UNIVERSIDAD MIGUEL HERNÁNDEZ

RELACIÓN ENTRE EL PROBLEMA DE ORDENAMIENTO LINEAL DE UNA MATRIZ ASIMÉTRICA Y EL ESCALAMIENTO MULTIDIMENSIONAL DE LA MISMA MATRIZ

Jaime García Amorós
Mercedes Landete Ruiz

TRABAJO FIN DE GRADO - ESTADÍSTICA EMPRESARIAL Curso 2016-2017

FACULTAD DE CIENCAS SOCIALES Y JURÍDICAS DE ELCHE

Profesor tutor del proyecto:

Dra. Mercedes Landete Ruiz

Profesora del Departamento de Matemática, Estadística e Informática. Área de Estadística e Investigación Operativa. Universidad Miguel Hernández de Elche, España.

Agradecimientos:

Principalmente me gustaría dar las gracias a mi tutora
Mercedes Landete Ruiz por guiarme, con paciencia,
a través de todo el transcurso de este trabajo:
Desde el día que contacté con ella para que fuese mi tutora, hasta hoy.
Gracias Mercedes.
Tampoco me gustaría olvidarme de mi familia y amigos,
de todos y cada uno de los profesores que he tenido a lo largo de estos años,
de mis compañeros de clase, personal de la universidad…
Ya que sin todos ellos, seguramente no estaría hoy aquí, ni sería quién soy.

Gracias a todos.

ÍNDICE DE CONTENIDO

1. INTRODUCCIÓN

En este estudio se van a analizar dos formas distintas de ordenación de un conjunto de n elementos. Vamos a analizar los resultados del Problema de Ordenamiento Lineal (de ahora en adelante LOP por sus siglas en inglés *Linear Ordering Problem*) y del Escalamiento Multidimensional (de ahora en adelante MS por sus siglas en inglés *Multidimensional Scaling*). El primero de ellos ordena estrictamente los elementos, y el otro, representa gráficamente en un mapa la distancia entre los mismos.

A través de este estudio, vamos a poder analizar si existe alguna relación entre los resultados de ambos métodos, y de esta forma, ver si el Escalamiento Multidimensional se puede usar para agilizar el cálculo del Problema de Ordenamiento Lineal, ya que este último es más costoso computacionalmente que el primero, es decir, necesita más tiempo de computación para tratar una matriz de la misma dimensión.

2. OBJETIVO

Cómo hemos avanzado antes, el objetivo principal de este estudio, consiste en analizar los resultados del LOP y del MS, y a partir de aquí ver si existe alguna relación entre ambos.

A priori, creemos que debe existir relación entre ambos métodos ya que en el mapa perceptual del MS los elementos similares, con valor pequeño en la posición de la matriz a analizar, se representan por puntos cercanos y en un ordenamiento lineal los elementos similares también han de estar próximos en

la ordenación óptima. A priori, sin embargo, no sabemos cómo se plasmará en un gráfico el hecho de que en un ordenamiento uno sea el mejor y otro el peor: ambos deberían estar alejados en el mapa perceptual pero no sabemos en qué cuadrante del mapa. Tampoco intuimos si el origen de coordenadas en el mapa perceptual tiene alguna implicación en el ordenamiento lineal.

En función de lo que observemos, un objetivo secundario es proponer algún método de resolución del LOP que use la solución del MS. No nos planteamos un método para resolver el MS que use el LOP porque el LOP es mucho más costoso computacionalmente. En breve, resolver el LOP implica resolver un problema de optimización combinatoria con variables binarias mientras que resolver el MS sólo implica el cálculo de autovalores y autovectores de una matriz así como su inversa.

Otros objetivos son aprender a usar CPLEX para la resolución de problemas de optimización combinatoria lineales. Este software no se enseña en el grado y es de gran interés para la resolución de problema lineales.

3. DATOS

Como hemos explicado arriba, el objetivo de este estudio es analizar los resultados del LOP y los resultados del MS para poder ver si guardan algún tipo de relación y así saber si ambos métodos podrían usarse de manera complementaria.

Para hacer las primeras pruebas vamos a utilizar los resultados del LOP para instancias ya resueltas en la literatura y extraídas de la web. Así el experimento es reproducible y tiene mayor interés.

Para ello nos hemos descargado varios ficheros de texto con los datos utilizados para algunos de estos estudios (de los que también tenemos los resultados) y los hemos transformado en un formato que me permita leerlos con SPSS para poder realizar el MS con ellos. Para esta transformación hemos programado un Shell Script en Bash para Unix que está descrito en el Anexo 1.

Una vez leídos los ficheros con SPSS hemos procedido a, con los mismos, realizar un MS para ver gráficamente la proximidad entre los distintos elementos y de esta forma concluir si los resultados de ambos métodos coinciden y por tanto, se pueden complementar.

Como describimos arriba, para este estudio hemos utilizado distintos ficheros de datos ya estudiados y resueltos con LOP, descargados de la web sobre investigación del LOP de la universidad de Heidelberg (Alemania).

Cada fichero consiste en una matriz cuadrada asimétrica de costes, que pueden, a su vez, ser consideradas como matrices de distancias.

Este es un conjunto bien conocido de problemas de ordenamiento lineal en el mundo real generados a partir de tablas de input-output de varias fuentes (Grötschel et al 1984). Las entradas originales en estas tablas no eran necesariamente enteras, pero para el LOP fueron escaladas a valores enteros.

Disponemos de 7 ficheros o matrices input-output distinas, descargadas de la web dedicada al LOP de la universidad de Heidelberg http://comopt.ifi.uni-heidelberg.de/software/LOLIB/:

1. be75eec (BELGIAN I/O MATRIX 1975 - IMPORTATIONS FROM THE EUROPEAN CONSUMER CENTRE)
2. be75np (BELGIAN I/O MATRIX 1975 - NATIONAL PRODUCTION)
3. be75oi (BELGIAN I/O MATRIX 1975 - OTHER IMPORTATIONS)
4. be75tot (BELGIAN I/O MATRIX 1975 - TOTAL)
5. stabu1 (INPUT-OUTPUT-TABELLE 1970 ZU AB-WERK-PREISEN – PRODUCCIÓN DOMÉSTICA)
6. stabu2 (INPUT-OUTPUT-TABELLE 1974 ZU AB-WERK-PREISEN - PRODUCCIÓN DOMÉSTICA)
7. stabu3 (INPUT-OUTPUT-TABELLE 1975 ZU AB-WERK-PREISEN - PRODUCCIÓN DOMÉSTICA)

3.1. INPUT-OUTPUT MATRICES

Se trata de un conjunto de problemas de optimización estándar, utilizadas para la evaluación, caracterización y medición del rendimiento del algoritmo de optimización

Estas son el tipo de matrices que vamos a utilizar para el análisis.

Las tablas input-output son un instrumento estadístico por el que se desagrega la producción nacional entre los sectores que la han originado y los sectores que la han absorbido; por ello también reciben el nombre de “tablas intersectoriales”, especialmente en el ámbito latinoamericano.

Para verlo más claramente, vamos a ver un ejemplo sencillo de tabla input-output de tres sectores, donde vemos la producción de los sectores agrícola, industrial y servicios, dónde podemos ver cómo se divide la producción de cada sector, entre cada uno de los tres sectores:

Tabla Input-Output			
	Sector Agrícola	Sector Industrial	Sector Servicios
Producción Agrícola	50	85	25
Producción Industrial	60	120	50
Producción de Servicios	30	150	120

Matriz obtenida de la web de la Junta de Andalucía.

En esta tabla input-output podemos ver que de lo que produce el sector agrícola, 50 va para el propio sector agrícola, 85 para el sector industrial y 25 para el sector servicios. Y así con cada uno de los sectores.

Tal y como indica el profesor Rafa Martí y el profesor Gerhard Reinelt en su libro The Linear Ordering Problem, el contenido de las tablas input-output que nosotros vamos a utilizar es el siguiente:

- Tablas belgas de 50 sectores

Estas tablas de entrada-salida (be75eec, be75np, be75oi y be75tot) son de 1975 y contienen 50 sectores para la economía belga.

- Tablas alemanas de 60 sectores

Estas tablas de entrada-salida (stabu1, stabu2 y stabu3) fueron recopiladas por el Statistisches Bundesamt de la República Federal de Alemania durante los años 1970, 1974 y 1975.

En primer lugar, como describimos arriba, nos hemos descargado las matrices de sectores de la siguiente página web: http://comopt.ifi.uni-heidelberg.de/software/LOLIB/

Nos hemos descargado 7 matrices y cada una venía en formato texto plano tabulado, pero venían con un número de filas y columnas que no correspondía con el original, por lo que he tenido que ordenarlas para que cada matriz tuviese el formato con el que se optimizaron en LOP, para poder así realizar el MS.

A continuación vamos a ver el nombre y descripción de cada una de las matrices, las filas y columnas de origen, y las filas y columnas que deben tener para poder realizar el análisis:

- **be75eec**: Matriz 50x50 de la economía belga de 1975. Viene como 250X10.
- **be75np**: Matriz 50x50 de la economía belga de 1975. Viene como 250X10.
- **be75oi**: Matriz 50x50 de la economía belga de 1975. Viene como 250X10.
- **be75tot**: Matriz 50x50 de la economía belga de 1975. Viene como 250X10.
- **stabu1**: Matriz 60x60 recopilada por el Statistisches Bundesamt de la República Federal de Alemania durante1970. Viene como 360x10.
- **stabu2**: Matriz 60x60 recopilada por el Statistisches Bundesamt de la República Federal de Alemania durante1974. Viene como 360x10.
- **stabu3**: Matriz 60x60 recopilada por el Statistisches Bundesamt de la República Federal de Alemania durante1975. Viene como 360x10.

En el anexo 1 veremos dos de los ficheros de datos originales: Uno de 50 sectores y otro de 60 sectores. Además, en el Anexo 2 veremos el script en Bash Unix que creé para poder transformar las matrices descargadas de la web, al formato óptimo para trabajar con ellas.

Una vez realizada la conversión de las matrices al formato óptimo para trabajar con ellas, con el script del Anexo 2, procedí a traspasar los datos a Excel para poder así poner a cero la diagonal y poder importar las matrices en SPSS con mayor facilidad.

4. LINEAR ORDERING PROBLEM

4.1. Introducción

Tal y como explica el Doctor Roy Wesley Tromble en su tesis para The Johns Hopkins University, el Linear Ordering Problem (LOP) es uno de los problemas clásicos de optimización combinatoria que ya estaba clasificado como NP-Duro en 1979 por Garey Y Johnson.

Ha recibido considerable atención en diversas áreas de aplicación que van desde la arqueología y la economía, hasta la psicología matemática.

En este apartado del trabajo, explicaremos en qué consiste el LOP y examinaremos las aplicaciones principales para la misma.

4.2. ¿Qué es?

El Problema de Ordenamiento Lineal es un problema de optimización de permutación, es decir, busca la optimización de su función objetivo, a través de la permutación en el orden de sus elementos.

En su versión gráfica el LOP se define **como sigue**:

Sea **Dn = (Vn, An)** un diagrama de n nodos con la propiedad de que para cada par de nodos **i** y **j** hay un arco **(i, j)** de **i** a **j** y un arco **(j, i)** de **j** a **i**.
Un torneo **T** en **An** consiste en un subconjunto de arcos que contiene para cada par de nodos **i** y **j** un arco **(i, j)** o arco **(j, i)**, pero no ambos.

Además el torneo es **acíclico**, es decir, no hay arcos que vayan de la siguiente forma **{(v1,v2),(v2,v3),(v3,v4),…,(vk,v1)}**. Esto quiere decir que hay **transitividad** y por tanto si existen los arcos **(v1,v2)** y **(v2,v3)**, por transitividad, no puede existir el arco **(v3,v1)**, si no el arco **(v1,v3)**.

Un ordenamiento lineal de los nodos **{1,2, . . . , n}** es una clasificación de los nodos dada como una permutación de los mismos, es decir, una biyección del conjunto hacia sí mismo.

Por lo general, las relaciones de ordenación se ponderan y tenemos pesos **Xij** dando el beneficio o coste resultante cuando el nodo **i** se clasifica antes del nodo **j**.

Alternativamente, el LOP puede definirse como un problema de triangulación de matrices.

Dada una **matriz cuadrada H(n, n) = (Hij)**. Determinar una permutación simultánea de filas y columnas de H, de modo que la **suma de los valores que quedan por encima de la diagonal, sea lo más grande posible**.

Aunque se define como una función, será útil pensar en una permutación π como la secuencia:

$$\pi = \pi_1 \ \pi_2 \ \ldots \ \pi_n$$

Es decir: π se define por el número que asigna a 1, seguido por el número que asigna a 2, etc.

Por ejemplo: Si π es 3 1 2, entonces π (1) = 2, π (2) = 3 y π (3) = 1, ya que para llegar a este orden partiendo de la matriz original [1 2 3] el elemento 1 tiene que moverse a la segunda posición, el 2 a la tercera y el 3 a la primera.

Esto quiere decir que el orden óptimo de los elementos sería: primero el elemento 3, seguido del elemento1 y por último el elemento 2.

En definitiva, si tenemos **n** elementos, el número de todas las posibles permutaciones es **n** factorial; por lo que el objetivo del LOP es encontrar la permutación óptima de elementos (de entre las posibles combinaciones) que maximice la suma de los valores que quedan por encima de la diagonal.

Esto se modeliza con una familia de variables y tres restricciones.

Las variables son Xij y son variables binarias que toman el valor 1 si i va delante de j y toman el valor 0 si j va delante de i.

Las 3 restricciones son las siguientes:

1- Si la variable X12 está por encima de X21(esto quiere decir que X12 vale 1), la variable X21 debe estar por debajo de la variable X12 (es decir, X21 vale 0). Por tanto, la suma de las variables X12 + X21, debe ser 1.

2- Transitividad. Si la variable X12 está por encima de la variable X23 (X12 es 1) y a su vez, la variable X23 está por encima de la variable X31 (X23 es 1 y X31 es 0), por transitividad, la variable X12 está por encima de la variable X31 y por tanto la suma de X12 + X23 + X31 debe ser como máximo 2.

3- Los valores que puede tomar cada variable pueden ser 0 ó 1.

