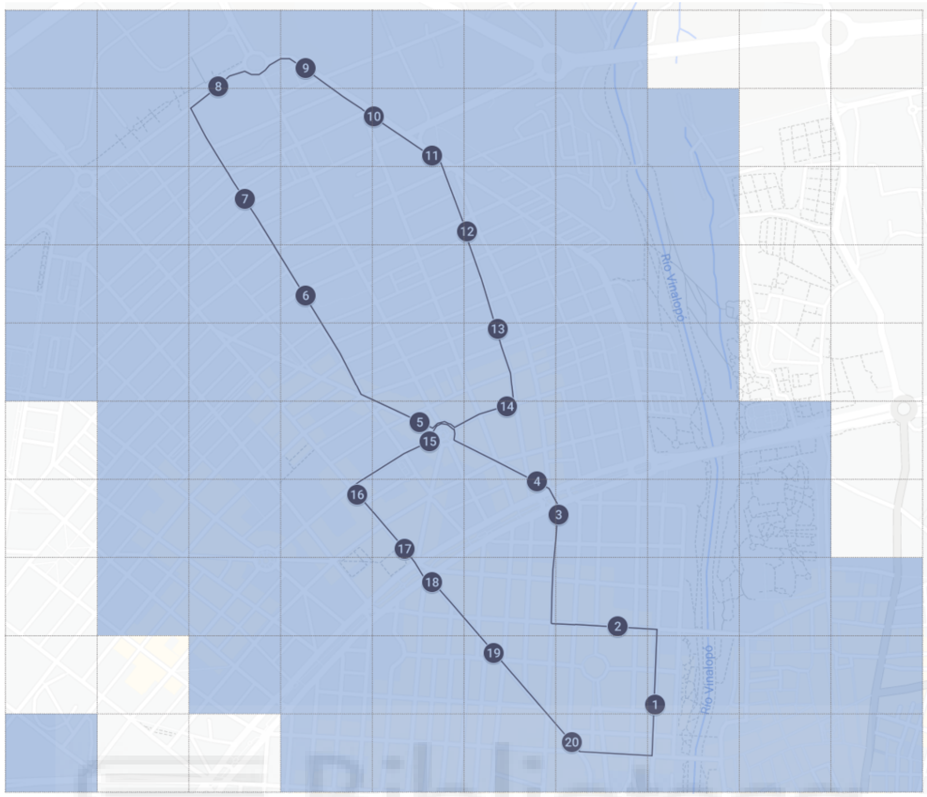

Figura 22: Mapa Paradas Reales al maximizar número de clientes con distancia menor a 600 m.

Cuando el parámetro $u_i = t$ entonces la parada i se visita en el momento t ($t = 1, 2, ..., n$).

Suponemos que el recorrido empieza y termina siempre en la parada 0, entonces: La restricción 16 fuerza a que cada parada $0, ..., n$ de salida llegue a exactamente otra parada. La siguiente restricción asegura que desde cada parada $1, ..., n$ se salga exactamente hacia otra. La 18, nos indica que una parada de salida no puede llegar a esa misma. Y la última de las restricciones obliga a que sea un único camino el que cubra a todas las paradas, y que la solución no nos dé varias rutas diferentes que de forma separada cubran todas las paradas.

4. Problema de enrutamiento de vehículos

En el caso de que tuviéramos que planificar la ruta de varios autobuses, el modelo sería similar al del apartado anterior, pero algo más complicado. Para resolver este problema nos basaríamos en el Problema de enrutamiento de vehículos o en inglés, Vehicle Routing Problem (VRP). Se trata de una extensión del TSP antes mencionado que engloba un conjunto de problemas donde se busca determinar un conjunto de recorridos para una flota de vehículos con base en uno o varios depósitos para un número ciudades o clientes.

En este problema, es habitual que el período de planificación sea de un día, y el tamaño de los vehículos puede ser o no el mismo. Llevándolo al problema de los buses, sabemos que existen autobuses de diferentes tamaños, unos que parecen furgonetas con menor capacidad, otros con dos plantas que casi tienen el doble de capacidad que uno normal...

