Universidad Miguel Hernández Elche

Facultad de Ciencias Sociales y Jurídicas

Subsetting the Linear Ordering Problem

Tutora: Mercedes Landete Ruiz
Cotutor: Juan Francisco Monge

Alumno: Rafael Antón Moya

10 de junio de 2024
Curso 2023-2024

Índice

1. Introducción

El objetivo de este documento es estudiar una modificación propuesta para el problema de ordenación lineal, en adelante LOP, conocido así de forma general en el campo de optimización por sus siglas en inglés Linear Ordering Problem. Concretamente se trata de un problema de optimización combinatoria estudiado por primera vez en 1958 por Chenery y Watanabe. Desde su aparición, son muchos los campos de estudio donde se ha aplicado, entre ellos arqueología (Glover et al., 1972), economía (Leontief, 2008), teoría de grafos (Charon and Hudry, 2007), traducción (Tromble and Eisner, 2009) y psicología matemática (Kemeny, 1959). Además ha visto una gran variedad de métodos para su solución, inicialmente se comenzó resolviendo mediante técnicas exactas, sin embargo, la cantidad de datos que soportaba en ciertas investigaciones era de tal tamaño que provocaba un tiempo computacional que imposibilitaba la consecución de una respuesta. Este matiz se solventó con el uso de técnicas heurísticas, y más adelante aplicando los avances producidos en la optimización meteheurística. Hoy en día el Algoritmo Memético (Memetic Algorithm - MA) y la Búsqueda Local Iterada (Iterated Local Search - ILS) propuestos por Schiavinotto y Stutzle en 2004, son los algoritmos que representan la forma más vanguardista de solucionar el LOP.

Los problemas de optimización combinatoria son ciertamente curiosos, pues como se ha visto para la mayoría de ellos, la complejidad de los mismos no sólo depende del tamaño de la instancia, sino de una serie de parámetros adicionales que son generalmente desconocidos. La comunidad científica ha tratado durante años de comprender cuáles son estas características con el fin de guiar los algoritmos para garantizar su robustez. En 2004 Schiavinotto y Stutzle dieron con ciertas propiedades que caracterizaban la dureza de las instancias del LOP, los autores definieron: disparidad, coeficiente de variación, asimetría y correlación de distancia de aptitud.

La modificación del LOP que se pretende tratar en el presente documento, busca estudiar una posible mejora del algoritmo mediante la agrupación de los individuos, de forma que se comparen los sujetos más parecidos entre ellos y no exista iteración entre los dispares, de esta forma obtendríamos grupos internamente homogéneos y externamente heterogéneos, lo cual garantizaría la calidad de la respuesta ofrecida.

El documento que se presenta es el fruto de la colaboración con el Departamento de Estadística de la Universidad Miguel Hernández justificada a través de la Beca de Colaboración, con código 998142, ofrecida por el Ministerio de Educación. Como resultado de la misma abordaremos soluciones del LOP en aplicaciones reales en el campo de la educación, modificaciones pertinentes en los datos del problema para mejorar la exactitud de la resolución, la modificación del modelo antes mencionada y una propuesta de aplicación del modelo en el sistema de reconocimiento académico concedido en cada curso por la Universidad Miguel Hernández de Elche al mejor alumno de la promoción.

2. Linear Ordering Problem

2.1. Introducción

Como hemos visto en el primer punto, el LOP es un problema clásico de combinatoria que busca encontrar el orden que mejor clasifica a los individuos de un conjunto. Supongamos un conjunto de elementos a ordenar $V = \{1,...,n\}$ sobre el cual aplicamos una comparación por parejas del tipo uno a uno para obtener una matriz cuadrada D de orden n. De este modo, la fila i de la matriz D, contendrá la información respectiva a comparar el elemento i con el resto de elementos del conjunto inicial. Así pues, cada valor de dentro de la matriz D, tal que d_{ij} será la unidad de bondad que resumirá en qué medida va antes el elemento i que j.

El LOP se nutre de la matriz cuadrada de preferencias D, donde cada elemento de esta matriz se puede ver como la *distancia* de i a j. Cuánto más grande sea la distancia del valor d_{ij}, mayor es la prueba de que i debe preceder a j. El número de órdenes que podemos crear de una matriz de tamaño n, es $n!$, es decir, si tenemos un conjunto con 5 individuos, podremos establecer 120 órdenes distintos. Se muestra de una forma intuitiva y gráfica en la $Tabla\ 2.2.1$, donde se pretende ilustrar el siguiente cálculo: $5 * 4 * 3 * 2 * 1 = 120$

Tabla 2.2.1: Demostración gráfica del número de órdenes posibles

1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5

2.2. Matriz de preferencias

El cálculo de la matriz de preferencias D y de los individuos d_{ij} consiste en realizar un conteo con el número de veces que el individuo i es mejor que el individuo j. Aunque se trate de aparentemente una tarea sencilla, esta se puede complicar cuando la cantidad de información de cada conjunto de individuos no es del mismo tamaño. En esos casos tendremos que valorar la posibilidad de realizar modificaciones en los datos que veremos de forma evidente en los ejemplos que presentamos a continuación, donde primero veremos el origen de los datos y cómo son tratados para pasar a sus respectivas matrices de preferencias.

Tabla 2.2.1: Lista de votaciones por jueces

Juez 1:	a	b	e	d	c
Juez 2:	c	a	e	b	d
Juez 3:	b	a	e	c	d
Juez 4:	c	d	b	e	a
Juez 5:	a	d	c	e	b

Tabla 2.2.2: Matriz de preferencias Tabla 2.2.1

	Juez 1	Juez 2	Juez 3	Juez 4	Juez 5
Juez 1	0	3	3	4	4
Juez 2	2	0	2	3	3
Juez 3	2	3	0	3	3
Juez 4	1	2	2	0	2
Juez 5	1	2	2	3	0

En la $Tabla\ 2.2.1$, vemos el orden establecido por cada uno de los cinco jueces para los participantes a, b, c, d, y e. En este modelo se clasifican de mejor a peor cada uno de ellos, de forma que el $Juez\ 1$ considera que el individuo a es el mejor y el c el peor, y así consecuentemente para el resto de los jueces. La forma de reflejar este orden en una matriz de preferencias, como se observa en la $Tabla\ 2.2.2$. Como se puede intuir por todo lo explicado hasta este punto, cualquier posición $i\ j$ de la matriz representa cuántas veces es mejor i que j, en el ejemplo, la posición $a\ b$ toma el valor 3, representando que el individuo a es mejor que el b en tres ocasiones. Respecto al problema planteado debemos tener una serie de cuestiones en mente para la correcta resolución del mismo:

1. El máximo valor que puede tomar cualquier posición $i\ j$ de la matriz es 5, en cuyo caso supondría que el individuo i se posicionaría por delante de su correspondiente en los 5 juicios. Por otro lado, el mínimo valor de cualquier individuo es el 0, que también supondría un consenso total, siempre y cuando se clasificara por detrás en los 5 juicios. De forma que si llamamos D a la matriz de la $Tabla\ 2.2.2$, se cumpliría que:
$max\{d_{ij}\} <= 5$; $min\{d_{ij}\} => 0\ \forall\ i, j \in \{1, 2, 3, 4, 5\}$

2. Como la lista de participantes es cerrada y en el caso son 5 los individuos, el valor de la posición ij más el valor de la posición ji debe ser igual a 5. Utilizando la nomenclatura anterior, se cumple que:
$d_{ij} + d_{ji} = 5\ \forall\ i, j \in \{1, 2, 3, 4, 5\}|\ i \neq j$

3. El elemento de la matriz correspondiente a la diagonal de la misma será siempre 0, ya que en ningún momento podremos comparar la posición de un individuo consigo mismo. Utilizando la nomenclatura anterior, se cumple que:
$d_{ij} = 0\ \forall\ i, j \in \{1, 2, 3, 4, 5\}|\ i = j$

Tabla 2.2.3: Nota media en matemáticas por ID de colegio y país

ID	País	Nota
001	España	9,30
002	España	7,50
003	España	8,00
004	España	9,00
005	España	5,50
001	Francia	7,10
002	Francia	5,40
003	Francia	6,00
004	Francia	8,20
001	Alemania	7,00
002	Alemania	6,20
003	Alemania	5,30
004	Alemania	7,20
001	Brasil	5,90
002	Brasil	7,90
003	Brasil	7,30
004	Brasil	4,50
005	Brasil	6,90

Tabla 2.2.4: Matriz de preferencias Tabla 2.2.3

	España	Francia	Alemania	Brasil
España	0	15	17	20
Francia	5	0	9	11
Alemania	3	7	0	9
Brasil	5	9	11	0

Tal y como hemos comentado anteriormente, en este segundo ejemplo vemos la realización de una matriz de preferencias a partir de unos datos originales donde las comparaciones no tienen el mismo tamaño, $Tabla\ 2.2.3$. En el caso de España y Brasil nos encontramos con 5 individuos mientras que en Francia y Alemania tenemos tan solo 4. En este caso, como vemos en la matriz de preferencias, la suma de los valores $d_{ij} + d_{ji}$ sigue siendo un valor que podemos calcular, y que tiene una lógica. Para explicarla, definamos $V_i \; \forall \; i \in \{España, Francia, Alemania, Brasil\}$, que representará el número de individuos de cada elemento, así pues, $V_{España} = V_{Brasil} = 5$ y $V_{Francia} = V_{Alemania} = 4$ .Una vez hemos definido estos conceptos, ya podemos asegurar que:

- $d_{ij} + d_{ji} = V_i * V_j \forall i, j \in \{1, 2, ..., I\} | i \neq j$ $\quad (Propiedad\ 2.2.1)$

Como vemos con la fórmula establecida, para la comparación $España - Alemania$, tendríamos $15 + 5 = 5 * 4 = 20$. Además, al igual que sucedía en el otro ejemplo, se cumplen las dos propiedades mencionadas, ahora con el matiz de que el tamaño es distinto para la segunda de ellas, de esta forma definiríamos:

- $d_{ij} = 0 \; \forall \; i, j \in \{1, 2, ..., I\} | \; i = j$ $\quad (Propiedad\ 2.2.2)$

- $max\{d_{ij}\} <= V_i {*} V_j \; ; min d_{ij} => 0$ $\quad (Propiedad\ 2.2.3)$

Como se ha podido intuir en el procedimiento de la creación de la matriz de preferencias, existe una pequeña desventaja en las comparaciones cuando los tamaños son distintos. Al comparar $España - Brasil$, obtenemos que $España$ es en 20 ocasiones mayor que $Brasil$, por el contrario, cuando lo comparamos con $Alemania$, este número se reduce a 17. Sin tener en cuenta el tamaño de los grupos de países, podríamos pensar que $España$ tiene más superioridad cuando compite contra $Brasil$, que cuando lo hace contra $Alemania$, sin embargo, es tan solo una cuestión de totales, pues si lo traducimos a porcentajes, obtendríamos el siguiente resultado:

Tabla 2.2.5: Matriz de preferencias transformada

	España	Francia	Alemania	Brasil
España	0	0.75	0.85	0.8
Francia	0.25	0	0.56	0.55
Alemania	0.15	0.44	0	0.45
Brasil	0.2	0.45	0.55	0

Como se refleja claramente en el ejemplo de la $Tabla\ 2.2.5$, realmente la mayor superioridad que encuentra $España$ es con Alemania, siendo mejor en el 85 % de las ocasiones. La normalización de los datos de la matriz de preferencias es la que tenemos que realizar si queremos garantizar una buena solución en el LOP, como hemos visto, este simple cambio ha transformado la manera de ver la matriz de preferencias y somos capaces de ofrecer una conclusión mucho más acertada. Si llamamos T a la matriz que se refleja en la $Tabla\ 2.2.5$,la transformación que se aplica a la $Tabla\ 2.2.4$ para obtener dicha matriz es:

- $t_{ij} = \frac{d_{ij}}{V_i * V_j}\ \forall\ i, j \in \{1, 2, ..., I\}|\ i \neq j$

2.3. Explicación del problema

Por lo que hemos ido apuntando hasta aquí, sabemos que el LOP es un problema que trata de descifrar el mejor orden para clasificar un número indefinido de individuos. Además, gracias a los puntos tratados tenemos algo más de información; sabemos que cuánto mayor es el número de individuos, mayor es la complejidad del sistema a resolver y por ende, mayor es la capacidad de procesamiento que requerimos en nuestros sistemas informáticos. Sabemos también, que para hallar una solución de este problema necesitamos una matriz de preferencias que ya hemos aprendido a calcular en el punto *2.2 Matriz de preferencias.*

Si la matriz de preferencias está reflejando para cada par $i - j$, el número de ocasiones que el individuo i es mejor que el individuo j, entonces, en el orden correcto, el primer individuo deberá tener una suma total mayor que la del resto. Para entender este concepto, utilizaremos la $Tabla\ 2.2.5$ sobre la cual aplicaremos permutaciones en las filas para mostrar distintos resultados.

Tabla 2.3.1: Matriz de preferencias Tabla 2.5.5 permutada (1)

	España	Francia	Alemania	Brasil
España	0	0.8	0.75	0.85
Francia	0.2	0	0.45	0.55
Alemania	0.25	0.55	0	0.56
Brasil	0.15	0.45	0.44	0

Tabla 2.3.2: Matriz de preferencias Tabla 2.5.5 permutada (2)

	España	Francia	Alemania	Brasil
España	0	0.75	0.8	0.85
Francia	0.25	0	0.55	0.56
Alemania	0.2	0.45	0	0.55
Brasil	0.15	0.44	0.45	0

Si realizamos la suma por filas, de la matriz de preferencias, estaremos obteniendo el total de veces que un individuo es mejor que el resto, así pues, si llamamos P a la matriz permutada representada en la $Tabla\ 2.3.1$, y calculamos la siguiente expresión:

- $\sum_{j=España}^{Alemania} p_{ij} \quad \forall i \in \{España, Brasil, Francia, Alemania\}$

Obtenemos los siguientes resultados para cada uno de los cuatro países:

- $\sum_{j=España}^{Alemania} p_{España\, j} = 2{,}4$
- $\sum_{j=España}^{Alemania} p_{Brasil\, j} = 1{,}2$
- $\sum_{j=España}^{Alemania} p_{Francia\, j} = 1{,}36$
- $\sum_{j=España}^{Alemania} p_{Alemania\, j} = 1{,}04$

Como vemos en estos resultados, el primer individuo siempre será el que obtenga, como hemos dicho, la suma más grande de todas las comparaciones, sin embargo, no sucederá lo mismo si seguimos observando el resto de individuos de la matriz. Este primer caso resulta de tal manera, porque estamos utilizando para el LOP todos los individuos de la fila, por el contrario, en cuanto bajamos de puestos, se reduce el número de individuos posibles para sumar, por lo que la solución del sistema requiere estrictamente la aplicación del algoritmo que estudiamos en este trabajo, y que básicamente optimiza los espacios restantes para obtener el orden óptimo.

El LOP como ya sabemos se trata de un problema clásico de combinatoria, por lo que tenemos que ser capaces de resolver lo que se ha plasmado, en un modelo de programación matemática. Si nos fijamos en las $Tablas\ 2.2.5,\ 2.3.1,\ y\ 2.3.2.$, encontraremos que podemos estudiar la mejoría de la solución (el orden) observando los individuos que se encuentran por encima de la diagonal. Si sumamos los elementos por encima de la matriz de las tablas obtenemos lo siguiente: $Tabla\ 2.2.5$

- $Tabla\ 2.2.5 = 3{,}96$
- $Tabla\ 2.3.1 = 3{,}96$
- $Tabla\ 2.3.2 = 4{,}06$

El problema del LOP es justamente resolver la acción que acabamos de explicar, partimos de una matriz de preferencias cual sea, en nuestro caso la $Tabla\ 2.2.5$, y realizamos una serie de permutaciones que nos hacen llegar a la matriz cuya diagonal superior es la máxima posible ($Tabla\ 2.3.2$).