Matemáticamente, se representaría de la siguiente manera:

$$\max \sum_{(i,j)\in A_n} c_{ij}x_{ij}$$

$$x_{ij} + x_{ji} = 1, \text{ for all } i,j \in V_n, i < j,$$

$$x_{ij} + x_{jk} + x_{ki} \leq 2, \text{ for all } i,j,k \in V_n, i < j, i < k, j \neq k,$$

$$x_{ij} \in \{0,1\}, \text{ for all } i,j \in V_n.$$

C es la matriz de costes o distancias, Vn es el conjunto de elementos y An es el conjunto de arcos.

4.3. APLICACIONES

El LOP se ha ido utilizando en distintos estudios e investigaciones desde hace muchos años. Entre algunas de las aplicaciones que se le dan al LOP, podemos encontrar las siguientes:

4.3.1. Relaciones de ancestros óptimos ponderados

El origen de este problema es un estudio antropológico en el que se desea especificar un ordenamiento cronológico global de datos antiguos del cementerio.

Considere un cementerio consistente en muchas tumbas individuales. Cada sepultura contiene artefactos hechos de diferentes tipos de cerámica. Como las tumbas se hunden durante los años y se reutilizan, es una suposición razonable que la profundidad de un tipo de cerámica está relacionada con su edad, así que cada sepulcro da un orden parcial de los tipos de alfarería contenidos en él.
En función de la tumba, la tarea de calcular una ordenación con tan pocas contradicciones como sea posible equivale a resolver un problema de ordenamiento lineal donde los nodos corresponden a los tipos de cerámica y los pesos de arco equivalen a cada relación de precedencia.

4.3.2. Percepción de la Corrupción

La organización Transparency International publica un informe anual con el índice de corrupción percibida, que clasifica a más de 150 países según su nivel percibido de corrupción.

Este índice se calcula a partir de evaluaciones de expertos y encuestas de opinión. Las respectivas evaluaciones y encuestas sólo consideran un subconjunto de todos los países y puede, por lo tanto, también ser visto como un orden parcial. El problema de ordenamiento lineal es utilizado para agregar estos rankings parciales. Esto muestra que la solución del problema de ordenación lineal, concuerda con la clasificación del índice en gran medida.

4.3.3. Ranking en torneos deportivos

Para poder verlo más claramente, vamos a ver un ejemplo sobre esta aplicación:

Si tenemos una matriz cuadrada de las selecciones de fútbol, con los partidos que cada selección (fila) ha ganado a cada selección (columna):

	España	Brasil	Alemania	Belgica
España		30	16	41
Brasil	63		43	16
Alemania	51	70		47
Belgica	43	70	30	

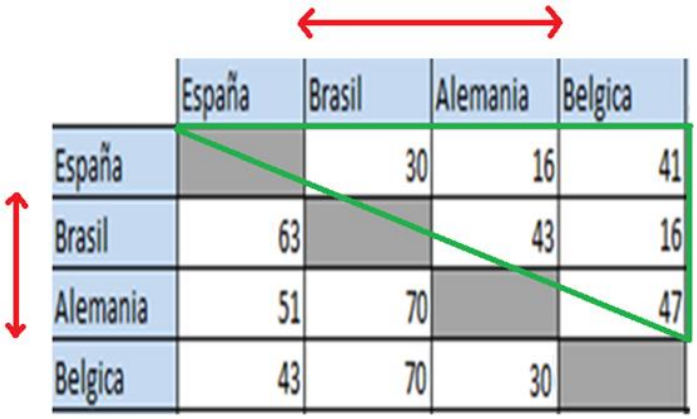
Donde **Vn** sería el conjunto de elementos, que en este caso sería 4, ya que es una matriz cuadrada y las mismas selecciones de fútbol que hay en las filas, están en las columnas y por tanto el número máximo de permutaciones posibles sería 4 factorial (4!), es decir 4x3x2x1=24 combinaciones.

An sería el total de variables. En este caso dónde tenemos una matriz cuadrada con 4 elementos, An sería 4x4=16, a esto le restamos las variables que conforman la diagonal (ya que no tienen valores) y por tanto el número de variables sería 16-4=12.

C seria el valor numérico que toma cada una de las 12 variables. Es decir, C de España-Brasil (C12) sería 30, C13=16, C14=41…

X sería el peso que toma cada variable en función de su orden. Es decir, si X12 (España-Brasil) está por encima de X21 (Brasil-España), entonces X12 valdrá 1 y X21 valdrá 0 (para este ejemplo).

Una vez tenemos esto, el LOP busca la permutación óptima de filas y columnas, que dé como resultado la maximización de la suma de los valores que quedan por encima de la diagonal, es decir, cambia la posición de las filas y columnas para que los valores que quedan por encima de la diagonal de la matriz, den como resultado la mayor suma que se podría dar con cualquier combinación de las 24 posibles.



	España	Brasil	Alemania	Belgica
España		30	16	41
Brasil	63		43	16
Alemania	51	70		47
Belgica	43	70	30	

En este ejemplo vemos que la suma de los valores que quedan por encima de la diagonal es:

30 + 16 + 41 + 43 + 16 + 47 = ***193.***

Ahora, utilizando el LOP, queremos obtener un número mayor a 193, que a su vez sea el mayor número que podríamos obtener con cualquier combinación de filas y columnas. Esto nos ordenará las selecciones de fútbol de mejor a peor, es decir, nos hará un ránking de selecciones.

Dado que este ejemplo es pequeño, vamos a ver como se resolvería en una hoja de cálculo.

Vemos que el número máximo de permutaciones posibles, es 4!= 24.

Lo que vamos a hacer es dibujar las 24 matrices distintas en un Excel y ver cuál de todas obtiene una suma mayor en los valores por encima de la diagonal (Apéndice 1).

Tras representar las matrices de las distintas combinaciones o permutaciones posibles, podemos observar que la permutación que maximiza el resultado de la suma de los valores que quedan por encima de la diagonal, es la siguiente:

	Alemania	Bélgica	Brasil	España
Alemania		47	70	51
Bélgica	30		70	43
Brasil	43	16		63
España	16	41	30	

Cuyo resultado de la suma de los elementos que quedan por arriba de la diagonal es de 344.

Podemos ver que el orden original de la matriz era:

1- España
2- Brasil
3- Alemania
4- Bélgica

El resultado que obtenemos con el LOP sería π = 3 4 2 1. Por tanto, con los datos de los que disponíamos en la matriz de arriba, y utilizando LOP, podemos concluir que el ránking de selecciones de fútbol (de mejor a peor) sería:

1- Alemania
2- Bélgica
3- Brasil
4- España

Una vez tenemos esto, ya podemos saber qué variables valen 1 y qué variables valen 0.

- Variables que valen 1: Todas la que quedan por encima de la diagonal
- Variables que valen 0: Todas las que quedan por debajo de la diagonal.

Matriz original:

	España	Brasil	Alemania	Bélgica
España		30	16	41
Brasil	63		43	16
Alemania	51	70		47
Bélgica	43	70	30	

Matriz ordenada linealmente:

	Alemania	Bélgica	Brasil	España
Alemania		X34=1 47	X32=1 70	X31=1 51
Bélgica	X43=0 30		X42=1 70	X41=1 43
Brasil	X23=0 43	X24=0 16		X21=1 63
España	X13=0 16	X14=0 41	X12=0 30	

Con esto y la fórmula matemática del LOP, podemos comprobar que el resultado óptimo cumple las restricciones del modelo:

$$\max \sum_{(i,j)\in A_n} c_{ij}x_{ij}$$
$$x_{ij} + x_{ji} = 1, \text{ for all } i,j \in V_n, i < j,$$
$$x_{ij} + x_{jk} + x_{ki} \leq 2, \text{ for all } i,j,k \in V_n, i < j, i < k, j \neq k,$$
$$x_{ij} \in \{0,1\}, \text{ for all } i,j \in V_n.$$

Max

(30x0)+(16x0)x(41x0)+(63x1)+(43x0)+(16x0)+(51x1)+(70x1)+(47x1)+(43x1)+(70x1)+(30x0)=344

s.a.

X12+X21= 0 + 1 =1, X13+X31= 0 +1 = 1, X14+X41= 0 + 1 = 1,
X23+X32= 0 + 1 =1, X24+X42= 0+1 =1,
X43+X34= 0 + =1

X12 + X23 + X31 = 0+0+1=1, X13+X32+X21= 0+1+1=2,
X14+X42+X21=0+1+1=2…

X entre 0 y 1.

Tras resolver el problema de ordenamiento linear asociado a este ejemplo, podemos observar fácilmente que se cumplen las restricciones, por lo tanto el orden obtenido es el orden que maximiza la suma de los valores que quedan por encima de la diagonal de la matriz y, por tanto, es el orden óptimo de selecciones de fútbol.

5. ESCALAMIENTO MULTIDIMENSIONAL

Desde hace unos años hasta ahora, la gran cantidad de datos que se general y su fácil acceso, ha provocado que en la mayor parte de las investigaciones, se analicen conjuntos grandes de datos, y por tanto, utilizando técnicas multivariantes para ello, por lo que las técnicas de análisis multivariante tienen cada vez una importancia mayor en las investigaciones y estudios.

Dentro del gran conjunto de técnicas multivariantes, encontramos el Escalamiento Multidimensional (Multidimensional Scaling (MS)).

El MS tiene su origen a principios del siglo XX y consiste en una técnica de análisis multivariante que intenta representar en un espacio geométrico de pocas dimensiones, las proximidades entre un conjunto de estímulos u objetos.

En este trabajo se pretende dar una visión general del funcionamiento del MS, viendo si existe relación con los resultados de otras técnicas como son el Linear Ordering Problem (LOP), de modo que pueda servir como alternativa y como complemento a la misma en cualquier investigación que utilice dichas técnicas, o por el contrario no guardan relación alguna.

5.1. Introducción

El MS es una técnica que busca la representación espacial sobre un mapa, de un conjunto de objetos o estímulos (candidatos, productos…) sobre los que se desea analizar la posición relativa.

La intención del MS es convertir las preferencias u opiniones de un conjunto de individuos sobre un conjunto de objetos o estímulos, en valores (distancias) susceptibles de poder representarse en un espacio multidimensional.

El MS se basa en la comparación de estos estímulos u objetos, de modo que si los individuos opinan que los objetos X e Y son los más parecidos, entonces el MS deberá representar gráficamente a los objetos X e Y de modo que la distancia entre ellos sea la distancia más pequeña existente entre cualquier pareja de objetos.

Actualmente, las técnicas de MS permiten una enorme cantidad de datos de entrada distintos (matrices de proximidad, correlaciones, tablas de contingencia…).

En definitiva, el Escalamiento Multidimensional es una técnica multivariante que genera un mapa aproximado, a partir de las similitudes de un conjunto de objetos o estímulos.

5.2. El modelo general de Escalamiento Multidimensional

De manera general, podríamos decir que, como entrada, el MS toma una matriz de proximidades $\Delta \in Mnxn$, donde n es el número de objetos de la matriz.

Cada elemento (δij) de la matriz de proximidades (Δ) representa la proximidad entre el objeto i y el objeto j.

$$\Delta = \begin{pmatrix} \delta_{11} & \delta_{12} & \cdots & \delta_{1n} \\ \delta_{21} & \delta_{22} & \cdots & \delta_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{n1} & \delta_{n2} & \cdots & \delta_{nn} \end{pmatrix}$$

Una vez tenemos esta matriz de proximidades, el MS nos crea otra matriz, en este caso una matriz de dimensiones $X \in Mnxm$, donde n sigue siendo el número de elementos, y m es el número de dimensiones.

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{pmatrix}$$

Cada valor obtenido para xij, representa la coordenada del objeto i en la dimensión j.

Una vez tenemos también esta matriz X de dimensiones, ya se puede proceder al cálculo de la distancia entre dos objetos i y j. Para ello basta con aplicar la fórmula general de la distancia de Minkowski:

$$d_{ij} = \left[\sum_{t=1}^{m} (x_{it} - x_{jt})^p \right]^p$$

dónde p podrá ser cualquier valor entre 1 e infinito.

Una vez tenemos estas distancias, ya podemos obtener la matriz de distancias que llamaremos D∈Mnxn:

$$D = \begin{pmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{nn} \end{pmatrix}$$

El MS nos deberá proporcional la solución que cumpla que haya la máxima correspondencia posible entre la matriz inicial de proximidades (Δ) y la matriz obtenida finalmente de distancias (D).

Respecto a la bondad del modelo, utilizamos el Stress.

$$Stress = \sqrt{\frac{\sum_{i,j}(f(\delta_{ij}) - d_{ij})^2}{\sum_{i,j} d_{ij}^2}}$$

La forma de utilizar el Stress es la siguiente:

Cuanto mayor sea la diferencia entre las disparidades y las distancias f(δij) y dij, mayor será el valor del Stress y, el modelo será peor. Por tanto, podemos decir que el Stress no mide la bondad del modelo, sino la no bondad del mismo.

El valor mínimo del Stress es 0, mientras que el valor máximo es . $\sqrt{1-(2/n)}$

Las interpretaciones del Stress son las siguientes:

- 0.2 → Pobre
- 0.1 → Aceptable
- 0.05 → Bueno
- 0.025 → Bastante bueno
- 0.0 → Excelente

Aunque estos valores numéricos son un consenso, es evidente que el stress mide lo diferente que son las distancias en el mapa perceptual de las distancias en la matriz de datos original y, por tanto, ha de ser un valor pequeño para que el mapa refleje la información dela matriz.

5.3. Modelos de Escalamiento Multidimensional

En cuanto al MS, existen 2 tipos de modelos:

- **Modelo de Escalamiento Métrico**: En este modelo consideramos que los datos de entrada están en escala de razón o de intervalo.
- **Modelo de Escalamiento No Métrico**: En este modelo consideramos que los datos de entrada están en escala ordinal.

5.3.1. Modelo de escalamiento métrico.

De forma general, en cualquier modelo de escalamiento, partimos de que las distancias son una función de las proximidades ($d_{ij}=f(\delta_{ij})$).

En este modelo concretamente, partimos de del supuesto de que la relación entre distancias y proximidades es lineal, es decir, $d_{ij}=a+b\delta_{ij}$.