En nuestro caso, podríamos considerar que la demanda ha aumentado y necesitamos más de una línea para poder dar servicio a todos los clientes. Sabiendo que disponemos de un número fijo de buses para atender a todas las personas, buscaremos una solución óptima que nos diga la ruta que deberá realizar cada bus de los que disponemos buscando que el coste sea el mínimo.

El modelo matemático que plantearíamos de forma general sería:

$$\text{mín} \sum_{i \in V} \sum_{j \in V} d_{ij} x_{ij} \tag{21}$$

$$\text{s.t.} \sum_{i \in V} x_{ij} = 1 \ \forall j \in V \setminus \{0\} \tag{22}$$

$$\sum_{j \in V} x_{ij} = 1 \ \forall i \in V \setminus \{0\} \tag{23}$$

$$\sum_{i \in V} x_{i0} = K; \tag{24}$$

$$\sum_{j \in V} x_{0j} = K; \tag{25}$$

$$\sum_{i \notin S} \sum_{j \in S} x_{ij} \geq r(S) \ \forall S \subseteq V \setminus \{0\}, \ S \neq \emptyset \tag{26}$$

Donde x_{ij} es una variable binaria:

$$x_{ij} = \begin{cases} 1 & \text{si el cliente } j \text{ se dirige hasta la parada } i \\ 0 & \text{en caso contrario} \end{cases}$$

La matriz distancias es d_{ij} y K es una constante que hace referencia al número total de vehículos que se utilizarán para las viajes.

El conjunto V indica las ciudades, o en este caso las paradas, y S son las capacidades de los vehículos.

Las primeras dos restricciones indican que un arco entra y uno deja cada vértice asociado con un cliente, respectivamente. Las restricciones 24 y 25 nos dicen que la misma cantidad de vehículos que salen y entran al depósito debe ser la misma. La última restricción es la de corte de capacidad, que nos indica que las rutas deben estar conectadas y que la demanda de cada una de ellas no debe superar la capacidad del vehículo.

5. Otros recorridos

Partiendo del Problema de Enrutamiento de Vehículos del apartado anterior, podemos encontrar diferentes versiones dependiendo de las restricciones que queramos añadir a ese modelo:

- Si los vehículos tienen una capacidad fija estaremos hablando del VRP capacitado.
- Si las entregas se deben realizar en unos días concretos, VRP periódico.
- Si los clientes pueden ser atendidos por varios vehículos diferentes, VRP con suministro dividido.
- Cuando algunas de las variables son aleatorias (el número de clientes, las demandas de estos, etc.) hablamos del VRP estocástico.
- Cuando a cada persona se le asigna una parada por ejemplo, y en ese momento se encuentra cerrada y necesitan acudir a otra, en inglés VRP with Backhauls (VRPB).

Estas serán las variantes que explicaremos por encima a continuación:

5.1. VRP Capacitado

El CVRP es muy similar al VRP, únicamente añadimos la condición de que todos los vehículos tienen una capacidad uniforme.

Buscaremos una solución que minimice la cantidad de buses utilizados y el tiempo de la ruta. Además, para que la solución sea válida se deberá cumplir la condición de que el número de viajeros no supere la capacidad del bus que da servicio en ese recorrido.

Por tanto, sería el mismo problema que en el apartado anterior, tenemos diferentes líneas para dar servicio a un número de clientes, pero aquí los buses tienen todos la misma capacidad.

5.2. VRP Periódico

La siguiente variante es el Periodic Vehicle Routing Problem o PVRP. Como hemos dicho anteriormente, el VRP suele ser para la planificación de un día, sin embargo, con este modelo ahora el período pasa a ser de M días.

El objetivo de este problema es determinar una flota de vehículos y una suma de tiempos mínimas. En el PVRP la solución factible se encuentra cuando se cumple que el vehículo no vuelve al depósito el mismo día que parte y además, que cada cliente es visitado mínimo una vez durante el período de días.