2.4. Modelo matemático

Tras lo visto al final de la sección anterior, sabemos que el modelo matemático que utilizamos para resolver el problema del LOP, y conseguir el **máximo** valor posible en la diagonal superior de la matriz de preferencias tiene como objetivo evidente maximizar, y se plasma de la siguiente forma:

$$\begin{array}{llrr}
\text{máx} & \displaystyle\sum_{r,s\in P:r\neq s} m_{rs}x_{rs} & & \\
s.a & x_{rs}+x_{sr}=1 & r,s\in P: r<s & (1)\\
 & x_{rs}+x_{st}+x_{tr}\leq 1 & r,s,t\in P: r,s,t \text{ pwd} & (2)\\
 & x_{rs}\in\{0,1\} & r,s\in P: r\neq s & (3)
\end{array}$$

En este modelo tenemos en cuenta que M es el indicativo para la matriz de preferencias correspondiente, por lo que m_{rs}, es el individuo de la matriz de preferencias que hemos explicado en las anteriores secciones, y que refleja el número de veces que el individuo r está por delante del s. Por otro lado, la variable x_{rs}, es una variable binaria, que toma el valor de 1, si afirmamos que el individuo r está por delante en el ranking que el s. De esta forma, la restricción (1), establece que, o bien r es preferible a s, o viceversa, pero en ningún momento podrán suceder las dos. La condición (2), indica el orden lógico cuando tenemos tres individuos, donde si r es mejor que s, y esta mejor que t, la suma siempre será menor o igual a dos. Para entender esta condición pondremos un ejemplo de un ranking ordenado y mostraremos como se cumple la condición:

Tabla 2.4.1: Ranking

Puesto	Orden
1er	A
2do	B
3er	C
4to	D

- $x_{AB}+x_{BC}+x_{CA}\leq 2$
- $x_{AC}+x_{CD}+x_{DA}\leq 2$
- $x_{AD}+x_{DB}+x_{BA}\leq 2$
- $x_{BD}+x_{DC}+x_{CB}\leq 2$

Para este orden, el valor de las variables que se muestran acompañando a la *Tabla* 2.4.1 debería ser el siguiente:

- $x_{AB}=1$
- $x_{BC}=1$
- $x_{CA}=0$
- $x_{AC}=1$
- $x_{CD}=1$
- $x_{DA}=0$
- $x_{AD}=1$
- $x_{DB}=0$
- $x_{BA}=0$
- $x_{BD}=1$
- $x_{DC}=0$
- $x_{CB}=0$

dCon el ejemplo propuesto y el valor de las variables dado, se explica de forma clara la sencillez y utilidad de la restricción, en este ejemplo, se puede ver también como actúa la restricción (1), ya que por ejemplo, $x_{AB}+x_{BA}=1$. La última restricción que queda por resolver es (3), que más que restricción es una indicación de que la variable x es una variable binaria comprendida entre 1 y 0.

2.5. Solución del LOP

A pesar de que la resolución del LOP la realizamos con un programa informático, sin el cual sería imposible estudiar las matrices que se verán más adelante, debemos destacar la correcta interpretación de los resultados para poder obtener soluciones.

El método que emplearemos, será la extracción del programa informático, en la mayoría de ocasiones Lingo, aunque también utilizaremos Python, y su importación en Excel para ser tratados. A continuación se detallarán los pasos que dan lugar a la resolución de cualquier matriz utilizando Lingo y el algoritmo del LOP.

Tabla 2.5.1: Matriz de preferencias a resolver

	A	B	C	D
A	0	13223	35725	142660
B	12233	0	30506	121701
C	223	426	0	91819
D	1648	2471	83532	0

```
MODEL:

SETS:

      EJE_I/EI1..EI4/;
      EJE_J/EJ1..EJ4/;

      MATRIZ(EJE_I,EJE_J): MAT,X,Z;

ENDSETS
DATA:

      MAT= 0, 13223,35725,142660,
12233,0,30506,121701,
223,426,0,91819,
1648,2471,83532,0;

ENDDATA

@FOR(MATRIZ(I,J):@BIN(X(I,J)));

[OBJETIVO] MAX=@SUM(MATRIZ(I,J):MAT(I,J)*X(I,J));

@FOR (EJE_I(I):
      @FOR(EJE_J(J)| J#GT#I:
            X(I,J)+X(J,I)=1));

@FOR(EJE_I(I):
      @FOR(EJE_J(J)| J#NE#I:
            @FOR(EJE_J(T)| T#NE#J #AND# T#NE#I:
                  X(I,J)+X(J,T)+X(T,I)<=2)));
END
```

```
X( EI1, EJ1)        0.000000        0.000000
X( EI1, EJ2)        1.000000       -13223.00
X( EI1, EJ3)        1.000000       -35725.00
X( EI1, EJ4)        1.000000       -142660.0
X( EI2, EJ1)        0.000000       -12233.00
X( EI2, EJ2)        0.000000        0.000000
X( EI2, EJ3)        1.000000       -30506.00
X( EI2, EJ4)        1.000000       -121701.0
X( EI3, EJ1)        0.000000       -223.0000
X( EI3, EJ2)        0.000000       -426.0000
X( EI3, EJ3)        0.000000        0.000000
X( EI3, EJ4)        1.000000       -91819.00
X( EI4, EJ1)        0.000000       -1648.000
X( EI4, EJ2)        0.000000       -2471.000
X( EI4, EJ3)        0.000000       -83532.00
X( EI4, EJ4)        0.000000        0.000000
```

Partiendo de la *Tabla* 2. 5. 1, lo primero que hacemos es volcar la información en Lingo, el programa que utilizaremos para resolver el modelo, aunque como hemos comentado puede ser Python o cualquier otro que genere confianza en lo que hacemos. Una vez tenemos la matriz expresamos el modelo mediante la programación correspondiente, por la inconveniencia de su explicación en este trabajo no vamos a entrar en detalle en ningún punto sobre la programación que utilizamos ni el funcionamiento de la misma, pero sí vamos a adjuntarla en el Anexo. Cuando calculamos el resultado, la solución del modelo será un único valor que reflejará el máximo encontrado, sin embargo, la que a nosotros realmente nos interesa es la combinación de las variables que dan lugar a ese máximo.

En este caso no utilizamos Excel para ahorrar espacio en el documento y ser más concisos y prácticos, aunque en soluciones con más datos, debemos sin duda recurrir a este software para hallar soluciones de forma rápida. Como vemos, obtenemos 1 y 0, pero como tras los apartados anteriores sabemos lo que significa ese resultado, somos capaces de interpretarlo. Como vemos, x_{12}, x_{13}, x_{14}, x_{23}, x_{24}, $x_{34} = 1$, lo que supone el siguiente resultado:

Tabla 2.5.2: Resultado

Puesto	Orden
1ro	1
2do	2
3er	3
4to	4

Hasta aquí se han dado todas las indicaciones para poder obtener la solución del LOP de cualquier conjunto de datos, siendo ahora capaces de partir de una base de datos cruda y paso a paso obtener los elementos necesarios para resolver el orden óptimo. De este punto en adelante, se presentará la realización completa del LOP sobre los datos PISA, para más adelante explicar las mejoras posibles para el modelo estudiado hasta ahora.

3. Resolución LOP informe PISA

3.1. Origen del problema

La idea de resolver el problema del LOP con los datos PISA es el punto de partida de la investigación llevada a cabo, y que se comenta más atrás en este documento. Como paso hacia el completo entendimiento del funcionamiento del problema del LOP, se planteó la resolución de la base de datos del informe PISA, ya que esta había sido utilizada por la tutora Mercedes Landete y el cotutor Juan Francisco Monge en el artículo publicado en 2019 titulado "A linear ordering problem of sets", añadido en la Bibliografía de este trabajo. Cabe destacar que en esta publicación solo se puede ver el resultado de la aplicación del LOP a dicha base de datos, por lo que ha servido como guía de aprendizaje. Para ofrecer mucha más amplitud de conocimiento en cuanto a la resolución ante este tipo de situaciones, en los puntos siguientes se detalle con exactitud todos los pasos seguidos.

3.2. Informe PISA

En primer lugar, describiremos de qué trata el informe PISA, por sus siglas en inglés (Programme for International Student Assessment), y en castellano Programa para la Evaluación Internacional de los Estudiantes. En él se refleja el estudio elaborado por la OCDE a nivel mundial, que mide el rendimiento académico de los estudiantes de 15 años en tres campos distintos: matemáticas, ciencias y lectura. Los exámenes se promueven en cada país desde el ámbito institucional, es decir, es el estado el encargado de su realización, el cual prara garantizar una muestra representativa de los datos requiere la realización de los exámenes de 4.500 a 10.000 estudiantes por cada país, aunque en algunos casos como México o Estados Unidos la suma aumenta hasta más de 30.000 estudiantes.

El archivo de datos que hemos utilizado para la resolución del problema es del tipo txt dividido en 6 columnas:

- CNT: Indica el país del que se recoge el estudio. Es una variable de tipo cualitativa nominal.
- UNIT: Recoge el número de escuela que se está evaluando entre todos los países. Es una variable de tipo cualitativa ordinal.
- SCHOOLID: Recoge el número de escuela que se está evaluando del mismo país, la cuenta se inicia desde uno para cada nuevo país. Es una variable de tipo cualitativa ordinal.
- PVMATH: Recoge la puntuación obtenida de esa escuela en el examen de matemáticas. Es una variable de tipo cuantitativa continua.
- PVREAD: Recoge la puntuación obtenida de esa escuela en el examen de lectura. Es una variable de tipo cuantitativa continua.
- PVSCIE: Recoge la puntuación obtenida de esa escuela en el examen de ciencas. Es una variable de tipo cuantitativa continua.

A continuación se representa en la *Tabla* 3.2.1 una visualización que busca reflejar el formato de este archivo, como se observa y teniendo en cuenta la descripción anterior de las variables, el último valor de la columna UNIT, que corresponde a USA (por orden alfabético en la variable CNT), evidencia el número total de escuelas utilizadas para el estudio 13.494. Si estimamos una media de 80 estudiantes por escuela, tenemos un total de 1.079.520 estudiantes que han realizado los exámenes.

Tabla 3.2.1: Matriz de preferencias a resolver

CNT	UNIT	SCHOOLID	PVMATH	PVREAD	PVSCIE
AUS	1	1	559.09	604.22	630.65
AUS	2	2	483.7	514.43	501.76
...	...	...	...	...	...
ESP	4964	771	507.73	518.67	508.37
ESP	4965	772	494.15	737.9	518.43
...	...	...	...	...	...
USA	13494	162	342.38	291.83	331.01

3.3. Objetivo del problema

Una vez sabemos cuál es la base de datos que vamos a tratar, las variables que la componen y del tipo que son cada una, solo queda saber qué objetivo queremos conseguir para poder establecer los pasos a seguir. En este caso el objetivo de la aplicación del LOP conosiste en conseguir el orden de países ordenado de mejor a peor según la puntuación en el examen de matemáticas para mostrar las diferencias respecto a otro tipo de criterios de ordenación.

3.4. Transformación base de datos a matriz de preferencias

Para aplicar todas las transformaciones requeridas a los datos que faciliten la resolución del problema, importaremos la base de datos txt en nuestro software de computación estadística, R-Studio, en adelante R. Antes de seguir, cabe destacar que toda la programación relacionada con lo que se comenta en este apartado queda reflejada en el anexo, pudiendo ser consultada para un mejor entendimiento. En R eliminaremos las columnas que no necesitamos, estas son $UNIT$, $SCHOOLID$, $PVREAD$, Y $PVSCIE$, por lo que quedarán únicamente CNT, y $PVMATH$.

Para obtener la matriz de preferencias, separaremos la base de datos original en tantas tablas como países existan en la base de datos, a continuación crearemos un bucle que pueda contar cuántas veces un país es mejor que otro, comparando entre sí individuo a individuo de cada uno de los 39 países .

En las siguientes tablas se trata de simular el paso desde que entra en el software hasta que sale en forma de matriz de preferencias. En la $Tabla$ 3.4.1, la primera columna es una representación de la visualización en el software, y que es indicativa del tamaño de cada tabla, es por ello que esta columna no cuenta con encabezado. Con las $Tablas$ 3.4.1 y 3.4.2 se busca representar la separación de los países en distintas tablas, al existir 39 países distintos representaremos solo las dos primeras para evitar saturar el documento.

En la $Tabla$ 3.4.3 se ve la matriz de preferencias obtenida de la comparación de todas las tablas, como vemos, en el caso de la comparación $Australia - Bulgaria$, tenemos que $Australia$ es superior a $Bulgaria$ en 108.433 ocasiones, mientras que al contrario solo sumamos 37.257. Con la representación de las $Tablas$ 2.4.1 y 2.4.2, podemos probar la veracidad de la $Propiedad$ 2.2.1, pues si llamamos D a la matriz de preferencias de la $Tabla$ 3.4.3, entonces $D_{AUS-BGR} + D_{BGR-AUS} = 145{,}700$, que es el equivalente a realizar todas las combinaciones posibles, es decir $775 * 188 = 145{,}700$

Tabla 3.4.1: Tabla representativa nota matemáticas en Australia

	CNT	PVMATH
1	AUS	559.09
2	AUS	483.7
...	...	...
775	AUS	468.53

Tabla 3.4.2: Tabla representativa nota matemáticas en Bulgaria

	CNT	PVMATH
1	BGR	361.15
2	BGR	403.64
...	...	...
188	BGR	328.3

Tabla 3.4.3: Matriz de preferencias obtenida

	AUS	BGR	BRA	...	USA
AUS	0	110964	606105	...	69387
BGR	34736	0	117423	....	7922
BRA	44120	40309	0	...	11408
...	...	...	...	...	...
USA	55388	22142	125086	...	0

Por lo que se ve en las tablas antes mostradas, el número de individuos de cada país no es el mismo, por lo que siguiendo las explicaciones de los apartados anteriores, el procedimiento a seguir en este caso es normalizar la matriz de preferencias para ahorrarnos problemas ocasionados por un tamaño excesivamente grande o pequeño en las comparaciones. La nueva matriz de preferencias, ahora normalizada es la reflejada en la $Tabla\ 3.4.4$

Tabla 3.4.4: Matriz de preferencias normalizada

	AUS	BGR	BRA	...	USA
AUS	0	0.7616	0.9321	...	0.5561
BGR	0.2384	0	0.744	....	0.2685
BRA	0.0679	0.2556	0	...	0.0740
...	...	...	...	...	...
USA	0.4439	0.7315	0.9260	...	0

Podemos comprobar la adecuación del cálculo ejecutado, en primer lugar, porque la suma de todas las parejas cumple lo siguiente: $N_{ij} + Nji = 1$, suponiendo que N es la matriz de la $Tabla$ 3. 4. 4. Además, vemos que existe diferencias significativa entre las dos tablas, si tomamos como ejemplo USA, vemos como $D_{USA-AUS} = 55388$, y $D_{USA-BGR} = 22142$, sin embargo, cuando estas son normalizadas tenemos $N_{USA-AUS} = 0{,}4439$, y $N_{USA-BGR} = 0{,}7315$

Con esta matriz podemos dar por finalizada la etapa de procesamiento de datos en R-Studio. Cada vez que procesamos el algoritmo que compara las distintas tablas para encontrar la matriz de preferencias que acabamos de estudiar, el software tarda en torno a 30 minutos para un tamaño de datos equivalente al dado. Por ese motivo exportamos la matriz obtenida a un archivo CSV que pueda ser más adelante tratado.

3.5. Solucion modelo de optimizacion

En este apartado trabajaremos en otro software, LINGO/LINDO, comunmente utilizado en matemáticas para la resolución de sistemas lineales, El código completo se podrá consultar en el anexo, aunque resulta exactamente igual que el empleado en el *Apartado* 2,5 modificando la matriz de datos empleada y el tamaño de esta.

Cambiando estos dos detalles, el cálculo del problema se eleva a la resolución de 1.521 variables, y 55.576 restricciones, cuyo tiempo de resolución en este programa resulta insignificante. Evidentemente, por el tamaño del problema, resulta imposible interpretar la solución sin ayuda externa, para ello recurriremos a Excel, donde con una simple fórmula obtendremos la solución.

En la $Tabla$ 3. 5. 1 se representa la solución obtenida que refleja el orden que genera el modelo del LOP para clasificar a los países de mejor a peor. Aunque la tabla solo contenga dos variables $RANKING$, y $COUNTRY$, sobra definir que refleja cada una, esta se representa en tres columnas para ahorrar espacio.