Se conoce que el primer escalamiento métrico fue realizado por Torgerson, basándose en el teorema de Young Householder, mediante el cual, partiendo de una matriz de distancias $D \in M_{nxn}$, podemos obtener una matriz de productos escalares $B \in M_{nxn}$.

El proceso consiste en realizar la transformación de la matriz de proximidades $\Delta \in M_{nxn}$ a una matriz de distancias $D \in M_{nxn}$, de modo que se cumplan los tres axiomas de la distancia euclídea.

Estos tres axiomas son los siguientes:

- **No negatividad**: Ningún valor es negativo (dij ≥ 0 = dii).
- **Simetría**: Matriz cuadrada, la cual es igual a su traspuesta (dij = dji).
- **Desigualdad triangular**: Es un teorema de geometría euclidiana que establece que en todo triángulo, la suma de las longitudes de dos lados cualquiera, es siempre mayor a la longitud del lado restante (dij ≤ dik + dkj).

El cumplimiento de los dos primeros axiomas es fácil de cumplir, pero no siempre se cumple el tercero.

Este problema es conocido como "estimación de la constante aditiva" y fue resuelto por Torgerson mediante la estimación del valor mínimo de C, que verifica la desigualdad triangular de la forma que vemos a continuación:

$$c_{min} = \max_{ijk} \{ \delta_{ij} - \delta_{ik} - \delta_{kj} \}$$

Mediante este método, obtenemos las distancias mediante la suma de las proximidades con la constante C, es decir, dij=δij+c.

Para verlo más claro vamos a hacer un ejemplo. Imaginemos que tenemos una matriz de proximidades como la siguiente:

$$\Delta = \begin{pmatrix} 0 & 1 & 5 \\ 1 & 0 & 2 \\ 5 & 2 & 0 \end{pmatrix}$$

Podemos observar fácilmente que esta matriz no verifica el axioma de desigualdad triangular, ya que se incumple que **δ13≤δ12+δ23 (5>1+2)** y por tanto debemos procedes al cálculo del valor mínimo de la constante aditiva **C**.

Para ello, debemos calcular todas las diferencias de todos los subíndices como hemos visto arriba.

En este caso calcularíamos **5-1-2=2** y obtendríamos que el valor mínimo de la constante aditiva **C** es **2**.

Una vez obtenido el valor de C, sumaríamos este valor a la matriz de proximidades y obtendríamos la siguiente matriz de distancias D∈Mnxn:

$$D = \begin{pmatrix} 0 & 3 & 7 \\ 3 & 0 & 4 \\ 7 & 4 & 0 \end{pmatrix}$$

Una vez tenemos esta matriz de distancias, hay que transformarla en una matriz de productos escalares B∈Mnxn de la siguiente forma:

$$b_{ij} = -\frac{1}{2}(d_{ij}^2 - d_{i.}^2 - d_{.j}^2 + d_{..}^2) \quad \text{donde:}$$

$$d_{i.}^2 = \frac{1}{n}\sum_{j=1}^{n} d_{ij}^2 \qquad \text{(distancia cuadrática media por fila)}$$

$$d_{.j}^2 = \frac{1}{n}\sum_{i=1}^{n} d_{ij}^2 \qquad \text{(distancia cuadrática media por columna)}$$

$$d_{..}^2 = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}^2 \qquad \text{(distancia cuadrática media de la matriz)}$$

Finalmente, cuando ya estamos aquí, el último paso es hacer la transformación de la matriz B∈Mnxn a la matriz X∈Mnxm de la siguiente forma:

B=X·X'

dónde X es la matriz que nos proporciona las coordenadas de cada uno de los n objetos, en cada una de las m dimensiones.

Para transformar **B** en **X·X'**, podemos utilizar cualquier método de factorización.

El procedimiento total puede resumirse como sigue:

Δ (Proximidades) → D (Distancias) → B (Productos escalares) → X (coordenadas)

5.3.1.1. ELECCIÓN DEL SOFTWARE A UTILIZAR

El Escalamiento Multidimensional puede realizarse con distintos softwares como R o SPSS. En nuestro caso, para este estudio, hemos decidido utilizar SPSS ya que es el software que hemos utilizado durante el curso. No obstante, más adelante en este mismo trabajo, veremos un ejemplo de cómo hacer Escalamiento Multidimensional con R (Apéndie 3).

El Escalamiento Multidimensional en SPSS puede realizarse con dos métodos distintos: Método ALSCAL y Método PROXSCAL. El método ALSCAL se utiliza para matrices de disimilitudes, mientras que el método PROXSCAL puede utilizarse tanto para matrices de disimilitudes, como para matrices de similitudes.

Particularmente, para nuestro análisis, vamos a utilizar el Escalamiento Multidimensional ALSCAL para matrices de disimilitudes.

La elección de ALSCAL viene motivada por dos razones:

La primera razón es porque a las matrices input-output que vamos a utilizar les hemos eliminado la diagonal dándole valor cero, ya que el LOP no tiene en cuenta la diagonal de la matriz. Al eliminar la diagonal de las matrices, el valor entre un elemento frente a él mismo es el menor, y esto las convierte en matrices de disimilitudes.

La segunda razón es porque tras probar el Escalamiento Multidimensional con ALSCAL y PROXSCAL, hemos visto que ALSCAL se ajustaba mejor y nos daba unos resultados más claros que PROXSCAL.

6. ESTUDIO COMPUTACIONAL

En primer lugar, con los ficheros de datos ya preparados como indicamos en el apartado de Datos, procedimos a importar cada fichero Excel a SPSS y hacer el Escalamiento Multidimensional con el método ALSCAL (este procedimiento está explicado en el Anexo 3) y tras ello, os presentamos a continuación los resultados obtenidos para cada una de las matrices.

Como hemos explicado en el apartado ALSCAL Y PROSXCAL, hemos utilizado el método ALSCAL ya que obtuvimos una mayor bondad y un mayor ajuste que con el modelo PROXSCAL.

En todas las matrices, obtuvimos un Stress menor a 0.001, lo que nos garantiza la bondad del modelo, y un buen ajuste lineal, es decir, la nube de puntos se ajusta bastante bien a la recta de la diagonal. Estos dos factores nos indican que el análisis es adecuado.

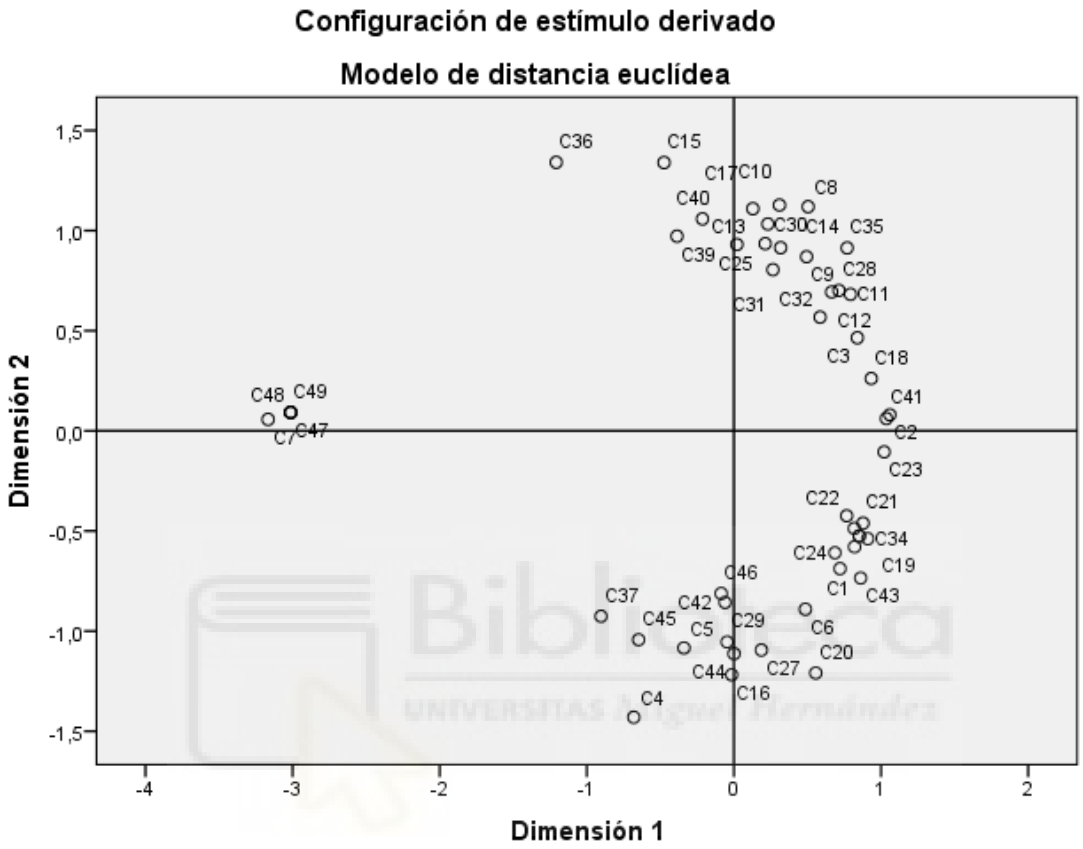
- **Be75eec**

El resultado obtenido por los investigadores del Linear Ordering Problem para optimizar esta matriz y conseguir que la suma de los valores que quedan por encima de la diagonal sea la mayor posible, es el siguiente:

```
Value            : 264940
Value + Diagonals  : 459657
Degree of linearity : 0.879609
Linear ordering    :  7 48 49 33 38 50 11 28 25 27 41 45 37 23 22 20 24 26 42 43 1 5 21 39 34 19 15 18 16 17 14 32 8 9 10 35 36 30 46 29 31 13 12 3 44 2 40 4 6 47
Number of b&c nodes :           1
Total time       :      0:04.23
```

Esto quiere decir, que la primera línea y columna sería la 7, después la 48, 49 y así hasta la última que sería la fila y columna 47. Además, la suma óptima es 264940.

Tras esto, hemos realizado el MS en SPSS y hemos obtenido un valor del Stress que garantiza la bondad del modelo, buen ajuste lineal y siguiente mapa de distancias:



Podemos ver que el MS nos separa por una parte los elementos 7, 48, 49, 47 donde nos dice que estos son los más similares entre ellos y a la vez los más distintos a los resto de elementos, que el SPSS los grafica al lado contrario de estos.

En la resolución del LOP podemos ver que los sectores 7,48 y 49 están los primeros, por lo que podemos decir que en esta matriz, el MS coincide con el LOP en la proximidad de los primeros sectores, pero no coincide en el resto, ya que no encontramos un patrón de comportamiento que nos relacione el resultado del LOP con el resultado del MS más allá de lo que hemos visto de los primeros elementos.

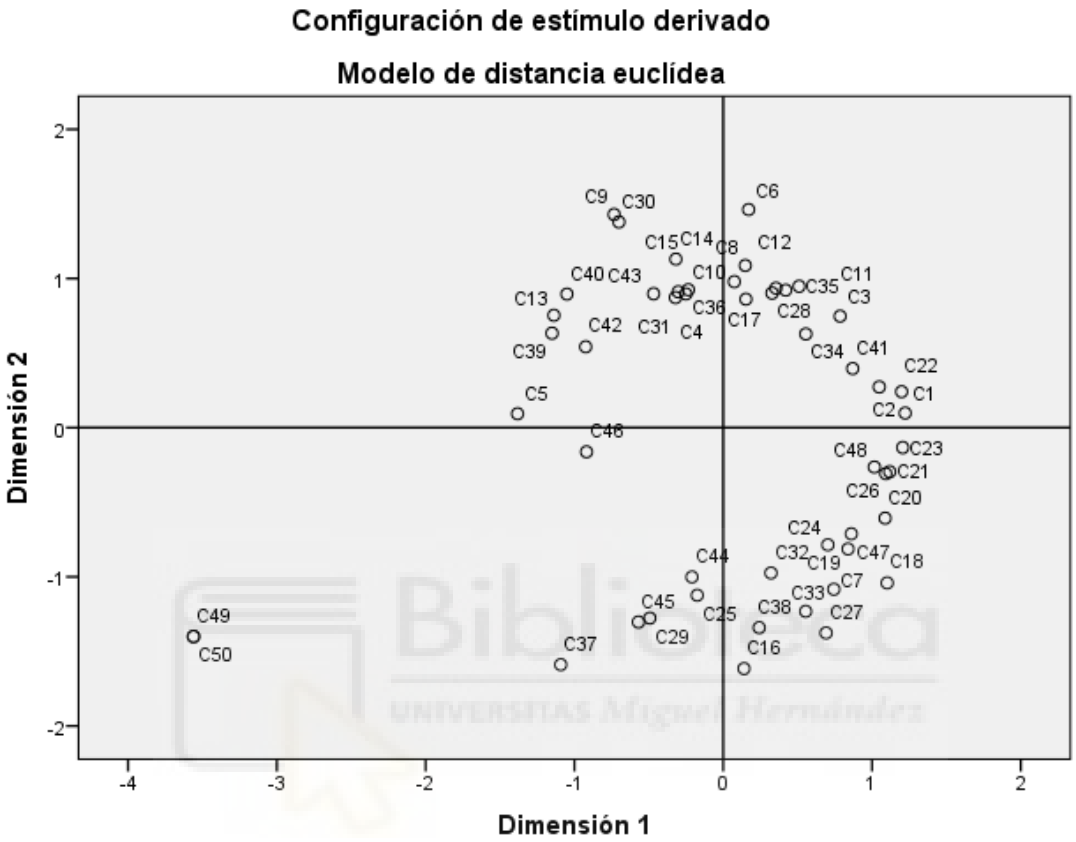
- **Be75np**

El resultado obtenido por los investigadores del Linear Ordering Problem para optimizar esta matriz y conseguir que la suma de los valores que quedan por encima de la diagonal sea la mayor posible, es el siguiente:

```
Value               : 790966
Value + Diagonals   : 977721
Degree of linearity : 0.888153
Linear ordering     :  49 41 19 50 18 48 16 27 37 20 23 21 24 47 33 17 8 3 2 28 30 25 26 1 10 11 13 22 12 15 14 32 39 36 9 34 35 42 31 40 43 45 46 7 29 44 38 5 6 4
Number of b&c nodes :              3
Total time          :        0:10.00
```

Esto quiere decir, que la primera línea y columna sería la 49, después la 41, 19, 50 y así hasta la última que sería la fila y columna 4. Además la suma óptima es 790966.