5.3. VRP con Suministro dividido

El SDVRP o Split Delivery VRP, en inglés, permite que diversos vehículos puedan atender a un mismo cliente, siempre y cuando el coste final se reduzca.

El objetivo, también es la de hallar una solución donde se minimice la flota de vehículos y el tiempo del viaje para satisfacer a todos los clientes, sin embargo, será más difícil obtener una solución óptima que en problema básico de enrutamiento de vehículos.

Un ejemplo donde se utilizaría este tipo de problemas sería cuando en empresas se sirven pedidos de gran tamaño, donde al igual, la forma óptima de transportar esa mercancía es dividirla en diferentes vehículos.

5.4. VRP Estocástico

Seguimos con el siguiente ejemplo, el SVRP. En este tipo de problema encontramos diversos ejemplos dependiendo de la variable aleatoria que tengamos:
- Hablaremos de clientes estocásticos cuando estos tengan una probabilidad de coger o no el bus.
- Nos referiremos a tiempos estocásticos cuando los tiempos de servicio y de ruta sean variables aleatorias.

Al tener datos aleatorios no podemos exigir que se cumplan todas las restricciones. Para poder encontrar una solución óptima a este problema podremos:
- Añadir en el modelo un conjunto de acciones que se llevarán a cabo cuando una de las condiciones no se satisfaga.
- Requerir el cumplimiento de las condiciones del problema con una probabilidad.

En este tipo de problema siempre se buscará la ruta con el mejor valor esperado y es útil cuando por ejemplo, encontramos rutas que tienen paradas cercanas a colegios o institutos. La probabilidad de que un pasajero en edad escolar suba o baje en esas paradas será muy alta. Sin embargo, esta probabilidad variará según el día de la semana, ya que durante los fines de semana o días no lectivos, la probabilidad de utilizar esas paradas disminuirá. Por esta razón, podremos hacer que los recorridos durante esos días no paren en esas paradas.

5.5. VRP con Redes de retorno

Por último, hablamos del VRP with Backhauls o BVRP. Es un problema dentro del VRP donde repartimos productos y dejamos que los clientes puedan devolver otros.

En este problema buscamos un conjunto de recorridos donde la suma de las distancias que se recorra en total sea mínima. Para poder encontrar una solución válida, nos debemos asegurar de que las entregas se hayan realizado antes de coger estos productos que los clientes quieren devolver, y de que además, estos caben en el vehículo.

El BVRP lo podríamos utilizar cuando tenemos una parada principal asignada a cada cliente. Sin embargo, puede que algún día esta esté cerrada por algún motivo (obras, calle cortada, ...) y necesiten acudir a cualquier otra para poder coger el bus. Con este modelo podríamos asignar una parada secundaria a cada cliente y así asegurarnos de que siempre tienen servicio.

6. Conclusiones

Ya para terminar, vamos a hacer un breve resumen de este trabajo. El objetivo principal, como ya hemos mencionado en varias ocasiones, es la de dar a conocer la importancia que tiene la optimización combinatoria en nuestro día a día. Por ello, se han planteado y resuelto algunos ejemplos de problemas en este TFG.

En este caso, hemos centrado casi todo el trabajo en el análisis de paradas de bus en la ciudad de Elche. Todos alguna vez hemos cogido el autobús, ya sea para desplazarnos por nuestra ciudad o para hacer viajes de más distancia, por lo que podía ser muy comprensible y por ello, se eligió como tema de interés. Sin embargo, la optimización combinatoria se puede aplicar a muchísimos otros campos como la banca, la industria, la logística...

Como hemos podido comprobar en este trabajo, las soluciones de los problemas pueden cambiar totalmente si realizamos modificaciones en las condiciones, por pequeñas que sean estas. Es decir, se pueden crear infinidad de modelos para dar respuesta a diferentes cuestiones que se puedan plantear. Podemos jugar con la función objetivo dependiendo de lo que busquemos, o con los parámetros (tiempo, probabilidades, ...).