Tabla 3.5.1: Solución LOP con datos PISA

RANKING	COUNTRY	RANKING	COUNTRY	RANKING	COUNTRY
1	SGP	14	AUS	27	SVN
2	HKG	15	ESP	28	HUN
3	KOR	16	FRA	29	ROU
4	JPN	17	NOR	30	THA
5	POL	18	GBR	31	TUR
6	NLD	19	LVA	32	CHL
7	EST	20	SWE	33	BGR
8	FIN	21	USA	34	MEX
9	CHE	22	ITA	35	URY
10	DEU	23	RUS	36	TUN
11	CAN	24	SVK	37	COL
12	CZE	25	ISR	38	IDN
13	NZL	26	LTU	39	BRA

4. Modificando el Problema de Ordenación Lineal

4.1. Objetivo

Una vez sabemos proceder correctamente con el cálculo y obtención de la solución del LOP, podemos empezar a pensar en posibles modificaciones del problema que den lugar a una solución mucho más robusta. En nuestro caso, sugerimos mejorar dicho modelo mediante la comparación de los individuos en subgrupos, evidentemente de un tamaño menor al inicial. El objetivo es encontrar grupos internamente homogéneos y externamente heterogéneos, con la idea de establecer comparaciones únicamente con los individuos más similares entre sí, y evitar comparar entre paises muy desparejos que simplemente crean un mayor peso, muchas veces descompensado en la matriz de preferencias.

Con ánimo de precisar el objetivo de este documento, volveremos a utilizar la base de datos PISA, la cual ha sido explicada con detalle en la sección anterior, incluida la transformación de los datos. Partimos de la matriz correspondiente a la $Tabla$ 3.4.4. Una vez sabemos esto, podemos especificar que con la aplicación de dicho modelo queremos conseguir 3 grupos de 13 individuos cada uno con las características antes mencionadas, además queremos ordenar internamente los grupos.

4.2. Modelo

Para comprender el modelo, tenemos que tener muy presente las indicaciones presentadas en el apartado anterior; en el caso del modelo original del LOP, buscamos estudiar todos los valores que quedan por encima de la diagonal superior, sin embargo, ahora tenemos que estudiar únicamente las comparaciones entre individuos que pertenezcan al mismo grupo.

Además, dicho modelo se modifica según el objetivo que busquemos, por ello se ha especificado en los párrafos anteriores el número de grupos e individuos que queremos conseguir. Se observará en el modelo el uso de las letras A para la restricción (6), y B para la (7), siendo estas las únicas que encuentren variaciones. Antes de presentar el nuevo modelo, recogemos en los siguientes puntos la explicación de los índices y variables que interactúan en este y que serán de utilidad para entender la mecánica matemática:

- P: Conjunto de individuos a comparar r,s $\in \{1,2,...,39\}$.
- K: Conjunto de grupos a realizar k,t $\in \{1,2,3\}$.
- x_{rs}: Variable binaria, toma valor de 1 cuando preferimos el individuo r al s, y ambos están en el mismo grupo.
- z_{rs}: Variable binaria, toma valor de 1 cuando preferimos el individuo r al s, y ambos no están en el mismo grupo.
- g_{kr}: Variable binaria, toma valor de 1 cuando asignamos el individuo r al grupo k.

$$
\begin{aligned}
\text{máx} \quad & \sum_{r\in P}\sum_{s\in P:r\neq s} m_{rs}x_{rs}\frac{1}{\sum_{r\in P}\sum_{s\in P:r\neq s} m_{rs}z_{rs}} & & \\
s.a \quad & x_{rs}+x_{sr}+z_{rs}+z_{sr}=1 & \forall\, r,s\in P: r<s & \quad (4)\\
& x_{rs}+x_{st}+x_{tr}+z_{rs}+z_{st}+z_{tr}\leq 1 & \forall\, r,s,t\in P: r,s,t \text{ pwd} & \quad (5)\\
& \sum_{r\in P}\sum_{s\in P:r\neq s} z_{rs}=A & & \quad (6)\\
& \sum_{r\in P} g_{kr}=B & \forall\, k\in K & \quad (7)\\
& \sum_{k\in K} g_{kr}=1 & \forall\, r\in P & \quad (8)\\
& x_{rs}+x_{sr}\leq 2-g_{kr}+g_{ts} & \forall\, r,s\in P; k,t\in K & \quad (9)\\
& x_{rs}\in\{0,1\} & \forall\, r,s\in P: r\neq s & \quad (10)\\
& z_{rs}\in\{0,1\} & \forall\, r,s\in P: r\neq s & \quad (11)\\
& g_{kr}\in\{0,1\} & \forall\, r\in P, k\in K & \quad (12)
\end{aligned}
$$

Tal y como hemos comentado, hacemos uso tanto de las z, como de las x para estudiar aquellos individuos que hay dentro de los grupos y buscar aquella combinación cuya proporción de valores dentro y fuera de todos los grupos sea la máxima posible. Así, si vemos la función objetivo, maximizamos los valores que vamos a introducir en los grupos, y minimizar los que estén fuera; este es el sentido de la función objetivo, y que permite establecer grupos internamente homogéneos, que por ende son externamente heterogéneos.

Con la restricción (4), estamos indicando la preferencia en cuanto a la comparación de dos individuos r, y s de la matriz, estén o no dentro del mismo grupo. La restricción (5) busca el mismo objetivo de establecer preferencias ahora con tres individuos r, s, y t, y que estas cumplan con el orden lógico, de nuevo interactuando con individuos del mismo o distintos grupos. La condición (6), caracterizada como hemos comentado anteriormente por el uso de la letra A, busca contar cuántos individuos no pertenecen al mismo grupo, este cálculo lo obtenemos del siguiente razonamiento explicado con la $Tabla\ 4.2.1$:

Tabla 4.2.1: Solución LOP con datos PISA

	1	2	3	4
1	0	m_{12}	m_{13}	m_{14}
2	m_{21}	0	m_{23}	m_{24}
3	m_{31}	m_{32}	0	m_{34}
4	m_{41}	m_{42}	m_{43}	0

En el hipotético caso de establecer los siguientes grupos $G1 = \{1, 2\},\ y\ G2 = \{3, 4\}$, es decir, los individuos 1 y 2 pertenecen al grupo 1, y los individuos 3 y 4 al grupo 2, las únicas comparaciones que se establecerían serían las del mismo grupo, dejando por fuera los valores m_{13},m_{14},m_{23}, y m_{24}, y dando lugar a la restricción:

- $\sum_{r\in P} \sum_{s\in P:r\neq s} z_{rs} = 4$

Este valor se puede calcular a mano como hemos hecho, sin embargo, resultaría muy tedioso repetirlo para la matriz de orden 39 que tenemos que estudiar, por lo que podemos obtener el valor con un cálculo. Adelantamos que el valor de A para las asunciones iniciales que hemos tomado equivale a 507. Para entender la fórmula descrita abajo será necesario saber que n_i hace referencia al tamaño del grupo i, y N es el orden de la matriz de preferencias. Apuntamos, que utilizamos el sumatorio para el caso en el que el número de individuos por grupo no fuera siempre el mismo, caso el cual podríamos aplicar al modelo muy sencillamente. De esta forma tenemos el siguiente cálculo:

- $A = \frac{N^2-N}{2} - \sum_{i\in I} \frac{n_i^2-n_i}{2} = \frac{16-4}{2} - \frac{4-2}{2} + \frac{4-2}{2} = 4$

La restricción (7), es otra de las restricciones que varía según el objetivo, en este caso el cálculo se reduce a la lógica, dando a B el valor de 13 para dicho caso, pues es el número de individuos de cada grupo, en caso de que quisiéramos que los grupos no tuvieran el mismo número de individuos lo podríamos forzar manualmente restringiendo para cada grupo, el número de individuos que lo componen. La restricción (8), indica que cada individuo debe pertenecer únicamente a un grupo, es decir, no podemos dejar ningún individuo sin grupo, ni que el mismo individuo pertenezca a más de uno, pues la primera supondría un problema y la segunda una incoherencia. Finalmente, la restricción (9) sirve para asociar los individuos que se van a comparar por estar en el mismo grupo; x, con el grupo al que pertenece cada individuo; g, y que no existan divergencias entre ambos, por simplificar la aclaración, que los individuos i, j que se comparan pertenezcan ambos al mismo grupo.

Para la realización de este modelo se han estudiado antes muchos brutos, que no han dado un buen resultado, pero que han ayudado a obtener la solución que comentaremos más adelante. Por ello, estos se incluirán en el anexo para que puedan servir como guía de qué caminos no tienen solución, y sirvan de ayuda e idea para trabajos e investigaciones futuras.

Si nos hemos fijado en el modelo que queremos estudiar, nos daremos cuenta de que no estamos ante un modelo lineal, por ello, habrá que aplicar una serie de modificaciones al mismo para que pueda ser perfectamente estudiado por los softwares que disponemos. Con estas modificaciones, observamos una importante transformación, a la cual requerimos añadir una serie de variables; y, v, y w que permiten su linealización. En este trabajo, no se explicarán los procedimientos seguidos para linealizar el modelo, simplemente se ofrecerá el resultado final.

En relación a los formatos del modelo, mantendremos la terminología utilizada anteriormente con las letras A, y B para indicar que en las restricciones (18), y (26), siguen existiendo modificaciones dependientes del número de individuos por grupo. Además, añadimos la letra M, que representa un valor inalcanzable para la consecución del objetivo; se puede consultar en el Anexo los valores utilizados para las distintas matrices, en este caso, equivale a 1,000.

$$
\begin{aligned}
\text{máx} \quad & \sum_{r\in P} \sum_{s\in P: r\neq s} m_{rs} w_{rs} & & \\
s.a \quad & x_{rs} + x_{sr} + z_{rs} + z_{sr} = 1 & \forall\, r,s \in P : r < s & \quad (13)\\
& x_{rs} + x_{st} + x_{tr} + z_{rs} + z_{st} + z_{tr} \leq 1 & \forall\, r,s,t \in P : r,s,t \text{ pwd} & \quad (14)\\
& \sum_{r\in P} \sum_{s\in P: r\neq s} z_{rs} = A & & \quad (15)\\
& & & \quad (16)\\
& \sum_{r\in P} \sum_{s\in P: r\neq s} m_{rs} v_{rs} = 1 & & \quad (17)\\
& w_{rs} - y \leq 0 & \forall\, r,s \in P : r < s & \quad (18)\\
& w_{rs} - M x_{rs} \leq 0 & \forall\, r,s \in P : r < s & \quad (19)\\
& & & \quad (20)\\
& v_{rs} - y \leq 0 & \forall\, r,s \in P : r < s & \quad (21)\\
& v_{rs} - M z_{rs} \leq 0 & \forall\, r,s \in P : r < s & \quad (22)\\
& v_{rs} - y - M z_{rs} \geq -M & \forall\, r,s \in P : r < s & \quad (23)\\
& \sum_{r\in P} g_{kr} = B & \forall\, k \in K & \quad (24)\\
& \sum_{k\in K} g_{kr} = 1 & \forall\, r \in P & \quad (25)\\
& x_{rs} + x_{sr} \leq 2 - g_{kr} + g_{ts} & \forall\, r,s \in P; k,t \in K & \quad (26)\\
& x_{rs} \in \{0,1\} & \forall\, r,s \in P : r \neq s & \quad (27)\\
& z_{rs} \in \{0,1\} & \forall\, r,s \in P : r \neq s & \quad (28)\\
& g_{kr} \in \{0,1\} & \forall\, r \in P, k \in K & \quad (29)
\end{aligned}
$$

4.3. Aplicación a los datos PISA

Antes de comentar los resultados de la aplicación del modelo presentado sobre los datos PISA, cabe mencionar que esta no es la solución más óptima posible, pues el tiempo de carga del modelo se extiende indefinidamente, de hecho, el procesamiento más largo que se ha realizado ha superado las 358 horas, se adjunta una captura de dicha carga en el Anexo, es decir, cerca de 15 días cargando para obtener una solución.

Debido a dicho obstáculo encontrado, hemos decidido acotar el tiempo de procesamiento del programa a unas pocas horas, lo que ha dado lugar a la siguiente solución; mostrada en la *Tabla* 4.3.1.

Tabla 3.5.1: Solución nuevo modelo con datos PISA

Clasificación	Grupo 1	Clasificación	Grupo 2	Clasificación	Grupo 3
1	POL	1	HKG	1	SGP
2	NLD	2	KOR	2	JPN
3	EST	3	CAN	3	FIN
4	CZE	4	FRA	4	CHE
5	NZL	5	NOR	5	DEU
6	ESP	6	GBR	6	THA
7	AUS	7	LVA	7	CHL
8	SWE	8	ITA	8	BGR
9	USA	9	RUS	9	MEX
10	SVN	10	SVK	10	URY
11	ROU	11	ISR	11	TUN
12	TUR	12	LTU	12	COL
13	IDN	13	HUN	13	BRA

Hemos observado que a medida que aumentamos el tiempo de carga, mejor es la solución que obtenemos, al menos en el sentido de aproximación al objetivo inicialmente planteado. Es decir, más homogéneos son los grupos internamente, y más heterogéneos externamente, sin embargo, esta matriz no resulta útil para comparar los algoritmos utilizados.

Otro punto muy importante a comentar sobre el algoritmo es la importancia que tiene en la actualidad este tipo de modelos. Durante el pasado mes de mayo de 2024, en el congreso *International Symposium on Combinatorial Optimization* se presentó el paper ***Cycle of Clusters***, añadido a la bibliografía del trabajo, una investigación cuyo objetivo es generar clusters de individuos (como se entiende, homogéneos internamente y heterogéneos respecto a los demás), y después ordenarlos entre sí. Como vemos, el alcance de dicha investigación es muy similar al nuestro, sumando que nosotros ordenamos además internamente los clusters, añadiendo de esta manera un punto de complejidad y sofisticación al resultado.

Gracias a dicha investigación, obtenemos ideas sobre cómo acortar el tiempo de procesamiento del modelo; por ejemplo, mediante restricciones de desigualdad, ya sean triangulares, de partición, o de subviaje, que se detallan en el paper de la investigación. Gracias a la asistencia a este congreso de la tutora Mercedes, podemos entender cómo de relevante es el modelo que estamos creando, y cómo de actual es este tema en la Investigación Operativa.

4.4. Comparación de algoritmos

Debido al tiempo de procesamiento que requiere el nuevo modelo formulado para resolver cualquier matriz grande, hemos recortado la matriz de los datos PISA a 12 individuos para poder hacer pruebas y comparar cómo funciona nuestro modelo respecto al LOP.

Para poder tener unas comparaciones efectivas, debemos comparar qué solución nos ofrece el LOP para dicha matriz, qué solución ofrece nuestro modelo, y cómo de buena es la solución si forzamos el resultado del LOP en nuestro modelo, lo que equivale a partir la solución del LOP en tantos grupos como hayamos probado en nuestro modelo. En la siguiente $Tabla$ 4. 4. 1, se podrá consultar el resultado ordenar los individuos con los tres criterios mencionados. Destacamos además, que, para ahorrar espacio en el documento, adjuntaremos el código de Lingo, junto a las matrices utilizadas en el Anexo del trabajo.

Tabla 3.5.1: Solución nuevo modelo con datos PISA

Clasificación LOP		Clasificación Modelo		LOP aplicado al Modelo	
1	SGP	1	SGP	1	SGP
2	HKG	2	HKG	2	HKG
3	FIN	3	FIN	3	FIN
4	DEU	4	ESP	4	DEU
5	ESP	5	DEU	5	ESP
6	NOR	6	NOR	6	NOR
7	ISR	7	ISR	7	ISR
8	ROU	8	COL	8	ROU
9	THA	9	ROU	9	THA
10	COL	10	THA	10	COL
11	IDN	11	IDN	11	IDN
12	BRA	12	BRA	12	BRA

Para dichos resultados cabe destacar que el modelo que nosotros hemos programado considera 3 grupos con 4 individuos para cada uno, por lo que las dos últimas columnas de la $Tabla\ 4.4.1$, $Clasificación\ Modelo$, y $LOP\ aplicado\ al\ Modelo$, se dividen en grupos cada 4 individuos, esto no lo hemos representado porque no es relevante para el análisis de los resultados.

Como apreciamos, existen diferencias entre el resultado que nos ofrece el LOP y el que ofrece el Modelo, cambiando el orden de los países $ESP - DEU - ROU - COL - THA$. Lo que buscamos encontrar ahora es si el modelo que hemos creado mejora la solución del LOP, para ello nos centraremos en los objetivos de los modelos, los cuales son los siguientes:

- LOP: 52,11
- LOP aplicado al Modelo: 1,17989
- Modelo: 1,276524

Vistos los resultados, y sabiendo que el objetivo del Modelo es maximizar el objetivo, podemos concluir de esta forma, que existen diferencias entre el LOP (comparar todos los individuos posibles), con nuestro modelo (comparar solo aquellos individuos que clasificamos en el mismo grupo), y que además esta diferencia es ventajosa por un lado, en cuanto a resultado obtenido, y por otro, en cuanto a diversidad de la respuesta. Específicamente, la solución del Modelo mejora el ajuste del LOP en un 8,2 %, por lo que existe un incremento significativo. Destacamos además de estos tres puntos mencionados arriba, que la respuesta del LOP no se puede comparar con los otros dos por tener un objetivo completamente distinto, pero es útil como información que refleja las tres soluciones.