Tras esto, hemos realizado el MS en SPSS y hemos obtenido un valor del Stress que garantiza la bondad del modelo, buen ajuste lineal y siguiente mapa de distancias:



Podemos ver que el MS nos separa por una parte los elementos 49 y 50 dónde nos dice que estos son los más similares entre ellos y a la vez los más distintos al resto de elementos, que el SPSS los grafica al lado contrario de estos.

En la resolución del LOP podemos ver que los sectores 49 y 50 están entre los 4 primeros, por lo que podemos decir que en esta matriz, el MS coincide con el LOP, aunque menos que en la anterior, en la proximidad de los primeros sectores, pero no coincide en el resto, ya que no encontramos un patrón de comportamiento que nos relacione el resultado del LOP con el resultado del MS más allá de lo que hemos visto de los primeros elementos.

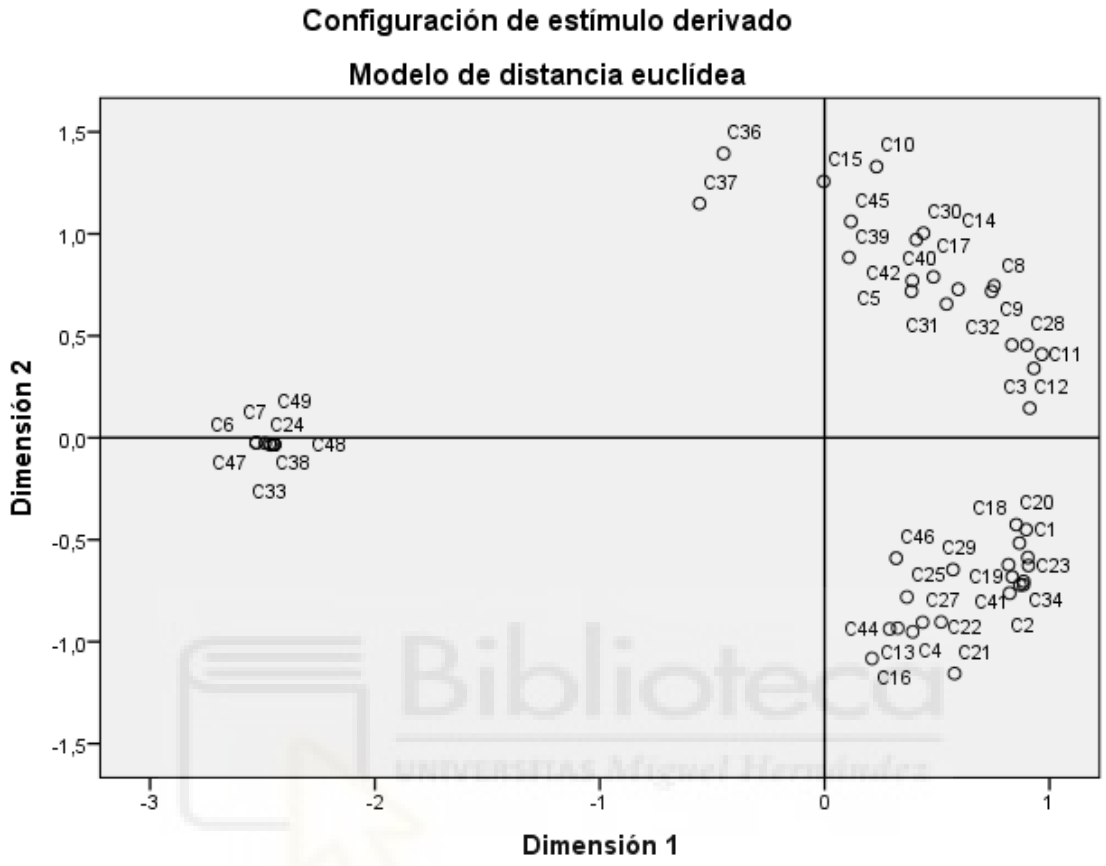
- **Be75oi**

El resultado obtenido por los investigadores del Linear Ordering Problem para optimizar esta matriz y conseguir que la suma de los valores que quedan por encima de la diagonal sea la mayor posible, es el siguiente:

```
Value               : 118159
Value + Diagonals   : 300989
Degree of linearity : 0.923334
Linear ordering     :  33 38 48 49 11 22 20 21 3 25 50 39 34 41 42 24 6 43 5 7 47 2 19 15 18 16 17 32 28 14 8 30 31 26 27 45 37 23 1 10 13 12 9 35 40 36 46 44 29 4
Number of b&c nodes :              1
Total time          :        0:11.12
```

Esto quiere decir, que la primera línea y columna sería la 33, después la 38, 48, 49 y así hasta la última que sería la fila y columna 4. Además la suma óptima es 118159.

Tras esto, hemos realizado el MS en SPSS y hemos obtenido un valor del Stress que garantiza la bondad del modelo, buen ajuste lineal y siguiente mapa de distancias:



Podemos ver que el MS nos separa por una parte los elementos 33, 38, 49, 48, 47, 24, 7 y 6 dónde nos dice que estos son los más similares entre ellos y a la vez los más distintos al resto de elementos, que el SPSS los grafica al lado contrario de estos.

En la resolución del LOP podemos ver que los sectores 33, 38, 49, 48 están los primeros, por lo que podemos decir que en esta matriz, el MS coincide con el LOP en la proximidad de los primeros sectores, pero no coincide en el resto, ya que no encontramos un patrón de comportamiento que nos relacione el resultado del LOP con el resultado del MS más allá de lo que hemos visto de los primeros elementos.

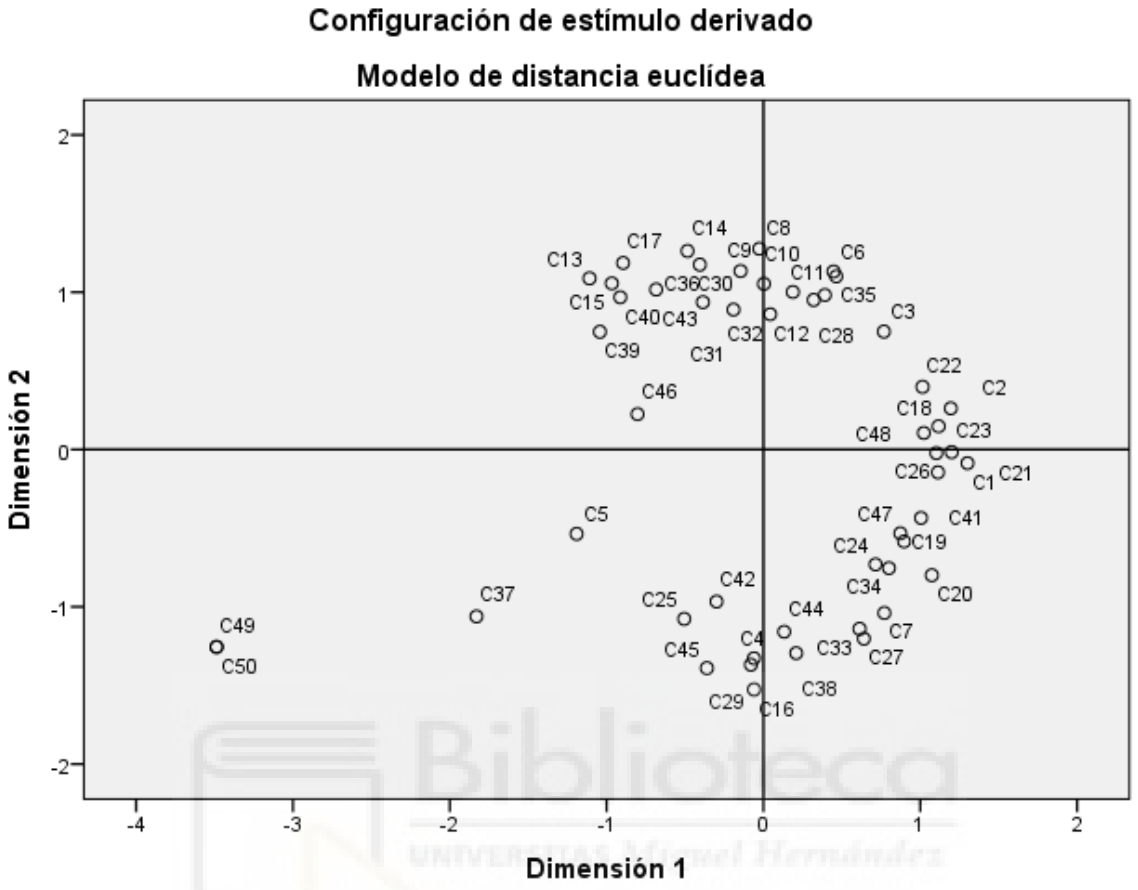
- **Be75tot**

El resultado obtenido por los investigadores del Linear Ordering Problem para optimizar esta matriz y conseguir que la suma de los valores que quedan por encima de la diagonal sea la mayor posible, es el siguiente:

```
Value               : 1127387
Value + Diagonals   : 1691689
Degree of linearity : 0.854245
Linear ordering     :  49 41 50 48 25 27 37 23 24 22 20 21 42 47 33 38 19 11 28 26 1 15 18 16 17 30 14 32 8 31 10 13 12 39 36 9 3 40 43 34 35 45 5 6 2 46 7 29 44 4
Number of b&c nodes :             1
Total time          :       0:05.00
```

Esto quiere decir, que la primera línea y columna sería la 49, después la 41, 50 y así hasta la última que sería la fila y columna 4. Además la suma óptima es 1127387.

Tras esto, hemos realizado el MS en SPSS y hemos obtenido un valor del Stress que garantiza la bondad del modelo, buen ajuste lineal y siguiente mapa de distancias. El mapa de distancias obtenido con el MS para esta matriz es el siguiente:



Podemos ver que el MS nos separa por una parte los elementos 49 y 50 dónde nos dice que estos son los más similares entre ellos y a la vez los más distintos al resto de elementos, que el SPSS los grafica al lado contrario de estos.

En la resolución del LOP podemos ver que los sectores 49 y 50 están entre los 3 primeros, por lo que podemos decir que en esta matriz, el MS coincide con el LOP en la proximidad de los primeros sectores, pero no coincide en el resto, ya que no encontramos un patrón de comportamiento que nos relacione el resultado del LOP con el resultado del MS más allá de lo que hemos visto de los primeros elementos.

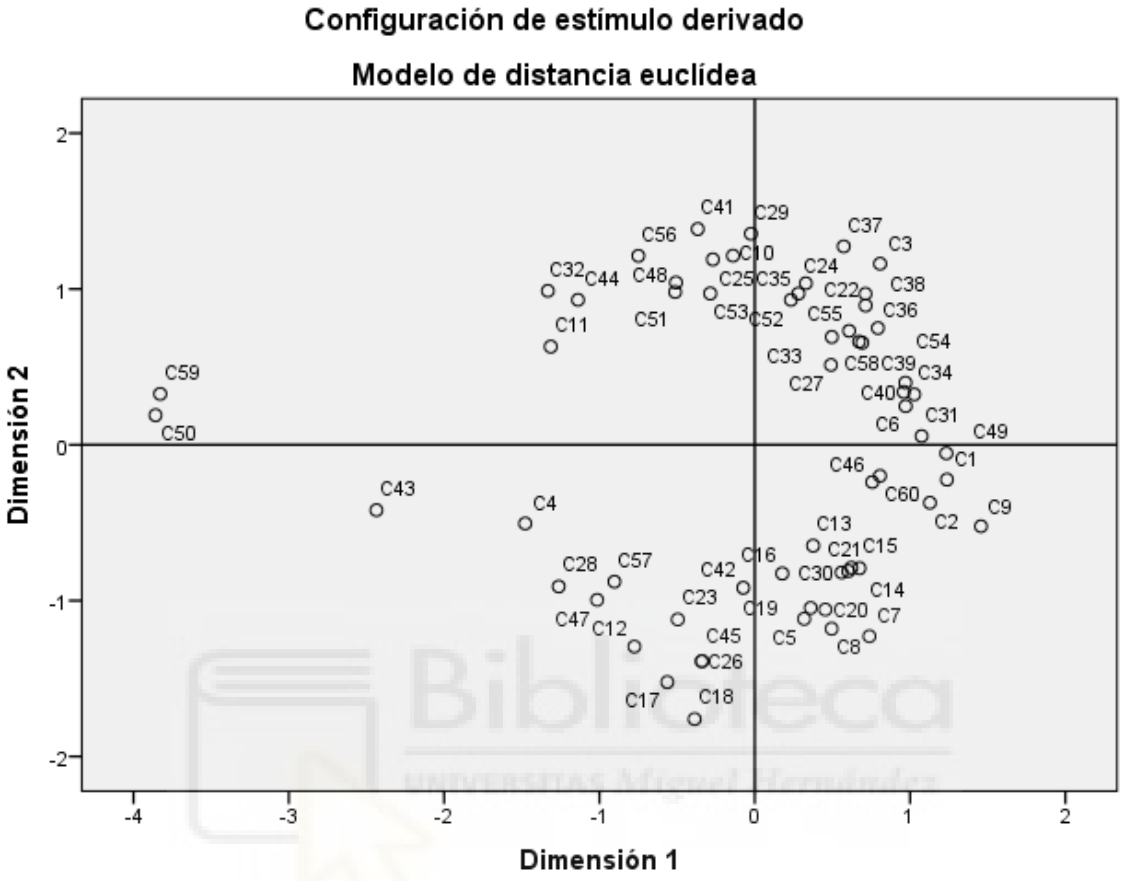
- **Stabu1**

El resultado obtenido por los investigadores del Linear Ordering Problem para optimizar esta matriz y conseguir que la suma de los valores que quedan por encima de la diagonal sea la mayor posible, es el siguiente:

```
  Value              : 422088
  Value + Diagonals  : 579565
  Degree of linearity : 0.831864
  Linear ordering    :  60 58 41 15 46 22 21 52 39 36 35 37 38 40 1 54 29 24 23 27 33 30 3 44 25 26 17 13 28 20 18 14 16 19 12 50 55 34 10 8 42 43 49 53 32 31 4 5 7 2 47 45 51 57 56 48 11
6 9 59
  Number of b&c nodes :             1
  Total time         :       1:42.52
```

Esto quiere decir, que la primera línea y columna sería la 60, después la 58, 41 y así hasta la última que sería la fila y columna 59. Además la suma óptima es 422088.