Al principio del trabajo hemos explicado que el número de posibles soluciones va creciendo según lo va haciendo el problema y por ello, muchas de las veces es muy complicado hallar una solución óptima. De hecho, cuando se implementan algoritmos heurísticos se dice que se encuentra una solución buena, factible, pero no óptima.

El hecho de poder resolver un sinfín de problemas de diferentes áreas a pesar de su complejidad es lo que hace tan atractiva la optimización combinatoria.

Por último, quiero mencionar algunas de las asignaturas que hemos estudiado durante estos cuatro años de carrera y que han tenido especial relevancia en este TFG.

Empezamos aprendiendo optimización desde el primer curso con la asignatura de Matemáticas y durante los siguientes años pudimos seguir ampliando conocimientos sobre este tema con Optimización de Recursos y Modelos de Optimización. En cuarto, con Gestión y Planificación de la Producción, nos enseñaron un poco más en profundidad a plantear modelos y a poder resolverlos. En este último curso también estudiamos sobre la optimización combinatoria y su aplicación en el mundo de las inversiones con la asignatura Gestión de Carteras e Inversiones. Aquí es donde utilicé por primera vez la librería Lpsolve con la que hemos resuelto todos los problemas de este TFG.

Además de lo mencionado, he adquirido nuevos conocimientos. No había tenido la oportunidad de hacer un trabajo donde viera tan de cerca su aplicación real. Al empezar me di cuenta de que los casos en clase eran mucho más pequeños y se podían resolver de manera más rápida. En cambio, aquí tenía problemas con muchas más variables y veía cómo de costoso era plantearlos y resolverlos. Además he descubierto nuevos problemas que no habíamos visto antes como el VRP y sus variantes, y he podido seguir aprendiendo más sobre lo interesante que es la optimización combinatoria.

7. Bibliografía

Capacitated VRP — Vehicle Routing Problem. (2013). Retrieved June 5, 2022, from Lcc.uma.es website: https://neo.lcc.uma.es/vrp/vrp-flavors/capacitated-vrp/

Movilidad sostenible – Ayuntamiento de Elche. (2022). Retrieved June 12, 2022, from Elche.es website: https://www.elche.es/movilidad/movilidad-sostenible/

Parra, O., Antonio, M., Chavez, C., Técnico, R. (2006). El Problema del Transporte. Retrieved from http://www.gridmorelos.uaem.mx/ mcruz/surveykoko.pdf

Pedro, J. (n.d.). Modelado mediante Optimización Combinatoria Página 1 de 31. Retrieved from http://personales.upv.es/jpgarcia/LinkedDocuments/MCOIOptimizacionCombinatoria.pdf

Periodic VRP — Vehicle Routing Problem. (2013). Retrieved June 5, 2022, from Lcc.uma.es website: https://neo.lcc.uma.es/vrp/vrp-flavors/periodic-vrp/

Split Delivery VRP — Vehicle Routing Problem. (2013). Retrieved June 5, 2022, from Lcc.uma.es website: https://neo.lcc.uma.es/vrp/vrp-flavors/split-delivery-vrp/

Stochastic VRP — Vehicle Routing Problem. (2013). Retrieved June 5, 2022, from Lcc.uma.es website: https://neo.lcc.uma.es/vrp/vrp-flavors/stochastic-vrp/

Vehicle Routing Problem — Vehicle Routing Problem. (2013). Retrieved June 5, 2022, from Lcc.uma.es website: https://neo.lcc.uma.es/vrp/vehicle-routing-problem/

VRP with Backhauls — Vehicle Routing Problem. (2013). Retrieved June 5, 2022, from Lcc.uma.es website: https://neo.lcc.uma.es/vrp/vrp-flavors/vrp-with-backhauls/

Wikipedia Contributors. (2022, June 1). Travelling salesman problem. Retrieved June 5, 2022, from Wikipedia website: https://en.wikipedia.org/wiki/Travelling$_{s}alesman_{p}roblem$

Wikipedia Contributors. (2022, May 7). Vehicle routing problem. Retrieved June 5, 2022, from Wikipedia website: https://en.wikipedia.org/wiki/Vehicle$_{r}outing_{p}roblem$