5. Propuesta aplicación modelo para Universidad Miguel Hernández

En este apartado describiremos la propuesta que pretendemos hacer para la Universidad Miguel Hernández, en la que detallaremos cómos se puede aplciar el Modelo que hemos inventado, para ofrecer una alternativa que mejore un sistema de clasificación que utiliza un criterio poco robusto. El foco de la propuesta no es otro que dar valor al trabajo realizado destacando que no se trata simplemente de un avance teórico en el campo de la Investigación Operativa, sino que además tiene relevantes implicaciones prácticas, y que pueden empezar a ser aplicadas en sistemas internos de la Universidad.

5.1. Contexto de la propuesta

La Universidad Miguel Hernández homenajea anualmente a los mejores estudiantes de cada grado universitario, para dicho reconocimiento, el criterio de selección se basa en la media global del total de asignaturas que se cursan durante los cuatro años de carrera, y aquel expediente con mayor media es premiado por su trabajo. Mientras trabajábamos en la creación del algoritmo, surgió internamente la idea de ofrecer otro criterio sobre el sistema de clasificación de alumnos.

5.2. Objetivo

Con esta propuesta buscamos demostrar que el criterio de clasificación que utiliza la media para reflejar el orden de los individuos no resulta el más robusto, y menos en este tipo de casos donde disponemos de una gran cantidad de datos para trabajar, donde además existen casos en los cuales el alumno graduado ha cursado un número distinto de asignaturas que sus compañeros por haberlas convalidado.

Con esta demostración, pretendemos poder trabajar conjuntamente con departamentos internos de la Universidad Miguel Hernández para ayudar a escoger el mejor expediente de cada promoción de una forma más sensata, y teniendo en cuenta la información del número total de asignaturas cursadas.

5.3. Datos y modelo

Aunque nos hubiera gustado poder trabajar con datos reales para hacer la aplicación sobre un caso ya existente, no ha sido posible tener acceso a estos de ninguna manera, por lo que hemos tenido que simular las 40 notas de 12 alumnos para poder trabajar. Esta simulación se ha realizado en *Excel*, y los datos se han trabajado en $R-Studio$ para luego pasarlos a *Lingo* y poder obtener resultados. Se pueden consultar los datos utilizados en el Anexo en la descripción.

Los datos se han creado aleatoriamente utilizando valores comprendidos entre el 5 y el 10, esto es así porque no puede ser considerado ningún alumno para el premio extraordinario, si no ha finalizado la carrera, y para ello se requiere haber superado todas las asignaturas. Respecto a la simulación de los datos, las 12 observaciones han quedad muy centradas en torno al 8,5, por lo que los resultados no serán totalmente fieles a la realidad de las universidades.

Respecto a nuestro modelo, es importante comentar que vamos a trabajar tal y como lo hemos hecho durante las anteriores aplicaciones, con 3 grupos de 4 individuos cada uno, por lo que en la *Tabla* 5.4.1, habrá que tener en cuenta que existe separación de grupos cada 4 individuos.

5.4. Solución

Para la base de datos que hemos escogido sobre estos 12 individuos, no apreciamos cambios notables en las primeras posiciones. Esto se debe en primer lugar a que la base de datos no representa fielmente la distribución de las notas de los estudiantes, y no existe forma de encontrar dicha distribución porque no disponemos de ningunos datos originales que sirvan de guía. En segundo lugar, y quitando el efecto de la base de datos, si no existieran cambios, o modificaciones en el orden de la clasificación podría ser una mera coincidencia en los datos, pues el criterio de ordenamiento es completamente distinto.

Como apreciamos con los siguientes resultados, patentes en la $Tabla$ 5. 4. 1, al igual que se producen cambios en las posiciones más bajas, esto se podrían haber dado en las más altas obteniendo un resultado totalmente distinto de los premios extraordinarios.

Para entender la $Tabla$ 5. 4. 1, destacamos que cuando indicamos, $A.X$, siendo X un valor entre uno y doce, hacemos referencia a $Alumno.X$ el número que lo identifica para poder llevar un seguimiento de su variación a media que aplicamos los distintos resultados.

Tabla 3.5.1: Solución nuevo modelo con datos PISA

Clasificación media	Clasificación LOP	Clasificación Modelo
A.12	A.12	A.12
A.8	A.8	A.9
A.9	A.9	A.8
A.2	A.2	A.2
A.6	A.6	A.6
A.7	A.7	A.7
A.3	A.3	A.3
A.4	A.1	A.10
A.1	A.10	A.1
A.11	A.4	A.11
A.10	A.11	A.4
A.5	A.5	A.5

Como hemos comentado antes, las posiciones más bajas son las que más cambios tienen, pero se debe a una mera coincidencia, pues los datos se han calculado de una forma aleatoria, y no siguiendo ningun tipo de distribución, por lo que la única forma de probar el cambio que produciría dicho algoritmo en la práctica sería obteniendo los datos reales.

6. Conclusión del trabajo

En este último apartado de redacción, comentaremos las conclusiones que obtenemos después de muchos meses de trabajo sobre este algoritmo, así como las asunciones que podemos hacer de cara a un futuro.

En primer lugar, hay que destacar que gracias a esta investigación, existe un avance en la clasificación de individuos y una mejora respecto al modelo original LOP, pues ahora estamos dividiendo en grupos internamente homogéneos y externamente heterogéneos, que a su vez se clasifican internamente de mayor a menor. Sabemos que existe una mejora, porque en primer lugar obtenemos una solución distinta al LOP, es decir, no es lo mismo clasificar comparando todos los individuos entre sí, que hacerlo comparando únicamente aquellos que forman parte del mismo grupo y, en segundo lugar, porque al hacerlo así es, como se ha visto en el documento, mejor.

Por otro lado, es importante comentar como punto negativo sobre el algoritmo que hemos realizado, el tiempo de carga que requiere para encontrar soluciones. Resulta muy contraproducente tener que esperar la cantidad de horas vistas en el documento para poder concluir y avanzar con las investigaciones, este hecho ha ralentizado los avances. Esto se puede dar porque, uno, en nuestro modelo (por una cuestión de tiempo), no podemos añadir restricciones desigualdad que agilicen la búsqueda del objetivo, y con las cuales adelantaríamos muchas horas de trabajo, y/o dos, porque el software que utilizamos no es realmente potente para el cálculo de problemas de esta dimensión, que requieren consecuentemente un número de restricciones realmente alto.

Apoyando dicho punto negativo, sería muy interesante continuar dicha investigación añadiendo las restricciones de desigualdad comentadas para poder estudiar matrices mucho más grandes con mucho menor tiempo de carga, sería en este momento cuando el Modelo conseguiría su máximo potencial logrando alcances sorpresivos para muchos campos.

Divagando sobre estos alcances, podríamos crear ránquines de todo tipo, con tamaños de grupos sin restricciones para estudiar otro tipo de información que requiere de un volumen de datos más grande, por ejemplo, ciudades de un país, restaurantes de una provincia, ránquines deportivos, y otras muchas opciones que irían apareciendo a medida que se indagara más en el problema.

Otro de los puntos en los que se puede seguir trabajando en este Modelo es en la asignación de individuos por grupo. Actualmente, somos nosotros los que debemos indicar manualmente la distribución equitativa (o no) de los individuos en cada grupo, sin embargo, podría ser muy interesante obtener el valor óptimo dejando que el algoritmo elija la distribución que requiera. Este problema ha estado rondando en nuestras cabezas durante todos estos meses, y ahora más que nunca antes parece muy factible gracias a la alta calidad del modelo que hemos generado, sin embargo, y como hemos ido comentando, por una cuestión de presupuesto temporal hemos tenido que abandonar la tarea.

Al igual que queremos añadir en el Anexo esos modelos y pruebas que no han dado resultado, también consideramos oportuno comentar como hemos hecho en el párrafo anterior, esos puntos que han quedado pendientes, con el ánimo de volver tan pronto como sea al documento y ser capaces de continuar por donde lo hemos dejado sin perder las ideas que tenemos, además de que puedan servir de ayuda para futuros investigadores que quieran ilustrarse con nuevas ideas.

7. Bibliografía

Ailon, N., Charikar,M.,&Newman, A. (2008).Aggregating inconsistent information: Ranking and clustering. Journal of the ACM, 55, 1–27.

Arrow, K. J. (1951). Social choice and individual values (p. 16). New York: Wiley.

Burkard, R. E.,& Fincke, U. (1985). Probabilistic asymptotic properties of some combinatorial optimization problems. Discrete Applied Mathematics, 12, 21–29.

Charon, I., & Hudry, O. (2007). A survey on the linear ordering problem for weighted or unweighted tournament. 4OR, 5, 5–60.

Charon, I., & Hudry, O. (2010). An updated survey on the linear ordering problem for weighted or unweighted tournament. Annals of Operations Research, 175, 107–158.

Charon, I.,&Hudry, O. (2011).Maximum distance between Slater orders and Copeland orders of tournaments. Order, 28, 99–119.

Fagin, R., Kumar, R., Mahdian, M., Sivakumar, D., & Vee, E. (2006). Comparing partial rankings. SIAM Journal on Discrete Mathematics, 20, 628–648.

García-Nové, E.M. (2018). Nuevos problemas de agregación de rankings: Modelos y algoritmos, PhD Thesis. Spain: University Miguel Hernández of Elche.

García-Nové, E. M., Alcaraz, J., Landete, M., Puerto, J., & Monge, J. F. (2017). Rank aggregation in cyclic sequences. Optimization Letters, 11, 667–678.

Glover, F., Klastorin, T., & Kongman, D. (1974). Optimal weighted ancestry relationships. Management Science, 20(8), 1190–1193.

Hudry, O. (2010). On the complexity of Slater's problem. European Journal of Operational Research, 203, 216–221.

Kemeny, J. (1959). Mathematics without numbers. Daedalus, 88, 577–591.

Kendall, M. (1938). A new measure of rank correlation. Biometrika, 30, 81–89.

Lietz, P., Cresswell, J. C., Adams, R. J., & Rust, K. F. (Eds.). (2017). Implementation of large-scale education assessments. Hoboken: Wiley.

Martí, R., & Reinelt, G. (2011). The linear ordering problem: Exact and heuristic methods in combinational optimization (1st ed.). Berlin: Springer.

OECD. (2014). PISA 2012 Technical Report. Paris: OECD Publishing. Retrived March 21, 2018 from www.oecd.org/pisa/pisaproducts/pisa2012technicalreport.htm

Rahmaniani, R., Crainic, T. G., Michel, Gendreau,& Rei, W. (2017). The Benders decomposition algorithm: A literature review. European Journal of Operational Research, 259(3), 801–817.

Sculley, D. (2008). Rank aggregation for similar items. In Proceedings of the 2007 SIAM international conference on data mining.

Slater, P. (1961). Inconsistencies in a schedule of paired comparisons. Biometrika, 48, 303–312.

Tromble, R., & Eisner, J. (2009). Learning linear ordering problems for better translation. In Proceedings of the 2009 conference on empirical methods in natural language processing (Vol. 2, pp. 1007–1016). EMNLP.

Tsoukalas, A., Rustem, B., & Pistikopoulos, E. N. (2009). A global optimization algorithm for generalized semi-infinite, continuous minimax with coupled constraints and bi-level problems. Journal of Global Optimization, 44, 235–250.

Young, H. P., & Levenglick, A. (1978). A consistent extension of condorcet's election principle. SIAM Journal on Applied Mathematics, 35, 285–300.

Zahid, M. A., & Swart, H. (2015). The borda majority count. Information Sciences, 295, 429–440.

Arapicio, J., Landete, M., & Monge, J. F. (2019). A Linear Ordering Problem of Sets

Eifler, L., Witzig, J., & Gleixner, A. F. (2024). Branch and Cut for Partitioning a Graph into a Cycle of Clusters

8. Anexo

8.1. Bases de datos utilizadas

Tabla 8.8.1: Base de datos PISA cruda

CNT	UNIT	SCHOOLID	PVMATH	PVREAD	PVSCIE
ESP	4679	486	442.73	447.75	453.64
MEX	9054	13	336.9	322.45	324.39
CHL	3212	116	474.26	475.58	486.53
LTU	8637	21	442.91	456.58	467.1
EST	5157	62	464.49	454.55	513.6
MEX	10309	1269	392.12	375.89	395.79
LTU	8729	114	386.97	424.19	425.57
POL	11079	17	537.07	540.89	508.04
MEX	9110	69	411.51	410.07	421.31
IDN	6754	58	366.49	398.25	397.94
CHE	3014	329	539.16	509.31	501.88
BRA	1387	424	407.09	417.19	398.75
AUS	476	476	560.61	548.59	558.02
DEU	4032	69	459.13	468.95	474.94
HUN	6521	28	479.74	497.81	518.64
MEX	9302	261	534.13	581.3	532.09
RUS	11565	141	437.54	428.24	427.8
GBR	6140	302	459.91	463.54	484.62
NZL	11030	145	391.88	373.23	359.76
CHE	2967	282	459.68	452.47	421.51
RUS	11503	79	582.94	503.93	526.17
JPN	8349	80	659.02	663.32	657.33
LVA	9000	170	506.52	479.41	519.27
SVN	12150	100	479.47	438.35	497.54
THA	12742	150	393.27	395.57	416.32
MEX	9384	343	428.69	391.72	403.65
THA	12770	178	414.19	425.64	415.53
ESP	4518	325	510.95	495.44	499.96
ESP	4996	803	491.24	482.92	511.83
MEX	10011	971	390.1	387.64	373.03
AUS	617	617	507.84	519.99	515.19
SVK	11936	114	483.02	469.17	482.76
AUS	750	750	453.97	468.06	481.79
HUN	6669	177	348.27	346.64	376.49

Acceso a la tabla completa en el siguiente enlace Link

Acceso a tabla formato csv en el siguiente enlace Link

Tabla 8.8.1: Base de datos universidad cruda (1)

A.1	A.2	A.3	A.4	A.5	A.6
9,81	7,15	7,27	8,23	9,96	9,46
9,92	7,52	9,47	8,22	7,39	9,27
8,07	9,65	8,41	9,38	7,96	8,77
7,38	9,3	8,44	9,57	7,03	7,65
7,26	9,12	7,01	7,67	9,75	7,41
9,96	7,51	7,88	8,41	7,41	7,89
7,65	7,77	7,12	8,88	9,32	8,53
7,18	8,31	7,51	8,4	9,7	7,69
7,64	8,07	9,58	7	8,35	7,96
7,22	9,57	9,86	7,79	9,96	7,08
9,11	7,21	9,44	9,04	7,8	9,61
8,44	8,37	9,21	8,68	7,97	8,03
8,17	9,07	8,9	7,36	9,83	8,96
7,79	8,1	7,21	7,85	7,35	8,68
9,27	8,65	7,01	7,22	7,63	7,94
8,47	9,3	8,34	8,2	9,13	8,85
8,48	8,39	7,53	8,75	9,23	9,35
9,65	8,64	7,97	8,32	7,97	7,76
7,5	9,31	8,82	9,36	9,22	9,22
8,04	7,66	8,16	8,36	7,91	8,43
8,08	7,58	8,7	7,81	7,45	7,78
8,39	9,29	9,39	8,81	7,51	8,86
9,29	9,48	9,33	9,51	9,71	9,26
9,6	8,13	8,03	7,9	8,57	9,49
8,79	9,57	7,38	7,2	7,48	9,73
8,13	7,57	9,67	7,42	7,48	8,33
8,91	8,73	7,17	8,73	7,83	7,7
8,28	8,74	8,51	8,12	7,85	9,93
8,42	7,81	9,85	9,3	7,92	7,05
7,3	7,95	7,51	8,1	7,42	7,02
7,61	8,05	9,32	9,18	8,58	9,96
9,93	7,14	7,91	9,99	7,62	9,55
8,03	9,12	8,52	7,19	9,19	7,59
7,01	7,33	8,55	7,54	7,18	9,76
9,13	9,82	8,13	8,07	8,11	7,25
7,45	9,49	7,73	7,35	8,27	7,18
7,03	8,13	8,01	8,68	7,1	8,25
7,24	8,99	9,89	8,42	7,98	7,11
8,51	9,64	9,21	7,64	9,38	8,8
9,7	7,27	7,39	9,14	7,27	9,3