Tras esto, hemos realizado el MS en SPSS y hemos obtenido un valor del Stress que garantiza la bondad del modelo, buen ajuste lineal y siguiente mapa de distancias:



Podemos ver que el MS nos separa por una parte los elementos 59 y 50 dónde nos dice que estos son los más similares entre ellos y a la vez los más distintos al resto de elementos, que el SPSS los grafica al lado contrario de estos.

En la resolución del LOP podemos ver que el sector 50 está por en medio y el 59 es el último, por lo que podemos decir que en esta matriz, el MS no muestra ningún patrón que nos haga pensar que se guarda alguna relación entre éste y el LOP.

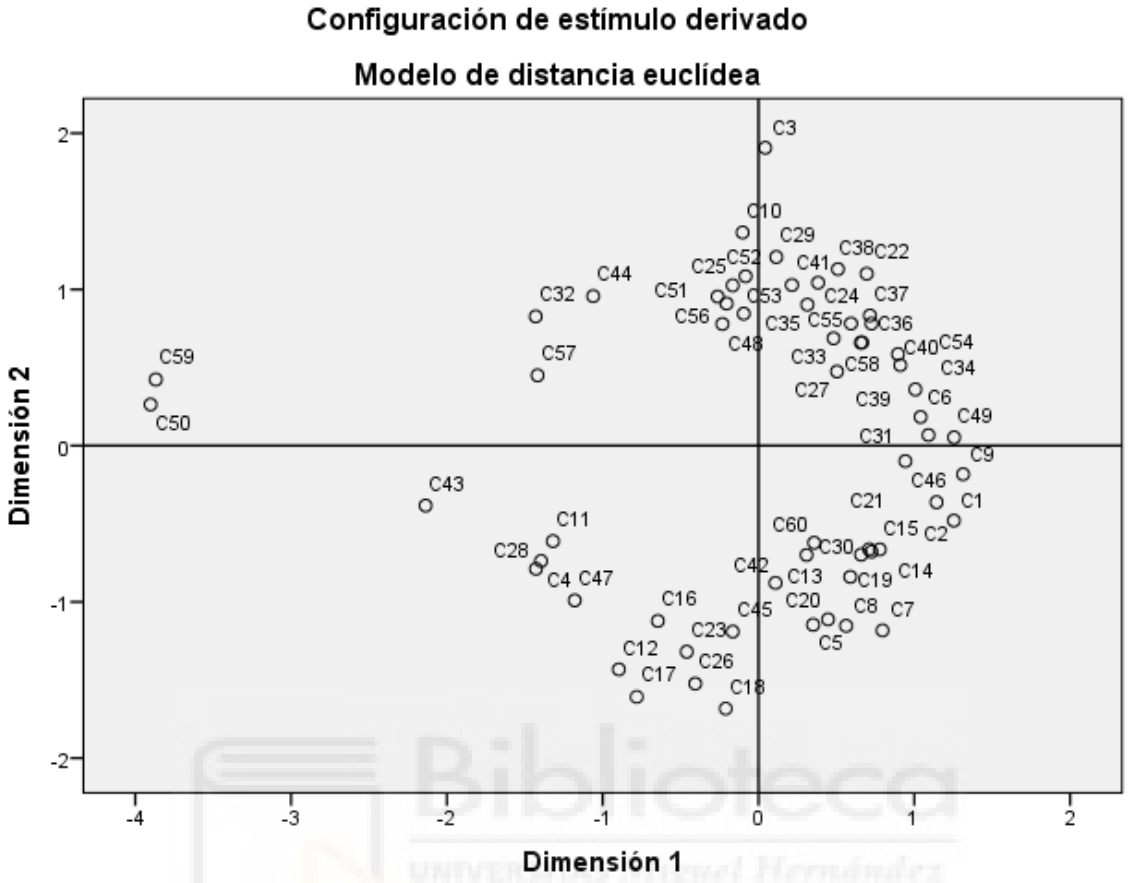
- **Stabu2**

El resultado obtenido por los investigadores del Linear Ordering Problem para optimizar esta matriz y conseguir que la suma de los valores que quedan por encima de la diagonal sea la mayor posible, es el siguiente:

```
  Value               : 627929
  Value + Diagonals   : 867486
  Degree of linearity : 0.839649
  Linear ordering     :  46 60 54 58 22 33 52 39 36 35 37 38 40 1 41 15 21 29 24 23 30 3 44 25 26 27 17 28 20 13 18 14 16 19 12 34 10 8 43 42 50 55 49 53 32 31 6 4 5 7 2 47 45 51 57 56 48
11 9 59
  Number of b&c nodes :             1
  Total time          :       0:50.15
```

Esto quiere decir, que la primera línea y columna sería la 46, después la 60, 54 y así hasta la última que sería la fila y columna 59. Además la suma óptima es 627929.

Tras esto, hemos realizado el MS en SPSS y hemos obtenido un valor del Stress que garantiza la bondad del modelo, buen ajuste lineal y siguiente mapa de distancias:



Podemos ver que el MS nos separa por una parte los elementos 59 y 50 dónde nos dice que estos son los más similares entre ellos y a la vez los más distintos al resto de elementos, que el SPSS los grafica al lado contrario de estos.

En la resolución del LOP podemos ver que el sector 50 está por en medio y el 59 es el último, por lo que podemos decir que en esta matriz, el MS no muestra ningún patrón que nos haga pensar que se guarda alguna relación entre éste y el LOP.

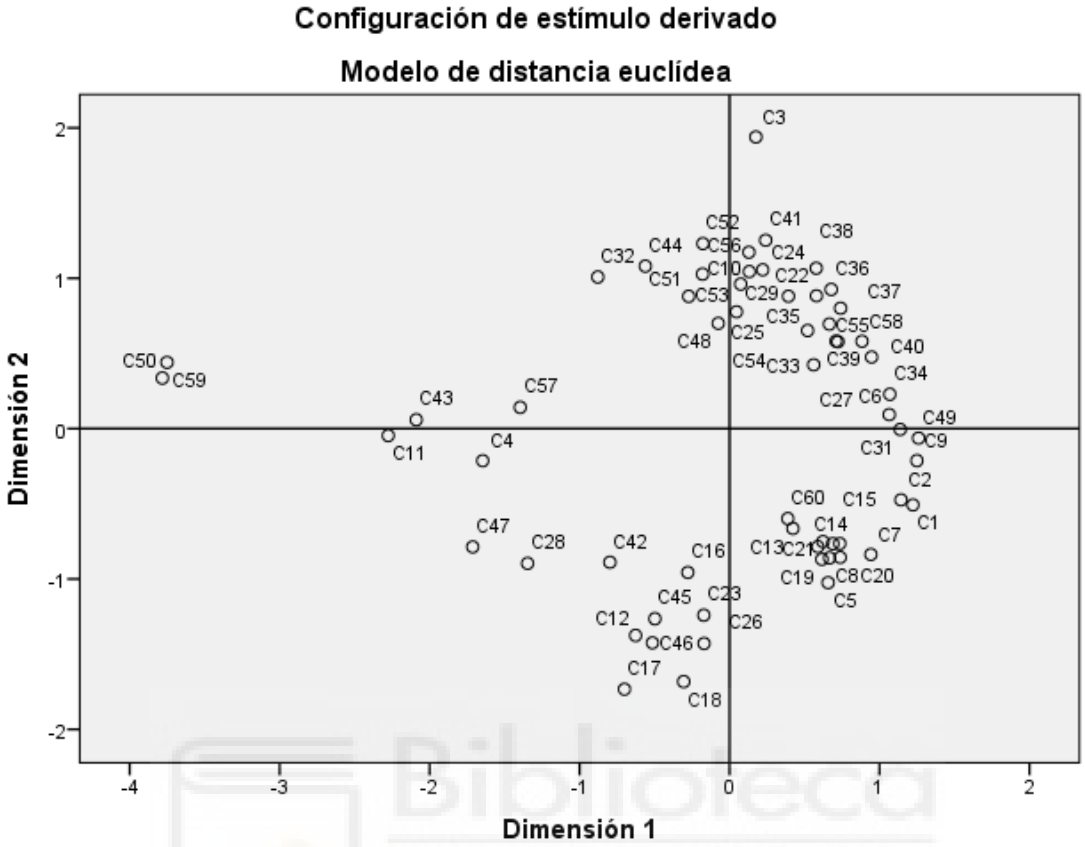
- **Stabu3**

El resultado obtenido por los investigadores del Linear Ordering Problem para optimizar esta matriz y conseguir que la suma de los valores que quedan por encima de la diagonal sea la mayor posible, es el siguiente:

```
  Value               : 642050
  Value + Diagonals   : 895029
  Degree of linearity : 0.837988
  Linear ordering     :  46 60 54 58 22 33 52 39 36 35 37 38 40 1 44 55 41 15 21 29 24 25 23 26 27 17 13 30 3 28 20 59 14 12 34 18 19 10 16 8 42 43 53 32 31 4 5 7 2 47 45 51 48 50 49 57 56
11 6 9
  Number of b&c nodes :             1
  Total time          :       1:32.93
```

Esto quiere decir, que la primera línea y columna sería la 46, después la 60, 54 y así hasta la última que sería la fila y columna 9. Además la suma óptima es 642050.

Tras esto, hemos realizado el MS en SPSS y hemos obtenido un valor del Stress que garantiza la bondad del modelo, buen ajuste lineal y siguiente mapa de distancias:



Podemos ver que el MS nos separa por una parte los elementos 59 y 50 dónde nos dice que estos son los más similares entre ellos y a la vez los más distintos al resto de elementos, que el SPSS los grafica al lado contrario de estos.

En la resolución del LOP podemos ver que el sector 59 está por en medio y el 50 está entre los últimos elementos, por lo que podemos decir que en esta matriz, el MS no muestra ningún patrón que nos haga pensar que se guarda alguna relación entre éste y el LOP.

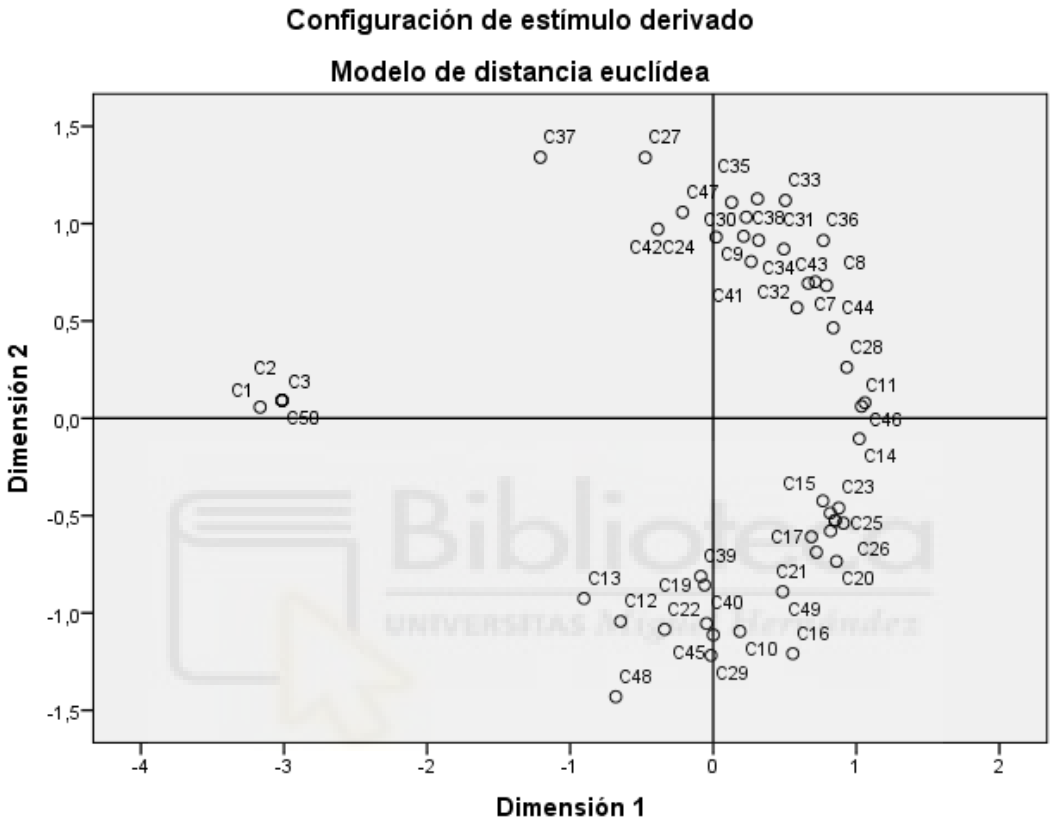
Una vez realizado el Escalamiento Multidimensional de todas nuestras matrices, y viendo que en las cuatro primeras el Escalamiento Multidimensional nos separa algunos de los elementos que el LOP nos ordena entre las primeras posiciones, del resto de objetos de nuestras instancias, hemos decidido volver a hacer el MS con estas cuatro matrices, pero renombrando los elementos (C1, C2, C3…, Cn) según los ordena el LOP en sus soluciones para las mismas, es decir, al elemento que el LOP pone en primera posición, en nuestra matriz lo llamaremos C1, al elemento que el LOP pone en segunda posición, lo llamaremos C2 y así sucesivamente.

Esta medida la hemos adoptado para poder ver más claramente las diferencias y similitudes entre el mapa que nos devuelve el MS y la solución del LOP para cada una de las instancias y, así, tratar de ver qué tipo de relación guarda el LOP con el MS en estas matrices, si es que guarda alguna.

Además, una vez obtenido el mapa del MS de cada una de estas cuatro matrices, hemos dibujado una diagonal en el mismo para tratar de ver si existe algún patrón de comportamiento en torno a esta diagonal.

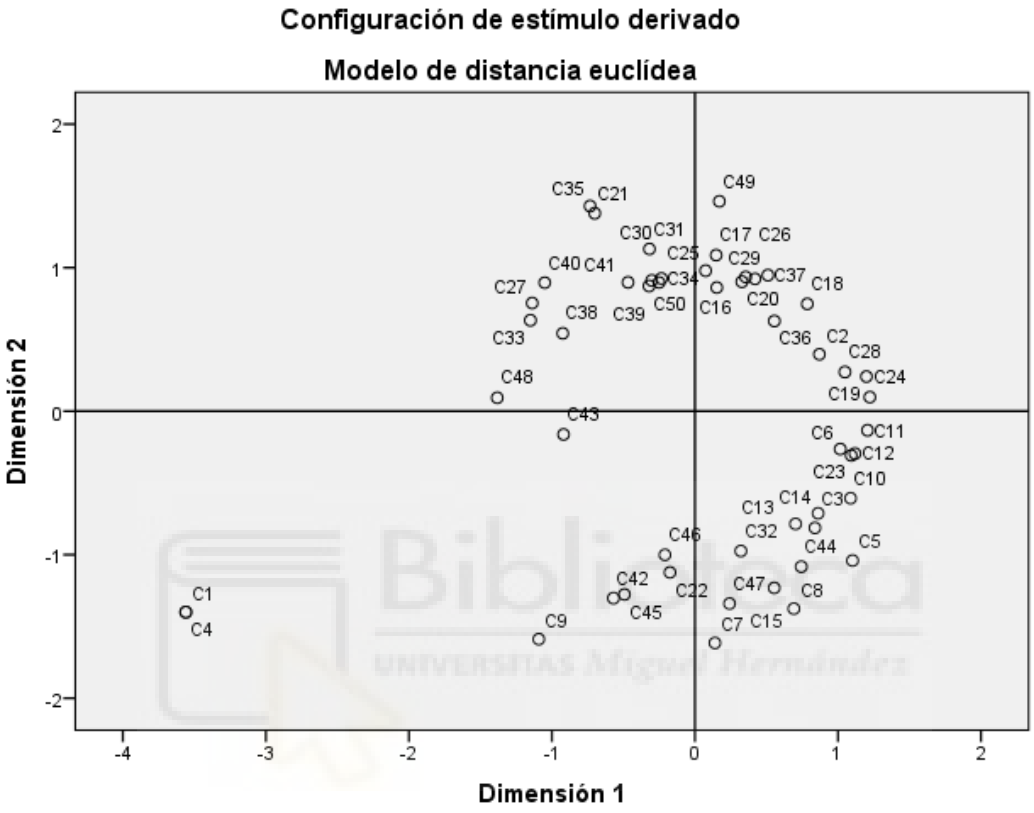
El resultado del MS para estas nuevas matrices renombradas es el siguiente:

- **Be75eec**

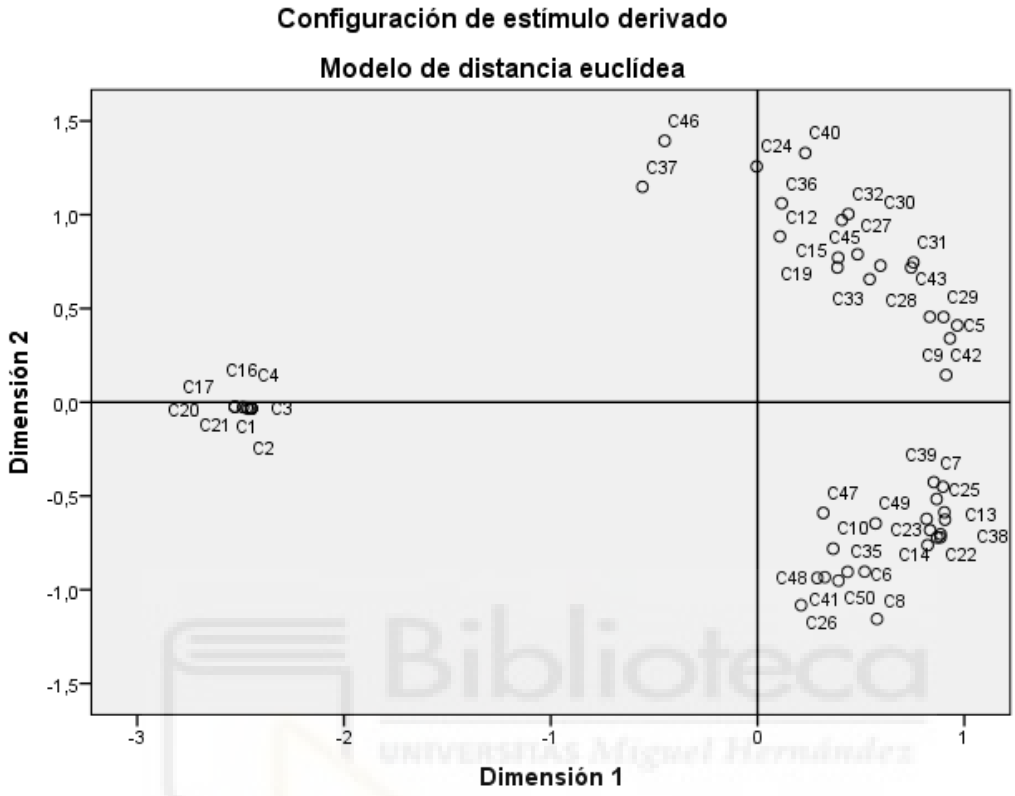

Configuración de estímulo derivado
Modelo de distancia euclídea
Dimensión 2
Dimensión 1
1,5
1,0
0,5
0,0
-0,5
-1,0
-1,5
-4
-3
-2
-1
0
1
2
C37
C27
C35
C33
C47
C30
C38
C31
C36
C42
C24
C9
C34
C43
C8
C41
C32
C7
C44
C28
C11
C46
C14
C15
C23
C25
C17
C26
C39
C21
C20
C13
C19
C12
C22
C40
C49
C16
C45
C10
C48
C29
C2
C1
C3
C50

-

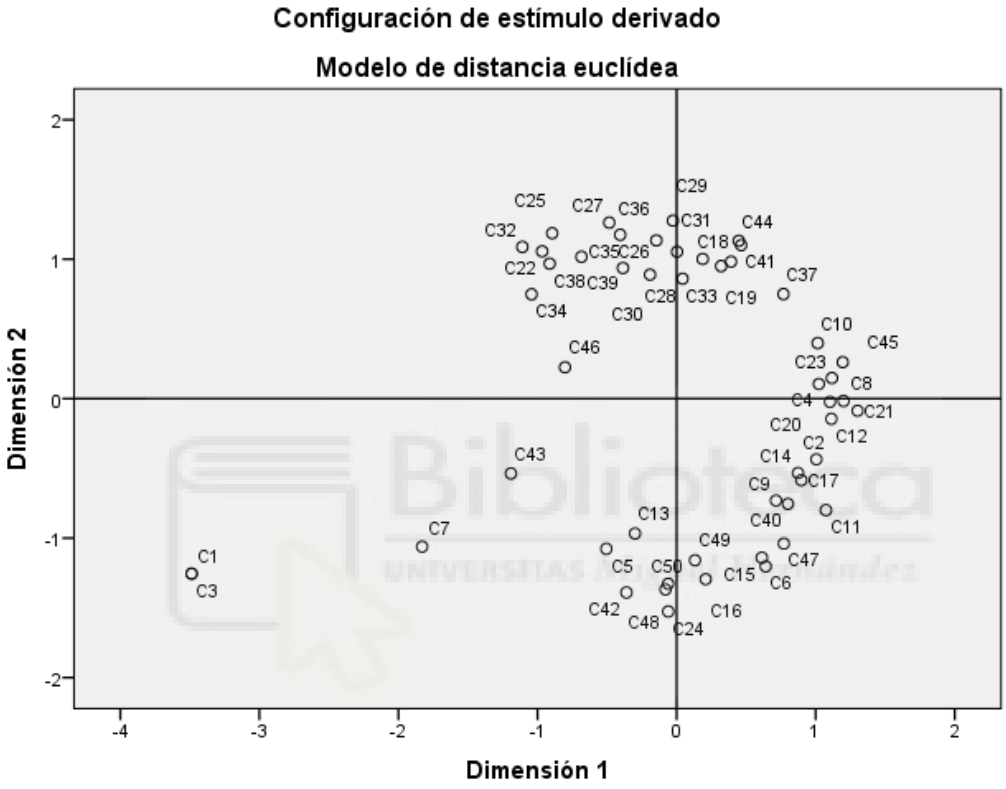
- be75np


Configuración de estímulo derivado
Modelo de distancia euclídea
Dimensión 2
Dimensión 1
2
1
0
-1
-2
-4
-3
-2
-1
0
1
2
C35 C21 C49
C30 C31 C17 C26
C40 C41 C25 C29
C27 C34 C37 C18
C38 C39 C50 C16 C20
C33 C36 C2
C48 C28 C24
C19
C43
C6 C11
C12
C23 C10
C13 C14 C30
C46 C32 C44 C5
C42 C22 C47 C8
C1 C45
C4 C9 C7 C15

- **Be75oi**


Configuración de estímulo derivado
Modelo de distancia euclídea
Dimensión 2
Dimensión 1
1,5
1,0
0,5
0,0
-0,5
-1,0
-1,5
-3
-2
-1
0
1
C46
C40
C24
C37
C36
C32
C30
C12
C27
C45
C31
C15
C43
C19
C33
C28
C29
C5
C9
C42
C16
C4
C17
C20
C21
C1
C3
C2
C39
C7
C47
C49
C25
C13
C10
C23
C38
C35
C14
C22
C48
C6
C41
C50
C8
C26

- **Be75tot**



Tras esto, podemos observar que el mapa perceptual del SPSS para el MS nos separa a un lado un conjunto de elementos que incluye a los mejores del LOP y a otro lado el resto de elementos sin ningún patrón aparente con respecto al LOP.

Esto nos hace ver que existe una relación entre el MDS y otro tipo de problema en la literatura similar al LOP que se llama el Top K Problem. Para entender la relación, estudiamos el Top K Problem a continuación.

7. TOP K PROBLEM y su relación con el MS

En los últimos años ha habido un gran interés en desarrollar técnicas efectivas para la búsqueda y recuperación de datos ad-hoc en bases de datos relacionales, bases de datos de documentos y multimedia, sistemas de información científica, etc.

Un paradigma popular para hacer frente a este problema es el Top K Problem (TKP de aquí en adelante), es decir, la clasificación de los resultados y la devolución de los k resultados con las puntuaciones más altas, en función del atributo que nos interese (en ejemplo de las selecciones de fútbol, este atributo serían los partidos ganados).

A diferencia del LOP, el TKP no nos devuelve la totalidad de elementos ordenados, sino los k primeros elementos de esa lista.

Como bien explica el departamento de ciencia computacional de la Univeridad de Columbia, el TKP o Top K Query, tiene aplicación, por ejemplo, en los motores de búsqueda web como Google.

Consideremos un escenario donde una persona esté interesada en encontrar restaurantes en una determinada zona de Madrid. Este caso puede ser modelado usando una relación con la información sobre diferentes restaurantes. Cada objeto de esta relación tiene una serie de atributos (dirección, precio, puntuación, etc), que están a cargo de diferentes bases de datos web como por ejemplo Google Maps para la dirección, Trip Advisor para el precio y la puntuación, etc.

Si el usuario que está haciendo la búsqueda, vive en Moncloa y está interesado en restaurantes cuyo menú sea de 25 € aproximadamente, este usuario podría hacer una top-10 query dónde sus parámetros son que esté por la zona de Moncloa, que el precio sea de 25 € aproximadamente y que la puntuación sea de 7 (en una escala de 1 a 10).

El resultado de esta búsqueda será la lista de 10 restaurantes con mayor puntuación, que cumplan las especificaciones de este usuario.

En tales escenarios, el acceso a los datos está limitado por las interfaces que proporcionan las fuentes, y las restricciones y costos de acceso a fuentes deben tenerse en cuenta para lograr un procesamiento eficiente de consultas.

Existen algoritmos secuenciales y paralelos que priorizan de forma adaptativa el procesamiento de los datos conocidos para identificar los top k de una consulta dada, haciendo elecciones dinámicas de qué objetos descartar y cuáles procesar, a medida que avanza el procesamiento de consultas

Según lo visto en la sección anterior el mapa perceptual del resultado del MS puede ser una buena solución heurística del Top K problema. Evidentemente habría que depurar la solución inicial con técnicas ad-hoc pero probablemente sería más rápido que resolverlo sin información inicial.

8. LOP Y CPLEX

Como hemos podido analizar, vemos que el resultado que obtenemos al realizar el Escalamiento Multidimensional, se aproxima más al resultado del problema de optimización TPK, que al resultado del Linear Ordering Problem. Sin embargo, el coste computacional del LOP y del TPK es similar. Por ello queremos saber cuál es este coste computacional, si de verdad tiene sentido tener técnicas de resolución alternativas a l modelo de optimización lineal.

En este apartado del estudio, vamos a proceder a comprobar cómo el coste computacional de realizar el Linear Ordering Problem (en este caso con CPLEX) va aumentando considerablemente a medida que aumentamos los elementos de la matriz asimétrica, lo que nos puede dar un indicio de que la utilización previa de un análisis multivariante puede ayudarnos a obtener un punto de partida en la resolución del LOP que nos reduzca el costo computacional del mismo.

Como hemos mencionado arriba, en esta sección vamos a utilizar el software de optimización CPLEX para resolver el LOP de las siete matrices utilizadas en este estudio, además de algunas matrices de mayor volumen (extraídas de la web http://www.optsicom.es/lolib/#instances) para poder ir viendo cómo se va incrementando el tiempo de resolución según va aumentando el tamaño de estas matrices.

8.1 CPLEX

CPLEX es un software de optimización que actualmente pertenece a IBM, pero que fue creado en 1988 por Robert E. Bixby y ofrecido comercialmente por CPLEX Optimization Inc, la cual fue adquirida por ILOG en 1997 y posteriormente IBM compró a ILOG en 2009.

IBM ILOG CPLEX ofrece bibliotecas C, C++, Java, .NET y Python que resuelven problemas de programación lineal (LP) y relacionados.

Resuelve de forma lineal o de forma cuadrática problemas de optimización restringidos, donde se busca el valor de una función objetivo sujeta a una serie de restricciones.