Tabla 8.8.1: Base de datos universidad cruda (2)

A.7	A.8	A.9	A.10	A.11	A.12
7	8,99	9,98	7,35	8,13	8,35
7,57	7,63	7,46	9,53	9,36	8,66
9,61	8,15	9,16	8,25	7,18	7,74
9,74	8,53	7,08	9,68	9,01	8,99
8,29	9,56	9,15	9,84	7,24	8,79
9,39	7,43	7,88	7,14	7,86	9,93
7,01	7,12	8,53	8,15	7,76	8,76
9,69	7,25	7,39	7,1	7,07	7,61
9,18	9,78	7,36	7,69	8,01	9,15
9,95	9,98	9,48	8,01	9,13	8,57
7,85	9,87	9,16	8,45	9,43	9,7
8,96	9,83	9,04	8,66	7,8	7,14
8,11	8,94	9,47	7,88	8,18	7,15
8,62	9,19	8,65	7,89	9,71	8,4
8,46	7,08	8,18	8,38	7,67	7,7
8,45	8,62	7,73	8,6	8,67	9,98
9,81	9,21	9,61	8,49	7,7	9,3
9,48	8,75	7,96	7,25	8,17	9,72
8,49	7,19	9,96	7,22	9,29	8,11
8,33	9,56	7,11	9,66	8,34	8,41
8,82	8,25	9	8,71	7,89	7,5
7,73	8,45	7,77	7,53	7,4	8,37
9,41	9,85	9,61	7,55	8,5	9,79
9,13	7,61	9,28	9,28	9,83	8,09
8,04	9,91	8,2	7,73	7,96	9,29
8,37	7,97	8,65	9,01	7,59	9,69
8,21	10	8,01	7,42	9,21	7,7
7,11	8,75	9,79	8,15	9,42	8,87
7,13	7,62	8,66	9,51	8,61	7,75
7,76	8,81	8,03	9,24	9,64	8,45
7,81	8,54	8,87	7,39	7,82	9,49
7,29	9,2	8,8	7,44	7,94	9,68
8,61	8,05	8,3	7,72	8,52	8,94
8,49	8,75	8,88	9,39	7,63	9,77
9,57	7,81	7,31	8,77	7,84	7,92
7,71	8,21	7,86	9,84	8,55	7,11
7,69	9,66	8,56	9,04	8,2	8,59
7,5	8,02	7,46	9,67	7	8,71
7,85	7,57	9,83	7,54	9,79	9,14
7,83	8,84	7,51	7,13	7,29	8,25

8.2. Programación R-Studio para consecución de matrices

Código utilizado para obtención de matrices de preferencia PISA

```
#library(tidyverse)
library(backports)


#Carga de datos

datos=read.table("DATOS_PISA2012.txt", header=T, sep="", dec=".")
datos
datos$CNT=as.factor(datos$CNT)
datos
paises=levels(datos$CNT)

base_mates=vector(mode="list", length=length(paises))
for (i in 1:length(paises)){
  x=data.frame(subset(datos,CNT==paises[i]))
  x=cbind(x[1],x[4])
  base_mates[[i]]=x
}
base_read=vector(mode="list", length=length(paises))
for (i in 1:length(paises)){
  x=data.frame(subset(datos,CNT==paises[i]))
  x=cbind(x[1],x[5])
  base_read[[i]]=x
}
base_read
base_scie=vector(mode="list", length=length(paises))
for (i in 1:length(paises)){
  x=data.frame(subset(datos,CNT==paises[i]))
  x=cbind(x[1],x[6])
  base_scie[[i]]=x
}
base_scie
length(base_mates)==length(paises)

dim=length(base_mates)
matriz=matrix(0,nrow=dim,ncol=dim)
colnames(matriz)=paises[1:dim]
rownames(matriz)=paises[1:dim]
View(matriz)

#-----------------------------------------------
contador=0
for (c in 1:(dim-1)){
  for (d in (c+1):(dim)){
    contador=0
    for (i in 1:nrow(base_mates[[c]])){
      for (j in 1:nrow(base_mates[[d]])){
        contador=(contador+ifelse(base_mates[[c]][i,2]>base_mates[[d]][j,2],1,0)
          )
      }
    }
    matriz[c,d]=contador
  }
}
matriz
View(matriz)
#-----------------------------------------------
```

```
contador=0
for (c in 1:(dim-1)){
  for (d in (c+1):(dim)){
    contador=0
    for (i in 1:nrow(base_read[[c]])){
      for (j in 1:nrow(base_read[[d]])){
        contador=(contador+ifelse(base_read[[c]][i,2]>base_read[[d]][j,2],1,0))
      }
    }
    matriz[c,d]=contador
  }
}
matriz
View(matriz)
write.csv(matriz, "matriz_pisa_read.csv")
#--------------------------------------------
#--------------------------------------------
contador=0
for (c in 1:(dim-1)){
  for (d in (c+1):(dim)){
    contador=0
    for (i in 1:nrow(base_scie[[c]])){
      for (j in 1:nrow(base_scie[[d]])){
        contador=(contador+ifelse(base_scie[[c]][i,2]>base_scie[[d]][j,2],1,0))
      }
    }
    matriz[c,d]=contador
  }
}
matriz
View(matriz)
write.csv(matriz, "matriz_pisa_scie.csv")
#-------------------------------------------- MATRIZ MATES ------------

matriz=read.csv2("matriz_pisa.csv",sep=",")
rownames(matriz)=matriz[,1]
matriz=matriz[,2:ncol(matriz)]
View(matriz)

matriz2=matrix(0,nrow=dim,ncol=dim)

for (i in 2:nrow(matriz)){
  for (j in 1:(i-1)){
    matriz2[i,j]=((nrow(base_mates[[i]]))*(nrow(base_mates[[j]])))
  }
}

View(matriz2)
colnames(matriz2)=rownames(matriz2)=colnames(matriz)
View(matriz2)

nrow(base_mates[[1]])*nrow(base_mates[[24]])



matrizb=matriz2-t(matriz)

View(matrizb)

for (i in 2:nrow(matriz)){
```

```
  for (j in 1:(i-1)){
    matriz[i,j]=matrizb[i,j]
  }
}
View(matriz)

#------------------------------------------ MATRIZ READ ----------

matriz=read.csv2("matriz_pisa_read.csv",sep=",")
rownames(matriz)=matriz[,1]
matriz=matriz[,2:ncol(matriz)]
View(matriz)

matriz2=matrix(0,nrow=dim,ncol=dim)

for (i in 2:nrow(matriz)){
  for (j in 1:(i-1)){
    matriz2[i,j]=((nrow(base_read[[i]]))*(nrow(base_read[[j]])))
  }
}

View(matriz2)
colnames(matriz2)=rownames(matriz2)=colnames(matriz)
View(matriz2)

nrow(base_read[[1]])*nrow(base_read[[24]])



matrizb=matriz2-t(matriz)

View(matrizb)

for (i in 2:nrow(matriz)){
  for (j in 1:(i-1)){
    matriz[i,j]=matrizb[i,j]
  }
}
View(matriz)

write.csv(matriz,"matriz_pisa2_read.csv",row.names= F)


#------------------------------------------ MATRIZ SCIE ----------

matriz=read.csv2("matriz_pisa_scie.csv",sep=",")
rownames(matriz)=matriz[,1]
matriz=matriz[,2:ncol(matriz)]
View(matriz)

matriz2=matrix(0,nrow=dim,ncol=dim)

for (i in 2:nrow(matriz)){
  for (j in 1:(i-1)){
    matriz2[i,j]=((nrow(base_mates[[i]]))*(nrow(base_mates[[j]])))
  }
}

View(matriz2)
colnames(matriz2)=rownames(matriz2)=colnames(matriz)
```

```
View(matriz2)

nrow(base_mates[[1]])*nrow(base_mates[[24]])



matrizb=matriz2-t(matriz)

View(matrizb)

for (i in 2:nrow(matriz)){
  for (j in 1:(i-1)){
    matriz[i,j]=matrizb[i,j]
  }
}
View(matriz)
## ejemplo de comprobacion - perfecto

write.csv(matriz,"matriz_pisa2_scie.csv")

which(verdad==FALSE)
verdad

write.csv(matriz,"matriz_pisa2_scie.csv",row.names=F)

matriz=read.csv2("matriz_pisa2_scie.csv",sep=",")
matriz
nombres=colnames(matriz)
rownames(matriz)=nombres

verdad=c()
for(i in 1:38){
  for(j in (i+1):39){
    verdad=c(verdad,matriz[i,j]+matriz[j,i]==nrow(base_mates[[i]])*nrow(base_
        mates[[j]]))
  }
}
verdad
which(verdad=="TRUE")
(39*39-39)/2==length(which(verdad=="TRUE"))

View(matriz)

### Visor de matrices ###

  #1) Read
matriz=read.csv("matriz_pisa2_read.csv")
rownames(matriz)=colnames(matriz)
View(matriz)

  #2) Scie
matriz=read.csv("matriz_pisa2_scie.csv")
rownames(matriz)=colnames(matriz)
View(matriz)

  #3) Mat
matriz=read.csv("matriz_pisa2_mat.csv")
rownames(matriz)=colnames(matriz)
View(matriz)
```

```
## Cambio de matrices a tanto por 1 ##
  # Mat ——————————————
matriz=read.csv("matriz_pisa2_mat.csv")
rownames(matriz)=colnames(matriz)
View(matriz)

matriz_nueva_mat=matrix(0,ncol=39,nrow=39)
for (i in 1:39){
  for(j in 1:39){
    matriz_nueva_mat[i,j]=matriz[i,j]/(matriz[i,j]+matriz[j,i])
  }
}
matriz_nueva_mat[is.na(matriz_nueva_mat)]=0
matriz_nueva_mat=round(matriz_nueva_mat, digits=4)
View(matriz_nueva_mat)
write.csv(matriz_nueva_mat,"mat_norm_mat.csv",row.names=F)
  # Scie ——————————————
matriz=read.csv("matriz_pisa2_scie.csv")
rownames(matriz)=colnames(matriz)
View(matriz)

matriz_nueva_scie=matrix(0,ncol=39,nrow=39)
for (i in 1:39){
  for(j in 1:39){
    matriz_nueva_scie[i,j]=matriz[i,j]/(matriz[i,j]+matriz[j,i])
  }
}
matriz_nueva_scie[is.na(matriz_nueva_scie)]=0
matriz_nueva_scie=round(matriz_nueva_scie, digits=4)
View(matriz_nueva_scie)
write.csv(matriz_nueva_scie,"mat_norm_scie.csv",row.names=F)
  # Read ——————————————
matriz=read.csv("matriz_pisa2_read.csv")
rownames(matriz)=colnames(matriz)
View(matriz)

matriz_nueva_read=matrix(0,ncol=39,nrow=39)
for (i in 1:39){
  for(j in 1:39){
    matriz_nueva_read[i,j]=matriz[i,j]/(matriz[i,j]+matriz[j,i])
  }
}
matriz_nueva_read[is.na(matriz_nueva_read)]=0
matriz_nueva_read=round(matriz_nueva_read, digits=4)
View(matriz_nueva_read)
write.csv(matriz_nueva_read,"mat_norm_read.csv",row.names=F)


c=c("SGP" ,"HKG" ,"IDN" ,"BRA")
seis=matrix(0,nrow=6,ncol=6)

matriz=read.csv("matriz_pisa2_mat.csv")
paises=colnames(matriz)
colnames(matriz_nueva_mat)=paises
rownames(matriz_nueva_mat)=paises
for (i in 1:6){
  for(j in 1:6){
```

```
      A=c[i]
      B=c[j]
      seis[i,j]=matriz_nueva_mat[A,B]
    }
}

seis

mates=read.csv("mat_norm_mat.csv")
View(mates)

sumas=rowSums(mates)
frame=data.frame()
for (i in 1:39){
  frame[i,1]=as.numeric(sumas[i])
}
frame
row.names(frame)=paises
frame
frame <- arrange(frame, v1)
library(dplyr)

# Prueba 2 matriz solucionada

mates=read.csv2("mat_norm_mat.csv",sep=",")
colnames(mates)=paises
row.names(mates)=paises
View(mates)
eleccion=c("SGP","BRA","DEU","ROU","ESP","ISR")
matriz62=matrix(nrow=6,ncol=6)
colnames(matriz62)=eleccion
row.names(matriz62)=eleccion

for (i in eleccion){
  for(j in eleccion){
    matriz62[i,j]=mates[i,j]
  }
}
View(matriz62)
write.csv(matriz62,"matriz62.csv")

#------------------------------------------------------------
datos=read.table("DATOS_PISA2012.txt", header=T, sep="", dec=".")
datos$CNT=as.factor(datos$CNT)
paises=levels(datos$CNT)

mates=read.csv2("mat_norm_mat.csv",sep=";")
colnames(mates)=paises
row.names(mates)=paises
View(mates)
eleccion=c("SGP","BRA","DEU","ROU","ESP","ISR","COL","HKG","FIN","THA","NOR","
    IDN")
matriz12=matrix(nrow=12,ncol=12)
colnames(matriz12)=eleccion
row.names(matriz12)=eleccion

for (i in eleccion){
  for(j in eleccion){
    matriz12[i,j]=mates[i,j]
  }
```

```
}
View(matriz12)
write.csv(matriz12,"matriz12.csv",row.names=F)
```

Código utilizado para obtención de matrices de preferencia de universidad

```
umh=read.csv2("datosumh.csv")
umh=umh[,2:13]
umh
precedencias_umh=matrix(0,ncol=12,nrow=12)

contador=0
for (c in 1:11){
  for (d in (c+1):12){
    contador=0
    for (i in 1:40){
      for (j in 1:40){
        contador=(contador+ifelse(umh[i,c]>umh[j,d],1,0))
      }
    }
    precedencias_umh[c,d]=contador
  }
}
precedencias_umh

contador=0
for (c in 1:11){
  for (d in (c+1):12){
    contador=0
    for (i in 1:40){
      for (j in 1:40){
        contador=(contador+ifelse(umh[i,d]>umh[j,c],1,0))
      }
    }
    precedencias_umh[d,c]=contador
  }
}
write.csv(precedencias_umh,"precedencias_umh.csv",row.names=F)

precedencias_umh_norm=matrix(0,ncol=12,nrow=12)

for(i in 1:12){
  for(j in 1:12){
    if(i == j){
      precedencias_umh_norm[i, j] = 0
    }
    else {
      precedencias_umh_norm[i, j] = precedencias_umh[i, j] / (precedencias_umh[i
        , j] + precedencias_umh[j, i])
    }
  }
}
precedencias_umh_norm
write.csv(precedencias_umh_norm,"precedencias_umh_norm.csv",row.names=F)
```