Para resolver problemas de programación lineal, CPLEX implementa optimizadores basados en los algoritmos simplex, entre otros.

8.2 RESOLUCIÓN DE LOP CON CPLEX

En este apartado del estudio, vamos a resolver los problemas de optimización LOP para las 7 matrices utilizadas a lo largo de todo el trabajo, además de algunas otras matrices de mayor volumen, para poder ir viendo cómo se va incrementando el tiempo de resolución según va aumentando el tamaño de estas matrices.

Para realizar este análisis, hemos programado un script en C, que introduciremos en CPLEX y al que le iremos pasando las matrices correspondientes en .txt para optener el resultado óptimo del LOP y el tiempo tardado en lograr el mismo. El script y los detalles del mismo los podemos revisar en el APÉNDICE 2: Resolución de LOP con CPLEX.

Para empezar, vamos a proceder a la resolución de la matriz de ejemplo utilizada anteriormente para explicar el ejemplo del LOP en el ránking de los equipos de fútbol.

La matriz era esta:

	España	Brasil	Alemania	Bélgica
España		30	16	41
Brasil	63		43	16
Alemania	51	70		47
Bélgica	43	70	30	

Recordemos que queremos obtener el valor máximo posible que se puede obtener al sumar los elementos que quedan por encima de la diagonal principal y que para ello CPLEX tendrá que reordenar las filas y columnas de la matriz hasta encontrar dicho valor.

$$\max \sum_{(i,j)\in A_n} c_{ij}x_{ij}$$

$$x_{ij} + x_{ji} = 1, \text{ for all } i,j \in V_n, i < j,$$

$$x_{ij} + x_{jk} + x_{ki} \leq 2, \text{ for all } i,j,k \in V_n, i < j, i < k, j \neq k,$$

$$x_{ij} \in \{0,1\}, \text{ for all } i,j \in V_n.$$

Max

(30x0)+(16x0)x(41x0)+(63x1)+(43x0)+(16x0)+(51x1)+(70x1)+(47x1)+(43x1)+(70x1)+(30x0)=344

s.a.

X12+X21= 0 + 1 =1, X13+X31= 0 +1 = 1, X14+X41= 0 + 1 = 1,
X23+X32= 0 + 1 =1, X24+X42= 0+1 =1,
X43+X34= 0 + =1

X12 + X23 + X31 = 0+0+1=1, X13+X32+X21= 0+1+1=2,
X14+X42+X21=0+1+1=2…

X entre 0 y 1.

Tras introducir el fichero con la matriz de ejemplo, una matriz 4x4, podemos ver que CPLEX nos da el siguiente resultado:

```
Best Integer

    344.0000
```

Y para llegar a dicha solución, ha tardado:

```
0.00 sec.
```

Es decir, que el coste computacional de realizar LOP con CPLEX para una matriz asimétrica de 4x4 es insignificante.

Una vez visto lo que queremos hacer en este apartado, vamos a crear una tabla de valores óptimos y tiempos para las 7 matrices principales del estudio (50x50 y 60x60), una matriz 100x100, una matriz 150x150, otra matriz 200x200 y finalmente una matriz 500x500 para comprobar si el costo computacional a medida que aumenta el volumen de las matrices, también aumenta considerablemente.

MATRIZ	TAMAÑO	VALOR ÓPTIMO	TIEMPO
Be75eec	50x50	Best Integer 264940.0000	0.18 sec.
Be75np	50x50	Best Integer 790966.0000	0.20 sec.
Be75oi	50x50	Best Integer 118159.0000	0.24 sec.
Be75tot	50x50	Best Integer 1127387.0000	0.20 sec.
Stabu1	60x60	Best Integer 422088.0000	0.36 sec.
Stabu2	60x60	Best Integer 627929.0000	0.22 sec.
Stabu3	60x60	Best Integer 642050.0000	0.22 sec.
N100	100x100	Best Integer 83583.0000	1.86 sec.
N150	150x150	Best Integer 183371.0000	7.87 sec.
N200	200x200		Más de 1 día
N500	500x500		Más de 1 día

9. CONCLUSIONES

Tras todo el estudio realizado en este trabajo, hemos aprendido en qué consiste el LOP (Linear Ordering Problem), su utilidad, interpretación y modos de resolución. Hemos visto una interpretación gráfica, otra en términos de matrices que se puede manejar con Excel y una última en términos de problemas de optimización.

Hemos hecho una búsqueda de bases de datos de la literatura del LOP, chequeado con bases de datos de la literatura habitual del mismo si la solución de este depende de algún modo del mapa perceptual del MS (Escalamiento Multidimensional) de la misma matriz, siendo el MS materia estudiada del grado de Estadística Empresarial.

Al observar que la relación entre LOP y MS es escasa, porque sólo discrimina a los elementos con mejor ranking, hemos estudiado otro problema de la literatura que es el TKP (Top K Problem) que sí está más relacionado con MS, ya que el TKP realiza una clasificación de los resultados y la devolución de los k resultados con las puntuaciones más altas, en función del atributo que nos interese.

Además de utilizar métodos estudiados en el grado de Estadística Empresarial (como el MS), hemos aprendido nuevas técnicas durante la realización de este estudio. En este segundo grupo entra el propio LOP, el TKP y CPLEX, ya que hemos resuelto problemas con datos de la literatura habitual del LOP con CPLEX para ilustrar la complejidad computacional de LOP observando como el tiempo de resolución de estos problemas va aumentando significativamente a medida que aumenta el tamaño de las matrices, llegando incluso en las

matrices 200*200 y 500*500 a no poder obtener la resolución de las mismas en un tiempo límite inferior a un día.

En resumen, el contenido del presente TFG se relaciona del siguiente modo con las materias estudiadas en el grado. La asignatura “Modelos de optimización” nos ha permitido comprender el modelo de optimización que se usa para el LOP así como la importancia de las variables binarias. En la asignatura “Minería de datos” aprendimos técnicas de reducción de dimensiones que se relacionan con mapas perceptuales y la utilidad del MS. La asignatura “Base de datos en internet” nos ha permitido obtener bases de datos adecuadas para el estudio computacional de este trabajo.
Además, la asignatura “Introducción a la programación” es el primer paso para aprender el manejo de CPLEX y cómo programar el script que prepara la matriz para ser utilizada en MS.

En definitiva, las materias cursadas y aprendidas durante el grado, han sido indispensables para poder realizar este trabajo, además de para haberme permitido aprender materias y métodos nuevos utilizados en el mismo.

Como trabajo futuro de investigación dejamos pendiente el desarrollo de un algoritmo heurístico o exacto de resolución del TKP y/o del LOP a partir de los mapas perceptuales.

10. Referencias Bibliográficas

- Universidad de Heidelberg. LOLIB. http://comopt.ifi.uni-heidelberg.de/software/LOLIB/ y http://www.optsicom.es/lolib/

- Web Optsicom Project http://www.optsicom.es/lolib/#instances

- Tesis de Roy Wesley Tromble “SEARCH AND LEARNING FOR THE LINEAR ORDERING PROBLEM WITH AN APPLICATION TO MACHINE TRANSLATION”. The Johns Hopkins University (2009). https://www.cs.jhu.edu/~jason/papers/tromble.thesis09.pdf

- Repositorio de la Junta de Andalucía. http://www.juntadeandalucia.es/averroes/centros-tic/14002996/helvia/aula/archivos/repositorio/250/271/html/economia/10/10-a.htm

- Rafael Martí, Gerhard Reinelt (2011) The Linear Ordering Problem. Editorial Springer.

- Departamento de Ciencias Computacionales de la Universidad de Columbia. RANK. http://rank.cs.columbia.edu/

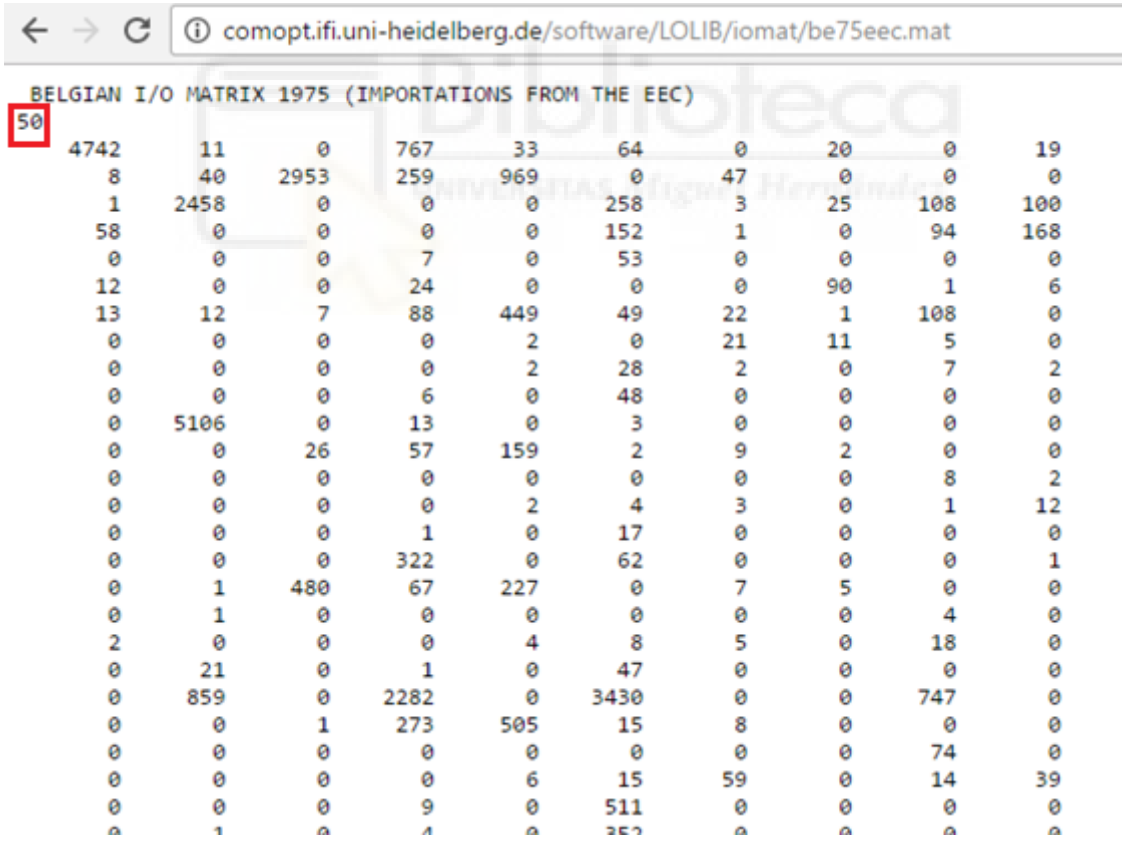
- Web IBM ILOG CPLEX https://www.ibm.com/support/knowledgecenter/es/SSSA5P_12.6.1/ilog.odms.cplex.help/CPLEX/UsrMan/topics/preface/preface_title_synopsis.html

11. Anexos

11.1 Anexo 1: Matrices de sectores bajadas de la web.

A continuación vamos a ver, como ejemplo, dos de las matrices de sectores recién bajadas de la web descrita en la conclusión del trabajo, donde veremos el formato en el que vienen y dónde indica el número de sectores que deben tener para poder trabajar con ellas.

1 – be75eec


comopt.ifi.uni-heidelberg.de/software/LOLIB/iomat/be75eec.mat

```
BELGIAN I/O MATRIX 1975 (IMPORTATIONS FROM THE EEC)
50
  4742     11      0    767     33     64      0     20      0     19
     8     40   2953    259    969      0     47      0      0      0
     1   2458      0      0      0    258      3     25    108    100
    58      0      0      0      0    152      1      0     94    168
     0      0      0      7      0     53      0      0      0      0
    12      0      0     24      0      0      0     90      1      6
    13     12      7     88    449     49     22      1    108      0
     0      0      0      0      2      0     21     11      5      0
     0      0      0      0      2     28      2      0      7      2
     0      0      0      6      0     48      0      0      0      0
     0   5106      0     13      0      3      0      0      0      0
     0      0     26     57    159      2      9      2      0      0
     0      0      0      0      0      0      0      0      8      2
     0      0      0      0      2      4      3      0      1     12
     0      0      0      1      0     17      0      0      0      0
     0      0      0    322      0     62      0      0      0      1
     0      1    480     67    227      0      7      5      0      0
     0      1      0      0      0      0      0      0      4      0
     2      0      0      0      4      8      5      0     18      0
     0     21      0      1      0     47      0      0      0      0
     0    859      0   2282      0   3430      0      0    747      0
     0      0      1    273    505     15      8      0      0      0
     0      0      0      0      0      0      0      0     74      0
     0      0      0      0      6     15     59      0     14     39
     0      0      0      9      0    511      0      0      0      0
```

En esta matriz hay 250 filas y 10 columnas, pero en la parte superior (con un recuadro rojo) vemos debe tener un formato de 50x50, por lo que utilizamos el script del Anexo 2 para convertirla en ese formato uniendo bloques de 5 filas en una sola.

2 – stabu1:

comopt.ifi.uni-heidelberg.de/software/LOLIB/iomat/stabu1.mat

```
INPUT-OUTPUT-TABELLE 1970 ZU AB-WERK-PREISEN (INLAENDISCHE PRODUKTION)
 60
 11954     0    19   604     0     6     2    23     0  1152
   944    26   155    31    76    28   148   655     0     3
     1     0   665   202     1    54     0   101     4    72
     3    70    47    23     3  4566   819    25    79     5
   263   607    69  1447   186    30  1092   138    36     0
   162    38    24   464     7    15   672     8     0    66
    39     0     0    26     1     0     0     0     0   178
     4     3     5     0     0     0     0    17     0     0
     0     0     7     7     0     0     0     1     0     0
     0     9     4     0     0     0     0     0     0     0
    16    16     3    55    22     1    79    10     6     0
     0     0     1    11     6     0    21     8     0     6
     0     0   365     5     0     0     0    15     0    16
    10    20    17     0     0     0     0    18     0     0
     0    42     0    20     0    16    16    16     0    16
    16    55     0    71    14    51     0     0    12     0
     0    24     5    78    86     0   195    20     0     0
    19    19    20     0     0     0    40     0     0     0
     0     0     0  1438   523     4  3122     3     0   188
   708    62     8     5     8    37     0    35     0     6
   644     0   206    43     3   500     1    23     7    64
     1   168     0    16    17     4     0     0    19    14
    30   163    55    28   128    10    54    62    37     0
    30     2    14    11    11     9   274     0     0    34
     0     0     0     9     2     0  1055     0   861     1
   709     0     0     0     0     0     0   606     1     4
    87     0    29     4     2     3    46     1     0     0
     0     5     0     1     0     0     0     0     7     3
     1   125    46    38    75     3    59     3     2     0
     1     5     3     0     4     1    39     0     0    15
     0     0     0    96     2     0    19     0     0    19
```

En esta matriz hay 360 filas y 10 columnas, pero en la parte superior (con un recuadro rojo) vemos debe tener un formato de 60x60, por lo que utilizamos el script del Anexo 2 para convertirla en ese formato uniendo bloques de 6 filas en una sola.

11.2 ANEXO 2: Script Unix Bash utilizado para formatear las matrices de sectores.

Script en Unix Bash para modificar los ficheros que vienen en formato 250 filas x 10 columnas, y convertirlos en una matriz 50x50 y para las ficheros de 360 filas x 10 columnas, para convertirlos en matrices de 60x60 (para ello cambiamos el módulo de 5 por módulo de 6).

```
bash
cont=1
while read line
do
  if (($cont % 5)); then
        line="$(echo $line | sed 's/\r/;/g')"
  fi
    echo -n $line >> FicheroModificado.csv
  cont=`expr $cont + 1`
done < /rutaDelFicheroOriginal/FicheroOriginal.csv
```

Lo que hace este script es leer el fichero línea por línea (mientras hayan líneas que leer) y va guardando en un contador la línea por la que va, sobre el que realiza la comprobación del módulo (de 5 o de 6, dependiendo de si ese fichero lo queremos convertir en 50x50 o 60x60), es decir, si el módulo de esa línea (resto de la división del número de línea contenido en el contador, entre 5 o 6, dependiendo del fichero que estemos leyendo, es distinto de 0) sustituimos el salto de línea final de esa línea por un punto y coma, sino, no se realiza nada.

Esto hace que al cambiar los saltos de línea por un punto y coma, nos una todas esas líneas en una sola y tengamos líneas de la longitud correcta para así poder analizar los ficheros con Escalamiento Multidimensional.

11.3 APÉNDICE 1: Resolución LOP con Excel

Representación en Excel de las distintas combinaciones posibles de la matriz de selecciones de fútbol, para el ejemplo de optimización presentado durante el trabajo.

	España	Brasil	Alemania	Bélgica	30+16+41+43+16+47=	193
España		30	16	41		
Brasil	63		43	16		
Alemania	51	70		47		
Bélgica	43	70	30			

	España	Brasil	Bélgica	Alemania	30+41+16+16+43+70=	176
España		30	41	16		
Brasil	63		16	43		
Bélgica	43	70		30		
Alemania	51	70	47			

	España	Alemania	Brasil	Bélgica	16+30+41+70+47+16=	220
España		16	30	41		
Alemania	51		70	47		
Brasil	63	43		16		
Bélgica	43	30	70			

	España	Alemania	Bélgica	Brasil	16+41+30+47+70+70=	274
España		16	41	30		
Alemania	51		47	70		
Bélgica	43	30		70		
Brasil	63	43	16			

	España	Bélgica	Brasil	Alemania	41+30+16+70+30+43=	230
España		41	30	16		
Bélgica	43		70	30		
Brasil	63	16		43		
Alemania	51	47	70			

	España	Bélgica	Alemania	Brasil		41+16+30+30+70+70=	257
España		41	16	30			
Bélgica	43		30	70			
Alemania	51	47		70			
Brasil	63	16	43				

	Brasil	España	Alemania	Bélgica		63+43+16+16+41+47=	226
Brasil		63	43	16			
España	30		16	41			
Alemania	70	51		47			
Bélgica	70	43	30				

	Brasil	España	Bélgica	Alemania		63+16+43+41+16+30=	209
Brasil		63	16	43			
España	30		41	16			
Bélgica	70	43		30			
Alemania	70	51	47				

	Brasil	Alemania	España	Bélgica		43+63+16+51+47+41=	261
Brasil		43	63	16			
Alemania	70		51	47			
España	30	16		41			
Bélgica	70	30	43				

	Brasil	Alemania	Bélgica	España		43+16+63+47+51+43=	263
Brasil		43	16	63			
Alemania	70		47	51			
Bélgica	70	30		43			
España	30	16	41				

	Brasil	Belgica	España	Alemania		
Brasil		16	63	43	16+63+43+43+30+16=	211
Belgica	70		43	30		
España	30	41		16		
Alemania	70	47	51			

	Brasil	Bélgica	Alemania	España		
Brasil		16	43	63	16+43+63+30+43+51=	246
Bélgica	70		30	43		
Alemania	70	47		51		
España	30	41	16			

	Alemania	España	Brasil	Bélgica		
Alemania		51	70	47	51+70+47+30+41+16=	255
España	16		30	41		
Brasil	43	63		16		
Bélgica	30	43	70			

	Alemania	España	Bélgica	Brasil		
Alemania		51	47	70	51+47+70+41+30+70=	309
España	16		41	30		
Bélgica	30	43		70		
Brasil	43	63	16			

	Alemania	Brasil	España	Bélgica		
Alemania		70	51	47	70+51+47+63+16+41=	288
Brasil	43		63	16		
España	16	30		41		
Bélgica	30	70	43			

	Alemania	Brasil	Bélgica	España		
Alemania		70	47	51	70+47+51+16+63+43=	290
Brasil	43		16	63		
Bélgica	30	70		43		
España	16	30	41			

	Alemania	Bélgica	España	Brasil		
Alemania		47	51	70	47+51+70+43+70+30=	311
Bélgica	30		43	70		
España	16	41		30		
Brasil	43	16	63			

	Alemania	Bélgica	Brasil	España		
Alemania		47	70	51	47+70+51+70+43+63=	344
Bélgica	30		70	43		
Brasil	43	16		63		
España	16	41	30			

	Bélgica	España	Brasil	Alemania		
Bélgica		43	70	30	43+70+30+30+16+43=	232
España	41		30	16		
Brasil	16	63		43		
Alemania	47	51	70			

	Bélgica	España	Alemania	Brasil		
Bélgica		43	30	70	43+30+70+16+30+70=	259
España	41		16	30		
Alemania	47	51		70		
Brasil	16	63	43			

	Bélgica	Brasil	España	Alemania		
Bélgica		70	43	30	70+43+30+63+43+16=	265
Brasil	16		63	43		
España	41	30		16		
Alemania	47	70	51			

	Bélgica	Brasil	Alemania	España		
Bélgica		70	30	43	70+30+43+43+63+51=	300
Brasil	16		43	63		
Alemania	47	70		51		
España	41	30	16			

	Bélgica	Alemania	España	Brasil		
Bélgica		30	43	70	30+43+70+51+70+30=	294
Alemania	47		51	70		
España	41	16		30		
Brasil	16	43	63			

	Bélgica	Alemania	Brasil	España		
Bélgica		30	70	43	30+70+43+70+51+63=	327
Alemania	47		70	51		
Brasil	16	43		63		
España	41	16	30			

11.4 APÉNDICE 2: Resolución de LOP con CPLEX

```
#include <ilcplex/cplex.h>
#include <ctype.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>

/*------------------------------------------------------------------------*/

CPXENVptr     env = NULL;
CPXLPptr      lp = NULL;

/*------------------------------------------------------*/

main (int argc, char* argv[])
{

/***************** Definición de variables ************************************/


int n;
int i,j,k;
int **d;
double iobjval1;
double *s=NULL;
double t1,t2;
```

```
FILE *puntRED;
FILE *puntLEC;
FILE *puntRES;
char *ficheroL=argv[1];
puntRED=fopen("lop.lp","w");
puntLEC=fopen(ficheroL,"r");
puntRES=fopen("lopres.txt","a+");



fscanf(puntLEC,"%d",&n);
d=matpuntero(n+1,n+1);


for (i=1; i<n+1; i++)
{
   for (j=1; j<n+1; j++)
   {
    fscanf(puntLEC,"%d",&d[i][j]);
   }
}

for (i=1; i<n+1; i++)
{
for (j=1; j<n+1; j++)
{
printf("%d    ",d[i][j]);
}
printf("\n");
```

```
}



int status = 0;
/********** ABRIMOS CPLEX *********************/
env = CPXopenCPLEX (&status);
status = CPXsetintparam (env, CPX_PARAM_SCRIND, CPX_ON);
lp= CPXcreateprob (env, &status, "prob");



/*********************************************/


printf("Escribiendo el problema lp.\n");
fprintf(puntRED,"Maximize\n");
fprintf (puntRED,"obj: ");

for (i=1; i<n+1; i++)
{
for (j=1; j<n+1; j++)
{
if ( i!= j)
{
fprintf (puntRED, " + %d xi%dv%d ",d[i][j],i,j);
}
}
}

fprintf (puntRED, "\n ");
```

```
fprintf(puntRED,"Subject To\n");

for (i=1; i<n+1; i++)
{
for (j=i+1; j<n+1; j++)
{

fprintf (puntRED, " + xi%dv%d +  xi%dv%d =1\n",i,j,j,i);

}}


for (i=1; i<n+1; i++)
{
for (j=i+1; j<n+1; j++)
{
for (k=i+1; k<n+1; k++)
{
 if (j != k)
 {
fprintf (puntRED, " + xi%dv%d +  xi%dv%d  +  xi%dv%d <=2\n",i,j,j,k,k,i);
}
}
}
}

fprintf(puntRED,"Binaries\n");
for (i=1; i<n+1; i++)
{
for (j=1; j<n+1; j++)
{
if ( i!= j)
```

```
{
fprintf (puntRED, " xi%dv%d\n",i,j);
}
}
}
fprintf (puntRED,"End\n");
fclose(puntRED);


status = CPXreadcopyprob(env, lp,"lop.lp", NULL);
if (status!=0) printf ("\n Un error se ha producido al leer el fichero ");



status=CPXmipopt(env,lp);


return;
}
```


Este script está escrito en lenguaje C y realiza lo siguiente.

Cargamos las siguientes librerias:

- Ctype.h = Ctype.h es un archivo de cabecera de la biblioteca estándar del lenguaje de programación C diseñado para operaciones básicas con caracteres. Contiene los prototipos de las funciones y macros para clasificar caracteres.

- Stdio.h = Stdio.h, que significa "standard input-output header" (cabecera estándar E/S), es el archivo de cabecera que contiene las definiciones de las macros, las constantes, las declaraciones de funciones de

-

-

- la biblioteca estándar del lenguaje de programación C para hacer operaciones estándar de entrada y salida, así como la definición de tipos necesarias para dichas operaciones.

- String.h = String.h es una librería que define un tipo de variable, una macro y varias funciones para la manipulación de caracteres.

- Stdlib.h = Stdlib.h (std-lib: standard library o biblioteca estándar). Es el archivo de cabecera de la biblioteca estándar de propósito general del lenguaje de programación C. Contiene los prototipos de funciones de C para gestión de memoria dinámica, control de procesos y otras.

- Time.h = Time.h es una librería que define cuatro tipos de variables, dos macro y varias funciones para manipular fechas y horas.

- Math.h = Math.h es un archivo de cabecera de la biblioteca estándar del lenguaje de programación C diseñado para operaciones matemáticas básicas. Muchas de sus funciones incluyen el uso de números en coma flotante.

El script lee la matriz en .txt dónde el primer número del fichero debe ser el tamaño de la matriz cuadrada, para que el script pueda crearla con los números siguientes a este, y posteriormente ya puede crear las restricciones pertinentes y resolver la función objetivo de forma iterative hasta encontrar el valor que optimice la suma de los valores de la matriz que quedan por encima de la diagonal.

11.5 APÉNDICE 3: Resolución de MS con R

```
# instalamos los paquetes
install.packages(c("lattice","permute","vegan"  ,"ecodist", "labdsv", "ape", "ade4",
"smacof"))

# cargamos los paquetes
library(lattice)
library(permute)
library(vegan)
library(ecodist)
library(labdsv)
library(ape)
library(ade4)
library(smacof)

#Iniciamos cronometro
t <- proc.time()

# convertimos eurodist en una matriz
distancias = as.matrix(eurodist)

#dim(distancias)

# vemos los primeros 5 elementos de la matriz
#distancias[1:5,1:5]
```

```
# Comenzamos el Escalamiento Multidimensional
cmdscale(distancias, k = 2, eig = FALSE, add = FALSE, x.ret = FALSE)

mds1 = cmdscale(eurodist, k=2)

# Lo pintamos
plot(mds1[,1], mds1[,2], type = "n", xlab = "", ylab = "", axes = FALSE,
    main = "cmdscale (stats)")
text(mds1[,1], mds1[,2], labels(eurodist), cex=0.9)

#Paramos cronometro
proc.time()-t
```


Este script hace lo siguiente, paso a paso:

Instalación y carga de paquetes

Excepto para cmdscale, el resto de las funciones que utilizaremos (o que podemos utilizar) no vienen con la distribución por defecto de R, por lo que tendremos que instalar sus paquetes correspondientes y posteriormente cargarlos para poder utilizarlos.

Datos eurodist

Utilizaremos el conjunto de datos eurodist que da las distancias de carretera (en km) entre 21 ciudades de Europa. Vemos que eurodist ya es un objeto de clase "dist" (distancia matricial).

El objetivo es aplicar métricas MDS para obtener una representación visual de las distancias entre ciudades europeas.

MDS con cmdscale

La función más popular para realizar una escala clásica es cmdscale (que viene con la distribución predeterminada de R).

Como hemos visto antes en la representación gráfica de este MDS, el gráfico obtenido nos permite representar las distancias entre ciudades en un espacio bidimensional. Sin embargo, la representación no es idéntica a un mapa geográfico de Europa: Atenas está en el norte mientras que Estocolmo está en el sur. Esta "anomalía" refleja el hecho de que la representación no es única; Si queríamos obtener una representación geográfica más precisa, tendríamos que invertir el eje vertical.