8.3. Código de programación Lingo para obtención de resultados

Código utilizado para obtener solución del LOP - datos PISA

```
MODEL:

SETS:

        EJE_I/EI1..EI39/;
        EJE_J/EJ1..EJ39/;

        MATRIZ(EJE_I,EJE_J): MAT,X;

ENDSETS

DATA:

        MAT= 0,0.7616,0.9321,0.4076,0.4003,0.7451,0.9308,0.4581,0.4324,0.4906,
0.3515,0.3953,0.5037,0.5284,0.2273,0.6248,0.9386,0.5975,0.5603,0.3244,
0.2436,0.6186,0.5356,0.8721,0.3983,0.5056,0.4769,0.3398,0.7396,0.5883,
0.1834,0.5837,0.6282,0.5492,0.7354,0.9204,0.7549,0.8564,0.5561,0.2384,
0,0.7444,0.1798,0.1737,0.4853,0.706,0.2375,0.2261,0.2253,0.142,0.1729,
0.2865,0.2455,0.0957,0.382,0.7336,0.3517,0.315,0.1512,0.1052,0.3204,
0.245,0.5665,0.2033,0.2218,0.223,0.1415,0.4326,0.2856,0.0735,0.3155,
0.3652,0.257,0.4586,0.6871,0.4814,0.5999,0.2685,0.0679,0.2556,0,0.042,
0.0372,0.2409,0.4288,0.088,0.0795,0.0594,0.0267,0.0442,0.1133,0.0616,
0.0199,0.1776,0.4764,0.1581,0.1255,0.0391,0.0219,0.103,0.061,0.271,0.063,
0.0516,0.0576,0.029,0.1703,0.079,0.0114,0.1225,0.1469,0.0704,0.197,
0.4202,0.2125,0.3407,0.074,0.5924,0.8202,0.958,0,0.491,0.8041,0.9599,
0.534,0.4987,0.5967,0.4347,0.489,0.553,0.6298,0.2707,0.6915,0.9676,
0.6733,0.6294,0.3852,0.2861,0.7141,0.6461,0.9291,0.4565,0.6234,0.5633,
0.423,0.8115,0.6927,0.2172,0.6736,0.6929,0.6654,0.7905,0.9525,0.7983,
0.9093,0.6566,0.5997,0.8263,0.9628,0.509,0,0.8098,0.9656,0.5469,
0.5198,0.5978,0.4491,0.4969,0.5759,0.6335,0.3012,0.7006,0.9714,
0.6809,0.642,0.4057,0.3203,0.7147,0.6461,0.9318,0.4817,0.6215,0.5744,
0.4358,0.8165,0.6921,0.2499,0.6783,0.7048,0.6604,0.7997,0.9581,
0.8103,0.9121,0.6582,0.2549,0.5147,0.7591,0.1959,0.1902,0,0.7218,
0.2505,0.2362,0.2453,0.1589,0.1936,0.2905,0.2652,0.1002,0.3962,
0.7507,0.3667,0.3275,0.1587,0.1061,0.339,0.2686,0.58,0.2106,0.2472,
0.2404,0.1571,0.4487,0.3063,0.0723,0.3341,0.3756,0.2817,0.4682,
0.6966,0.4853,0.6183,0.2877,0.0692,0.294,0.5712,0.0401,0.0344,
0.2782,0,0.0923,0.0866,0.058,0.0224,0.0423,0.1281,0.0618,0.0186,
0.2027,0.5493,0.1773,0.1411,0.0403,0.0213,0.1121,0.0605,0.3149,
0.0692,0.0469,0.0601,0.0254,0.1981,0.083,0.0099,0.1329,0.1686,
0.0701,0.2321,0.4819,0.2506,0.3926,0.0769,0.5419,0.7625,0.912,
0.466,0.4531,0.7495,0.9077,0,0.479,0.537,0.4164,0.4541,0.5463,
0.5654,0.2868,0.6457,0.9157,0.6265,0.5927,0.38,0.3143,0.6409,0.5716,
0.8534,0.4476,0.5474,0.5208,0.4005,0.7401,0.6155,0.2565,0.6129,0.6489,
0.5822,0.7378,0.8989,0.752,0.8438,0.5898,0.5676,0.7739,0.9205,0.5013,
0.4802,0.7638,0.9134,0.521,0,0.5673,0.4563,0.4937,0.571,0.5885,0.3061,
0.6626,0.9239,0.6513,0.6141,0.4031,0.3432,0.6585,0.5963,0.8544,0.465,
0.5749,0.5488,0.4331,0.7504,0.6362,0.2884,0.6307,0.6631,0.6052,0.7458,
0.9021,0.7532,0.8523,0.6151,0.5094,0.7747,0.9406,0.4033,0.4022,0.7547,
0.942,0.463,0.4327,0,0.3343,0.3857,0.4916,0.5446,0.2112,0.6328,0.9497,
0.602,0.5626,0.3144,0.2219,0.6386,0.5554,0.8951,0.3968,0.5254,0.4782,
0.3326,0.7596,0.6102,0.1511,0.6015,0.6363,0.5735,0.7459,0.9328,0.7639,
0.8749,0.5699,0.6485,0.858,0.9733,0.5653,0.5509,0.8411,0.9776,0.5836,
0.5437,0.6657,0,0.5586,0.5896,0.6902,0.3069,0.7349,0.9836,0.7254,
0.6753,0.4291,0.3193,0.7698,0.7108,0.9608,0.4981,0.6947,0.6204,0.4827,
0.8541,0.7537,0.2529,0.7263,0.7344,0.734,0.8255,0.9702,0.828,0.9375,
```

```
0.7219,0.6047,0.8271,0.9558,0.511,0.5031,0.8064,0.9577,0.5459,0.5063,
0.6143,0.4414,0,0.5455,0.6461,0.2724,0.6982,0.9658,0.6819,0.6334,
0.3892,0.2812,0.729,0.6663,0.9349,0.4629,0.6481,0.573,0.4327,0.8206,
0.7114,0.2103,0.6882,0.6975,0.6906,0.7935,0.9508,0.7982,0.9133,
0.6749,0.4963,0.7135,0.8867,0.447,0.4241,0.7095,0.8719,0.4537,0.429,
0.5084,0.4104,0.4545,0,0.5163,0.2489,0.6036,0.8846,0.5949,0.5485,
0.3455,0.2714,0.5876,0.5275,0.7888,0.3879,0.5107,0.4858,0.3826,0.6775,
0.562,0.234,0.5594,0.5935,0.5379,0.6814,0.851,0.6889,0.7977,0.5486,
0.4716,0.7545,0.9384,0.3702,0.3665,0.7348,0.9382,0.4346,0.4115,0.4554,
0.3098,0.3539,0.4837,0,0.2033,0.609,0.9457,0.5756,0.5399,0.2972,0.2192,
0.5961,0.5052,0.8717,0.3803,0.471,0.4464,0.3042,0.7292,0.5635,0.1546,
0.5626,0.6139,0.5184,0.7275,0.9258,0.7509,0.8556,0.5263,0.7727,0.9043,
0.9801,0.7293,0.6988,0.8998,0.9814,0.7132,0.6939,0.7888,0.6931,0.7276,
0.7511,0.7967,0,0.8228,0.9862,0.8299,0.7945,0.6028,0.5424,0.8458,0.8078,
0.9661,0.656,0.7978,0.7612,0.6543,0.8985,0.8345,0.4806,0.8152,0.8265,
0.8154,0.884,0.9789,0.8802,0.9545,0.8226,0.3752,0.618,0.8224,0.3085,
0.2994,0.6038,0.7973,0.3543,0.3374,0.3672,0.2651,0.3018,0.3964,0.391,
0.1772,0,0.818,0.4755,0.4375,0.2487,0.1906,0.466,0.3962,0.6934,0.3063,
0.373,0.3555,0.2578,0.5714,0.4373,0.1474,0.4496,0.4894,0.4079,0.5791,
0.7783,0.5939,0.7123,0.4139,0.0614,0.2664,0.5236,0.0324,0.0286,0.2493,
0.4507,0.0843,0.0761,0.0503,0.0164,0.0342,0.1154,0.0543,0.0138,0.182,
0,0.159,0.1259,0.035,0.0161,0.0987,0.0515,0.2867,0.0612,0.0402,0.0523,
0.0191,0.1754,0.0729,0.0062,0.1204,0.1523,0.061,0.2072,0.4471,0.229,
0.3555,0.0683,0.4025,0.6483,0.8419,0.3267,0.3191,0.6333,0.8227,0.3735,
0.3487,0.398,0.2746,0.3181,0.4051,0.4244,0.1701,0.5245,0.841,0,0.459,
0.2559,0.1832,0.5047,0.4323,0.7364,0.3163,0.4102,0.38,0.2701,0.6095,
0.4766,0.1333,0.4822,0.5166,0.4483,0.613,0.8097,0.6286,0.7446,0.4492,
0.4397,0.685,0.8745,0.3706,0.358,0.6725,0.8589,0.4073,0.3859,0.4374,
0.3247,0.3666,0.4515,0.4601,0.2055,0.5625,0.8741,0.541,0,0.2929,0.2225,
0.5394,0.4677,0.7708,0.352,0.4446,0.4221,0.3112,0.6465,0.5105,0.1766,
0.5151,0.5554,0.4803,0.6517,0.8413,0.6665,0.7777,0.4879,0.6756,0.8488,
0.9609,0.6148,0.5943,0.8413,0.9597,0.62,0.5969,0.6856,0.5709,0.6108,
0.6545,0.7028,0.3972,0.7513,0.965,0.7441,0.7071,0,0.4302,0.764,0.7145,
0.9266,0.5571,0.6987,0.6581,0.5435,0.8388,0.7479,0.3676,0.733,0.7511,
0.7255,0.8248,0.9496,0.8263,0.9167,0.7266,0.7564,0.8948,0.9781,0.7139,
0.6797,0.8939,0.9787,0.6857,0.6568,0.7781,0.6807,0.7188,0.7286,0.7808,
0.4576,0.8094,0.9839,0.8168,0.7775,0.5698,0,0.8354,0.7962,0.9619,0.6119,
0.7865,0.7459,0.6342,0.8905,0.8221,0.4485,0.796,0.8107,0.803,0.8731,
0.9741,0.8668,0.9504,0.8121,0.3814,0.6796,0.897,0.2859,0.2853,0.661,
0.8879,0.3591,0.3415,0.3614,0.2302,0.271,0.4124,0.4039,0.1542,0.534,
0.9013,0.4953,0.4606,0.236,0.1646,0,0.4055,0.7884,0.3141,0.3704,0.3575,
0.2298,0.6385,0.4626,0.1078,0.4756,0.5324,0.4199,0.6484,0.8675,0.6742,
0.7878,0.4287,0.4644,0.755,0.939,0.3539,0.3539,0.7314,0.9395,0.4284,
0.4037,0.4446,0.2892,0.3337,0.4725,0.4948,0.1922,0.6038,0.9485,0.5677,
0.5323,0.2855,0.2038,0.5945,0,0.8778,0.3734,0.464,0.4353,0.2878,0.7317,
0.561,0.1355,0.5595,0.6109,0.5144,0.7275,0.9315,0.7524,0.8592,0.5206,
0.1279,0.4335,0.729,0.0709,0.0682,0.42,0.6851,0.1466,0.1456,0.1049,
0.0392,0.0651,0.2112,0.1283,0.0339,0.3066,0.7133,0.2636,0.2292,0.0734,
0.0381,0.2116,0.1222,0,0.1271,0.0936,0.1186,0.044,0.348,0.1673,0.0142,
0.2143,0.2837,0.1322,0.3915,0.6476,0.4208,0.5472,0.1489,0.6017,0.7967,
0.937,0.5435,0.5183,0.7894,0.9308,0.5524,0.535,0.6032,0.5019,0.5371,
0.6121,0.6197,0.344,0.6937,0.9388,0.6837,0.648,0.4429,0.3881,0.6859,
0.6266,0.8729,0,0.6058,0.5862,0.4749,0.773,0.6618,0.3369,0.6593,0.6954,
0.632,0.7744,0.9205,0.7822,0.87,0.6455,0.4944,0.7782,0.9484,0.3766,
0.3785,0.7528,0.9531,0.4526,0.4251,0.4746,0.3053,0.3519,0.4893,0.529,
0.2022,0.627,0.9598,0.5898,0.5554,0.3013,0.2135,0.6296,0.536,0.9064,
0.3942,0,0.4603,0.3074,0.7623,0.5982,0.1392,0.5924,0.6361,0.5531,
0.7508,0.9483,0.7758,0.8812,0.5532,0.5231,0.777,0.9424,0.4367,0.4256,
0.7596,0.9399,0.4792,0.4512,0.5218,0.3796,0.427,0.5142,0.5536,0.2388,
0.6445,0.9477,0.62,0.5779,0.3419,0.2541,0.6425,0.5647,0.8814,0.4138,
```

```
0.5397,0,0.3666,0.7548,0.6149,0.1949,0.6069,0.643,0.5818,0.7472,0.9257,
0.7621,0.8686,0.5825,0.6602,0.8585,0.971,0.577,0.5642,0.8429,0.9746,
0.5995,0.5669,0.6674,0.5173,0.5673,0.6174,0.6958,0.3457,0.7422,0.9809,
0.7299,0.6888,0.4565,0.3658,0.7702,0.7122,0.956,0.5251,0.6926,0.6334,
0,0.8544,0.7544,0.2967,0.7318,0.7453,0.7297,0.83,0.97,0.8341,0.9356,
0.7222,0.2604,0.5674,0.8297,0.1885,0.1835,0.5513,0.8019,0.2599,0.2496,
0.2404,0.1459,0.1794,0.3225,0.2708,0.1015,0.4286,0.8246,0.3905,0.3535,
0.1612,0.1095,0.3615,0.2683,0.652,0.227,0.2377,0.2452,0.1456,0,0.3205,
0.0719,0.3505,0.4148,0.2807,0.528,0.7721,0.5551,0.6794,0.2974,
0.4117,0.7144,0.921,0.3073,0.3079,0.6937,0.917,0.3845,0.3638,0.3898,
0.2463,0.2886,0.438,0.4365,0.1655,0.5627,0.9271,0.5234,0.4895,0.2521,
0.1779,0.5374,0.439,0.8327,0.3382,0.4018,0.3851,0.2456,0.6795,0,0.1159,
0.5079,0.5669,0.4526,0.6866,0.9021,0.715,0.8209,0.4622,0.8166,0.9265,
0.9886,0.7828,0.7501,0.9277,0.9901,0.7435,0.7116,0.8489,0.7471,0.7897,
0.766,0.8454,0.5194,0.8526,0.9938,0.8667,0.8234,0.6324,0.5515,0.8922,
0.8645,0.9858,0.6631,0.8608,0.8051,0.7033,0.9281,0.8841,0,0.8521,0.8502,
0.876,0.9041,0.9833,0.8922,0.9745,0.8735,0.4163,0.6845,0.8775,0.3264,
0.3217,0.6659,0.8671,0.3871,0.3693,0.3985,0.2737,0.3118,0.4406,0.4374,
0.1848,0.5504,0.8796,0.5178,0.4849,0.267,0.204,0.5244,0.4405,0.7857,
0.3407,0.4076,0.3931,0.2682,0.6495,0.4921,0.1479,0,0.5519,0.4498,0.6552,
0.8554,0.6786,0.7814,0.461,0.3718,0.6348,0.8531,0.3071,0.2952,0.6244,
0.8314,0.3511,0.3369,0.3637,0.2656,0.3025,0.4065,0.3861,0.1735,0.5106,
0.8477,0.4834,0.4446,0.2489,0.1893,0.4676,0.3891,0.7163,0.3046,0.3639,
0.357,0.2547,0.5852,0.4331,0.1498,0.4481,0,0.3993,0.6012,0.8045,0.6187,
0.7334,0.4116,0.4508,0.743,0.9296,0.3346,0.3396,0.7183,0.9299,0.4178,
0.3948,0.4265,0.266,0.3094,0.4621,0.4816,0.1846,0.5921,0.939,0.5517,
0.5197,0.2745,0.197,0.5801,0.4856,0.8678,0.368,0.4469,0.4182,0.2703,
0.7193,0.5474,0.124,0.5502,0.6007,0,0.7156,0.9243,0.743,0.8482,0.5032,
0.2646,0.5414,0.803,0.2095,0.2003,0.5318,0.7679,0.2622,0.2542,0.2541,
0.1745,0.2065,0.3186,0.2725,0.116,0.4209,0.7928,0.387,0.3483,0.1752,
0.1269,0.3516,0.2725,0.6085,0.2256,0.2492,0.2528,0.17,0.472,0.3134,
0.0959,0.3448,0.3988,0.2844,0,0.729,0.5179,0.6496,0.2937,0.0796,0.3129,
0.5798,0.0475,0.0419,0.3034,0.5181,0.1011,0.0979,0.0672,0.0298,0.0492,
0.149,0.0742,0.0211,0.2217,0.5529,0.1903,0.1587,0.0504,0.0259,0.1325,
0.0685,0.3524,0.0795,0.0517,0.0743,0.03,0.2279,0.0979,0.0167,0.1446,
0.1955,0.0757,0.271,0,0.2985,0.4068,0.0905,0.2451,0.5186,0.7875,
0.2017,0.1897,0.5147,0.7494,0.248,0.2468,0.2361,0.172,0.2018,
0.3111,0.2491,0.1198,0.4061,0.771,0.3714,0.3335,0.1737,0.1332,
0.3258,0.2476,0.5792,0.2178,0.2242,0.2379,0.1659,0.4449,0.285,0.1078,
0.3214,0.3813,0.257,0.4821,0.7015,0,0.6239,0.2695,0.1436,0.4001,0.6593,
0.0907,0.0879,0.3817,0.6074,0.1562,0.1477,0.1251,0.0625,0.0867,0.2023,
0.1444,0.0455,0.2877,0.6445,0.2554,0.2223,0.0833,0.0496,0.2122,0.1408,
0.4528,0.13,0.1188,0.1314,0.0644,0.3206,0.1791,0.0255,0.2186,0.2666,
0.1518,0.3504,0.5932,0.3761,0,0.1629,0.4439,0.7315,0.926,0.3434,0.3418,
0.7123,0.9231,0.4102,0.3849,0.4301,0.2781,0.3251,0.4514,0.4737,0.1774,
0.5861,0.9317,0.5508,0.5121,0.2734,0.1879,0.5713,0.4794,0.8511,0.3545,
0.4468,0.4175,0.2778,0.7026,0.5378,0.1265,0.539,0.5884,0.4968,0.7063,
0.9095,0.7305,0.8371,0;


ENDDATA

!Variable binaria;

@FOR(MATRIZ(I,J):@BIN(X(I,J)));

[OBJETIVO] MAX=@SUM(MATRIZ(I,J):MAT(I,J)*X(I,J));

@FOR (EJE_I(I):
        @FOR(EJE_J(J)| J#NE#I:
```

```
                    X(I,J)+X(J,I)=1));

@FOR(EJE_I(I):
          @FOR(EJE_J(J)| J#NE#I:
                    @FOR(EJE_J(T)| T#NE#J #AND# T#NE#I:
                              X(I,J)+X(J,T)+X(T,I)<=2)));


END

END
```

Código utilizado para obtener solución del Modelo forzando datos LOP

```
MODEL:

SETS:

        EJE_I/EI1..EI39/;
        EJE_J/EJ1..EJ39/;
        EJE_K/EJ1..EJ3/;

        MATRIZ(EJE_I,EJE_J): MAT,X,Z,V,W;
        GRUPOS(EJE_K,EJE_I):G;

ENDSETS

DATA:

        MAT= 0,0.7616,0.9321,0.4076,0.4003,0.7451,0.9308,0.4581,0.4324,0.4906,
0.3515,0.3953,0.5037,0.5284,0.2273,0.6248,0.9386,0.5975,0.5603,0.3244,
0.2436,0.6186,0.5356,0.8721,0.3983,0.5056,0.4769,0.3398,0.7396,0.5883,
0.1834,0.5837,0.6282,0.5492,0.7354,0.9204,0.7549,0.8564,0.5561,0.2384,
0,0.7444,0.1798,0.1737,0.4853,0.706,0.2375,0.2261,0.2253,0.142,0.1729,
0.2865,0.2455,0.0957,0.382,0.7336,0.3517,0.315,0.1512,0.1052,0.3204,
0.245,0.5665,0.2033,0.2218,0.223,0.1415,0.4326,0.2856,0.0735,0.3155,
0.3652,0.257,0.4586,0.6871,0.4814,0.5999,0.2685,0.0679,0.2556,0,0.042,
0.0372,0.2409,0.4288,0.088,0.0795,0.0594,0.0267,0.0442,0.1133,0.0616,
0.0199,0.1776,0.4764,0.1581,0.1255,0.0391,0.0219,0.103,0.061,0.271,0.063,
0.0516,0.0576,0.029,0.1703,0.079,0.0114,0.1225,0.1469,0.0704,0.197,
0.4202,0.2125,0.3407,0.074,0.5924,0.8202,0.958,0,0.491,0.8041,0.9599,
0.534,0.4987,0.5967,0.4347,0.489,0.553,0.6298,0.2707,0.6915,0.9676,
0.6733,0.6294,0.3852,0.2861,0.7141,0.6461,0.9291,0.4565,0.6234,0.5633,
0.423,0.8115,0.6927,0.2172,0.6736,0.6929,0.6654,0.7905,0.9525,0.7983,
0.9093,0.6566,0.5997,0.8263,0.9628,0.509,0,0.8098,0.9656,0.5469,
0.5198,0.5978,0.4491,0.4969,0.5759,0.6335,0.3012,0.7006,0.9714,
0.6809,0.642,0.4057,0.3203,0.7147,0.6461,0.9318,0.4817,0.6215,0.5744,
0.4358,0.8165,0.6921,0.2499,0.6783,0.7048,0.6604,0.7997,0.9581,
0.8103,0.9121,0.6582,0.2549,0.5147,0.7591,0.1959,0.1902,0,0.7218,
0.2505,0.2362,0.2453,0.1589,0.1936,0.2905,0.2652,0.1002,0.3962,
0.7507,0.3667,0.3275,0.1587,0.1061,0.339,0.2686,0.58,0.2106,0.2472,
0.2404,0.1571,0.4487,0.3063,0.0723,0.3341,0.3756,0.2817,0.4682,
0.6966,0.4853,0.6183,0.2877,0.0692,0.294,0.5712,0.0401,0.0344,
0.2782,0,0.0923,0.0866,0.058,0.0224,0.0423,0.1281,0.0618,0.0186,
0.2027,0.5493,0.1773,0.1411,0.0403,0.0213,0.1121,0.0605,0.3149,
0.0692,0.0469,0.0601,0.0254,0.1981,0.083,0.0099,0.1329,0.1686,
0.0701,0.2321,0.4819,0.2506,0.3926,0.0769,0.5419,0.7625,0.912,
0.466,0.4531,0.7495,0.9077,0,0.479,0.537,0.4164,0.4541,0.5463,
0.5654,0.2868,0.6457,0.9157,0.6265,0.5927,0.38,0.3143,0.6409,0.5716,
0.8534,0.4476,0.5474,0.5208,0.4005,0.7401,0.6155,0.2565,0.6129,0.6489,
0.5822,0.7378,0.8989,0.752,0.8438,0.5898,0.5676,0.7739,0.9205,0.5013,
0.4802,0.7638,0.9134,0.521,0,0.5673,0.4563,0.4937,0.571,0.5885,0.3061,
0.6626,0.9239,0.6513,0.6141,0.4031,0.3432,0.6585,0.5963,0.8544,0.465,
0.5749,0.5488,0.4331,0.7504,0.6362,0.2884,0.6307,0.6631,0.6052,0.7458,
0.9021,0.7532,0.8523,0.6151,0.5094,0.7747,0.9406,0.4033,0.4022,0.7547,
0.942,0.463,0.4327,0,0.3343,0.3857,0.4916,0.5446,0.2112,0.6328,0.9497,
0.602,0.5626,0.3144,0.2219,0.6386,0.5554,0.8951,0.3968,0.5254,0.4782,
0.3326,0.7596,0.6102,0.1511,0.6015,0.6363,0.5735,0.7459,0.9328,0.7639,
0.8749,0.5699,0.6485,0.858,0.9733,0.5653,0.5509,0.8411,0.9776,0.5836,
0.5437,0.6657,0,0.5586,0.5896,0.6902,0.3069,0.7349,0.9836,0.7254,
0.6753,0.4291,0.3193,0.7698,0.7108,0.9608,0.4981,0.6947,0.6204,0.4827,
0.8541,0.7537,0.2529,0.7263,0.7344,0.734,0.8255,0.9702,0.828,0.9375,
```

```
0.7219,0.6047,0.8271,0.9558,0.511,0.5031,0.8064,0.9577,0.5459,0.5063,
0.6143,0.4414,0,0.5455,0.6461,0.2724,0.6982,0.9658,0.6819,0.6334,
0.3892,0.2812,0.729,0.6663,0.9349,0.4629,0.6481,0.573,0.4327,0.8206,
0.7114,0.2103,0.6882,0.6975,0.6906,0.7935,0.9508,0.7982,0.9133,
0.6749,0.4963,0.7135,0.8867,0.447,0.4241,0.7095,0.8719,0.4537,0.429,
0.5084,0.4104,0.4545,0,0.5163,0.2489,0.6036,0.8846,0.5949,0.5485,
0.3455,0.2714,0.5876,0.5275,0.7888,0.3879,0.5107,0.4858,0.3826,0.6775,
0.562,0.234,0.5594,0.5935,0.5379,0.6814,0.851,0.6889,0.7977,0.5486,
0.4716,0.7545,0.9384,0.3702,0.3665,0.7348,0.9382,0.4346,0.4115,0.4554,
0.3098,0.3539,0.4837,0,0.2033,0.609,0.9457,0.5756,0.5399,0.2972,0.2192,
0.5961,0.5052,0.8717,0.3803,0.471,0.4464,0.3042,0.7292,0.5635,0.1546,
0.5626,0.6139,0.5184,0.7275,0.9258,0.7509,0.8556,0.5263,0.7727,0.9043,
0.9801,0.7293,0.6988,0.8998,0.9814,0.7132,0.6939,0.7888,0.6931,0.7276,
0.7511,0.7967,0,0.8228,0.9862,0.8299,0.7945,0.6028,0.5424,0.8458,0.8078,
0.9661,0.656,0.7978,0.7612,0.6543,0.8985,0.8345,0.4806,0.8152,0.8265,
0.8154,0.884,0.9789,0.8802,0.9545,0.8226,0.3752,0.618,0.8224,0.3085,
0.2994,0.6038,0.7973,0.3543,0.3374,0.3672,0.2651,0.3018,0.3964,0.391,
0.1772,0,0.818,0.4755,0.4375,0.2487,0.1906,0.466,0.3962,0.6934,0.3063,
0.373,0.3555,0.2578,0.5714,0.4373,0.1474,0.4496,0.4894,0.4079,0.5791,
0.7783,0.5939,0.7123,0.4139,0.0614,0.2664,0.5236,0.0324,0.0286,0.2493,
0.4507,0.0843,0.0761,0.0503,0.0164,0.0342,0.1154,0.0543,0.0138,0.182,
0,0.159,0.1259,0.035,0.0161,0.0987,0.0515,0.2867,0.0612,0.0402,0.0523,
0.0191,0.1754,0.0729,0.0062,0.1204,0.1523,0.061,0.2072,0.4471,0.229,
0.3555,0.0683,0.4025,0.6483,0.8419,0.3267,0.3191,0.6333,0.8227,0.3735,
0.3487,0.398,0.2746,0.3181,0.4051,0.4244,0.1701,0.5245,0.841,0,0.459,
0.2559,0.1832,0.5047,0.4323,0.7364,0.3163,0.4102,0.38,0.2701,0.6095,
0.4766,0.1333,0.4822,0.5166,0.4483,0.613,0.8097,0.6286,0.7446,0.4492,
0.4397,0.685,0.8745,0.3706,0.358,0.6725,0.8589,0.4073,0.3859,0.4374,
0.3247,0.3666,0.4515,0.4601,0.2055,0.5625,0.8741,0.541,0,0.2929,0.2225,
0.5394,0.4677,0.7708,0.352,0.4446,0.4221,0.3112,0.6465,0.5105,0.1766,
0.5151,0.5554,0.4803,0.6517,0.8413,0.6665,0.7777,0.4879,0.6756,0.8488,
0.9609,0.6148,0.5943,0.8413,0.9597,0.62,0.5969,0.6856,0.5709,0.6108,
0.6545,0.7028,0.3972,0.7513,0.965,0.7441,0.7071,0,0.4302,0.764,0.7145,
0.9266,0.5571,0.6987,0.6581,0.5435,0.8388,0.7479,0.3676,0.733,0.7511,
0.7255,0.8248,0.9496,0.8263,0.9167,0.7266,0.7564,0.8948,0.9781,0.7139,
0.6797,0.8939,0.9787,0.6857,0.6568,0.7781,0.6807,0.7188,0.7286,0.7808,
0.4576,0.8094,0.9839,0.8168,0.7775,0.5698,0,0.8354,0.7962,0.9619,0.6119,
0.7865,0.7459,0.6342,0.8905,0.8221,0.4485,0.796,0.8107,0.803,0.8731,
0.9741,0.8668,0.9504,0.8121,0.3814,0.6796,0.897,0.2859,0.2853,0.661,
0.8879,0.3591,0.3415,0.3614,0.2302,0.271,0.4124,0.4039,0.1542,0.534,
0.9013,0.4953,0.4606,0.236,0.1646,0,0.4055,0.7884,0.3141,0.3704,0.3575,
0.2298,0.6385,0.4626,0.1078,0.4756,0.5324,0.4199,0.6484,0.8675,0.6742,
0.7878,0.4287,0.4644,0.755,0.939,0.3539,0.3539,0.7314,0.9395,0.4284,
0.4037,0.4446,0.2892,0.3337,0.4725,0.4948,0.1922,0.6038,0.9485,0.5677,
0.5323,0.2855,0.2038,0.5945,0,0.8778,0.3734,0.464,0.4353,0.2878,0.7317,
0.561,0.1355,0.5595,0.6109,0.5144,0.7275,0.9315,0.7524,0.8592,0.5206,
0.1279,0.4335,0.729,0.0709,0.0682,0.42,0.6851,0.1466,0.1456,0.1049,
0.0392,0.0651,0.2112,0.1283,0.0339,0.3066,0.7133,0.2636,0.2292,0.0734,
0.0381,0.2116,0.1222,0,0.1271,0.0936,0.1186,0.044,0.348,0.1673,0.0142,
0.2143,0.2837,0.1322,0.3915,0.6476,0.4208,0.5472,0.1489,0.6017,0.7967,
0.937,0.5435,0.5183,0.7894,0.9308,0.5524,0.535,0.6032,0.5019,0.5371,
0.6121,0.6197,0.344,0.6937,0.9388,0.6837,0.648,0.4429,0.3881,0.6859,
0.6266,0.8729,0,0.6058,0.5862,0.4749,0.773,0.6618,0.3369,0.6593,0.6954,
0.632,0.7744,0.9205,0.7822,0.87,0.6455,0.4944,0.7782,0.9484,0.3766,
0.3785,0.7528,0.9531,0.4526,0.4251,0.4746,0.3053,0.3519,0.4893,0.529,
0.2022,0.627,0.9598,0.5898,0.5554,0.3013,0.2135,0.6296,0.536,0.9064,
0.3942,0,0.4603,0.3074,0.7623,0.5982,0.1392,0.5924,0.6361,0.5531,
0.7508,0.9483,0.7758,0.8812,0.5532,0.5231,0.777,0.9424,0.4367,0.4256,
0.7596,0.9399,0.4792,0.4512,0.5218,0.3796,0.427,0.5142,0.5536,0.2388,
0.6445,0.9477,0.62,0.5779,0.3419,0.2541,0.6425,0.5647,0.8814,0.4138,
```

```
0.5397,0,0.3666,0.7548,0.6149,0.1949,0.6069,0.643,0.5818,0.7472,0.9257,
0.7621,0.8686,0.5825,0.6602,0.8585,0.971,0.577,0.5642,0.8429,0.9746,
0.5995,0.5669,0.6674,0.5173,0.5673,0.6174,0.6958,0.3457,0.7422,0.9809,
0.7299,0.6888,0.4565,0.3658,0.7702,0.7122,0.956,0.5251,0.6926,0.6334,
0,0.8544,0.7544,0.2967,0.7318,0.7453,0.7297,0.83,0.97,0.8341,0.9356,
0.7222,0.2604,0.5674,0.8297,0.1885,0.1835,0.5513,0.8019,0.2599,0.2496,
0.2404,0.1459,0.1794,0.3225,0.2708,0.1015,0.4286,0.8246,0.3905,0.3535,
0.1612,0.1095,0.3615,0.2683,0.652,0.227,0.2377,0.2452,0.1456,0,0.3205,
0.0719,0.3505,0.4148,0.2807,0.528,0.7721,0.5551,0.6794,0.2974,
0.4117,0.7144,0.921,0.3073,0.3079,0.6937,0.917,0.3845,0.3638,0.3898,
0.2463,0.2886,0.438,0.4365,0.1655,0.5627,0.9271,0.5234,0.4895,0.2521,
0.1779,0.5374,0.439,0.8327,0.3382,0.4018,0.3851,0.2456,0.6795,0,0.1159,
0.5079,0.5669,0.4526,0.6866,0.9021,0.715,0.8209,0.4622,0.8166,0.9265,
0.9886,0.7828,0.7501,0.9277,0.9901,0.7435,0.7116,0.8489,0.7471,0.7897,
0.766,0.8454,0.5194,0.8526,0.9938,0.8667,0.8234,0.6324,0.5515,0.8922,
0.8645,0.9858,0.6631,0.8608,0.8051,0.7033,0.9281,0.8841,0,0.8521,0.8502,
0.876,0.9041,0.9833,0.8922,0.9745,0.8735,0.4163,0.6845,0.8775,0.3264,
0.3217,0.6659,0.8671,0.3871,0.3693,0.3985,0.2737,0.3118,0.4406,0.4374,
0.1848,0.5504,0.8796,0.5178,0.4849,0.267,0.204,0.5244,0.4405,0.7857,
0.3407,0.4076,0.3931,0.2682,0.6495,0.4921,0.1479,0,0.5519,0.4498,0.6552,
0.8554,0.6786,0.7814,0.461,0.3718,0.6348,0.8531,0.3071,0.2952,0.6244,
0.8314,0.3511,0.3369,0.3637,0.2656,0.3025,0.4065,0.3861,0.1735,0.5106,
0.8477,0.4834,0.4446,0.2489,0.1893,0.4676,0.3891,0.7163,0.3046,0.3639,
0.357,0.2547,0.5852,0.4331,0.1498,0.4481,0,0.3993,0.6012,0.8045,0.6187,
0.7334,0.4116,0.4508,0.743,0.9296,0.3346,0.3396,0.7183,0.9299,0.4178,
0.3948,0.4265,0.266,0.3094,0.4621,0.4816,0.1846,0.5921,0.939,0.5517,
0.5197,0.2745,0.197,0.5801,0.4856,0.8678,0.368,0.4469,0.4182,0.2703,
0.7193,0.5474,0.124,0.5502,0.6007,0,0.7156,0.9243,0.743,0.8482,0.5032,
0.2646,0.5414,0.803,0.2095,0.2003,0.5318,0.7679,0.2622,0.2542,0.2541,
0.1745,0.2065,0.3186,0.2725,0.116,0.4209,0.7928,0.387,0.3483,0.1752,
0.1269,0.3516,0.2725,0.6085,0.2256,0.2492,0.2528,0.17,0.472,0.3134,
0.0959,0.3448,0.3988,0.2844,0,0.729,0.5179,0.6496,0.2937,0.0796,0.3129,
0.5798,0.0475,0.0419,0.3034,0.5181,0.1011,0.0979,0.0672,0.0298,0.0492,
0.149,0.0742,0.0211,0.2217,0.5529,0.1903,0.1587,0.0504,0.0259,0.1325,
0.0685,0.3524,0.0795,0.0517,0.0743,0.03,0.2279,0.0979,0.0167,0.1446,
0.1955,0.0757,0.271,0,0.2985,0.4068,0.0905,0.2451,0.5186,0.7875,
0.2017,0.1897,0.5147,0.7494,0.248,0.2468,0.2361,0.172,0.2018,
0.3111,0.2491,0.1198,0.4061,0.771,0.3714,0.3335,0.1737,0.1332,
0.3258,0.2476,0.5792,0.2178,0.2242,0.2379,0.1659,0.4449,0.285,0.1078,
0.3214,0.3813,0.257,0.4821,0.7015,0,0.6239,0.2695,0.1436,0.4001,0.6593,
0.0907,0.0879,0.3817,0.6074,0.1562,0.1477,0.1251,0.0625,0.0867,0.2023,
0.1444,0.0455,0.2877,0.6445,0.2554,0.2223,0.0833,0.0496,0.2122,0.1408,
0.4528,0.13,0.1188,0.1314,0.0644,0.3206,0.1791,0.0255,0.2186,0.2666,
0.1518,0.3504,0.5932,0.3761,0,0.1629,0.4439,0.7315,0.926,0.3434,0.3418,
0.7123,0.9231,0.4102,0.3849,0.4301,0.2781,0.3251,0.4514,0.4737,0.1774,
0.5861,0.9317,0.5508,0.5121,0.2734,0.1879,0.5713,0.4794,0.8511,0.3545,
0.4468,0.4175,0.2778,0.7026,0.5378,0.1265,0.539,0.5884,0.4968,0.7063,
0.9095,0.7305,0.8371,0;


ENDDATA
!Variable binaria;

@FOR(MATRIZ(I,J):@BIN(X(I,J)));
@FOR(MATRIZ(I,J):@BIN(Z(I,J)));
@FOR(GRUPOS(K,I):@BIN(G(K,I)));

@FOR(MATRIZ(I,J):@BND(0,W(I,J),0.1));
@FOR(MATRIZ(I,J):@BND(0,V(I,J),0.1));
@BND(0,Y,0.1);
```

```
! ------------------------------ objetivo ------------------------------;

[OBJETIVO] MAX=@SUM(MATRIZ(I,J)|J#NE#I:MAT(I,J)*W(I,J));

! ------------------------------ RESTRICCIONES ------------------------------;
! ------------------------------ 1 ------------------------------;
@FOR (EJE_I(I):
        @FOR(EJE_J(J)| J#NE#I:
                X(I,J)+X(J,I)+Z(I,J)+Z(J,I)=1));

! ------------------------------ 2 ------------------------------;
@FOR(EJE_I(I):
        @FOR(EJE_J(J)| J#NE#I:
                @FOR(EJE_J(T)| T#NE#J #AND# T#NE#I:
                        X(I,J)+X(J,T)+X(T,I)+Z(I,J)+Z(J,T)+Z(T,I)<=2)));

! ------------------------------ 3 ------------------------------;
@SUM(MATRIZ(I,J)|J#NE#I:Z(I,J))=507;

! ------------------------------ 4 ------------------------------;
@SUM(MATRIZ(I,J)|J#NE#I:MAT(I,J)*V(I,J))=1;

! ------------------------------ 5.1 ------------------------------;
@FOR (EJE_I(I):
        @FOR(EJE_J(J)| J#NE#I:
                W(I,J)-Y<=0));
! ------------------------------ 5.2 ------------------------------;
@FOR (EJE_I(I):
        @FOR(EJE_J(J)| J#NE#I:
                W(I,J)-1*X(I,J)<=0));


! ------------------------------ 6.1 ------------------------------;
@FOR (EJE_I(I):
        @FOR(EJE_J(J)| J#NE#I:
                V(I,J)-Y<=0));
! ------------------------------ 6.2 ------------------------------;
@FOR (EJE_I(I):
        @FOR(EJE_J(J)| J#NE#I:
                V(I,J)-1*Z(I,J)<=0));
! ------------------------------ 6.3 ------------------------------;
@FOR (EJE_I(I):
        @FOR(EJE_J(J)| J#NE#I:
                V(I,J)-Y-1*Z(I,J)>=-1));

! ------------------------------ 7.1 ------------------------------;
@FOR (EJE_K(K):
        @SUM(EJE_I(I):G(K,I))=13);
! ------------------------------ 7.2 ------------------------------;
@FOR (EJE_I(I):
        @SUM(EJE_K(K):G(K,I))=1);
! ------------------------------ 7.2 ------------------------------;
@FOR(EJE_I(I):
        @FOR(EJE_J(J)| J#NE#I:
                @FOR(EJE_K(K):
                        @FOR(EJE_K(T)| T#NE#K:
                                X(I,J)+X(J,I)<=2-G(K,I)-G(T,J)))));
```

```
!------------------------------------ FORZAMOS LOP ------------------------------------;
X(31,15)=1;
X(31,21)=1;
X(31,20)=1;
X(31,28)=1;
X(31,25)=1;
X(31,11)=1;
X(31,12)=1;
X(31,5)=1;
X(31,9)=1;
X(31,4)=1;
X(31,8)=1;
X(31,27)=1;
X(1,10)=1;
X(1,13)=1;
X(1,26)=1;
X(1,14)=1;
X(1,23)=1;
X(1,34)=1;
X(1,39)=1;
X(1,19)=1;
X(1,30)=1;
X(1,32)=1;
X(1,18)=1;
X(1,22)=1;
X(33,16)=1;
X(33,29)=1;
X(33,35)=1;
X(33,37)=1;
X(33,6)=1;
X(33,2)=1;
X(33,24)=1;
X(33,38)=1;
X(33,36)=1;
X(33,7)=1;
X(33,17)=1;
X(33,3)=1;
X(15,21)=1;
X(15,20)=1;
X(15,28)=1;
X(15,25)=1;
X(15,11)=1;
X(15,12)=1;
X(15,5)=1;
X(15,9)=1;
X(15,4)=1;
X(15,8)=1;
X(15,27)=1;
X(10,13)=1;
X(10,26)=1;
X(10,14)=1;
X(10,23)=1;
X(10,34)=1;
X(10,39)=1;
X(10,19)=1;
X(10,30)=1;
X(10,32)=1;
X(10,18)=1;
X(10,22)=1;
X(16,29)=1;
```

```
X(16,35)=1;
X(16,37)=1;
X(16,6)=1;
X(16,2)=1;
X(16,24)=1;
X(16,38)=1;
X(16,36)=1;
X(16,7)=1;
X(16,17)=1;
X(16,3)=1;
X(21,20)=1;
X(21,28)=1;
X(21,25)=1;
X(21,11)=1;
X(21,12)=1;
X(21,5)=1;
X(21,9)=1;
X(21,4)=1;
X(21,8)=1;
X(21,27)=1;
X(13,26)=1;
X(13,14)=1;
X(13,23)=1;
X(13,34)=1;
X(13,39)=1;
X(13,19)=1;
X(13,30)=1;
X(13,32)=1;
X(13,18)=1;
X(13,22)=1;
X(29,35)=1;
X(29,37)=1;
X(29,6)=1;
X(29,2)=1;
X(29,24)=1;
X(29,38)=1;
X(29,36)=1;
X(29,7)=1;
X(29,17)=1;
X(29,3)=1;
X(20,28)=1;
X(20,25)=1;
X(20,11)=1;
X(20,12)=1;
X(20,5)=1;
X(20,9)=1;
X(20,4)=1;
X(20,8)=1;
X(20,27)=1;
X(26,14)=1;
X(26,23)=1;
X(26,34)=1;
X(26,39)=1;
X(26,19)=1;
X(26,30)=1;
X(26,32)=1;
X(26,18)=1;
X(26,22)=1;
X(35,37)=1;
X(35,6)=1;
```

```
X(35,2)=1;
X(35,24)=1;
X(35,38)=1;
X(35,36)=1;
X(35,7)=1;
X(35,17)=1;
X(35,3)=1;
X(28,25)=1;
X(28,11)=1;
X(28,12)=1;
X(28,5)=1;
X(28,9)=1;
X(28,4)=1;
X(28,8)=1;
X(28,27)=1;
X(14,23)=1;
X(14,34)=1;
X(14,39)=1;
X(14,19)=1;
X(14,30)=1;
X(14,32)=1;
X(14,18)=1;
X(14,22)=1;
X(37,6)=1;
X(37,2)=1;
X(37,24)=1;
X(37,38)=1;
X(37,36)=1;
X(37,7)=1;
X(37,17)=1;
X(37,3)=1;
X(25,11)=1;
X(25,12)=1;
X(25,5)=1;
X(25,9)=1;
X(25,4)=1;
X(25,8)=1;
X(25,27)=1;
X(23,34)=1;
X(23,39)=1;
X(23,19)=1;
X(23,30)=1;
X(23,32)=1;
X(23,18)=1;
X(23,22)=1;
X(6,2)=1;
X(6,24)=1;
X(6,38)=1;
X(6,36)=1;
X(6,7)=1;
X(6,17)=1;
X(6,3)=1;
X(11,12)=1;
X(11,5)=1;
X(11,9)=1;
X(11,4)=1;
X(11,8)=1;
X(11,27)=1;
X(34,39)=1;
X(34,19)=1;
```

```
X(34,30)=1;
X(34,32)=1;
X(34,18)=1;
X(34,22)=1;
X(2,24)=1;
X(2,38)=1;
X(2,36)=1;
X(2,7)=1;
X(2,17)=1;
X(2,3)=1;
X(12,5)=1;
X(12,9)=1;
X(12,4)=1;
X(12,8)=1;
X(12,27)=1;
X(39,19)=1;
X(39,30)=1;
X(39,32)=1;
X(39,18)=1;
X(39,22)=1;
X(24,38)=1;
X(24,36)=1;
X(24,7)=1;
X(24,17)=1;
X(24,3)=1;
X(5,9)=1;
X(5,4)=1;
X(5,8)=1;
X(5,27)=1;
X(19,30)=1;
X(19,32)=1;
X(19,18)=1;
X(19,22)=1;
X(38,36)=1;
X(38,7)=1;
X(38,17)=1;
X(38,3)=1;
X(9,4)=1;
X(9,8)=1;
X(9,27)=1;
X(30,32)=1;
X(30,18)=1;
X(30,22)=1;
X(36,7)=1;
X(36,17)=1;
X(36,3)=1;
X(4,8)=1;
X(4,27)=1;
X(32,18)=1;
X(32,22)=1;
X(7,17)=1;
X(7,3)=1;
X(8,27)=1;
X(18,22)=1;
X(17,3)=1;
END
```

Nota: Para calcular el modelo normal habrá que quitar las restricciones que fuerzan el LOP

8.4. Código de programación Python para obtención de resultados

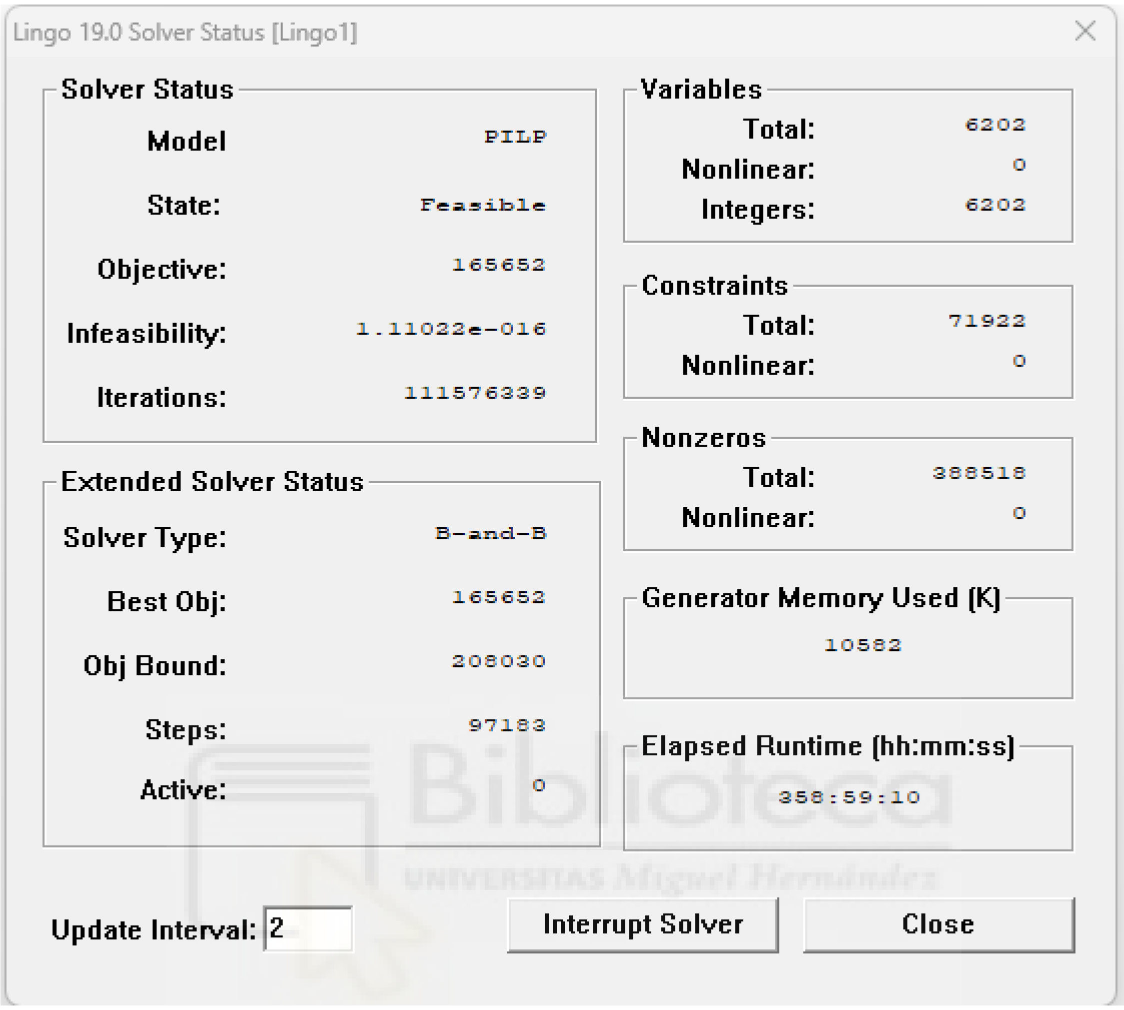
Código utilizado para obtener solución del Modelo

```
#Modelo modificado Mercedes
# 12 individuos , 3 grupos 4, 4, y 4

I=range(12)
K=range(3)
mdl2= Model()
#Variables
  #binarias
x =[[mdl2.binary_var(name="x_i%d_j%d"%(i,j)) for j in I ] for i in I  ]
z =[[mdl2.binary_var(name="z_i%d_j%d"%(i,j)) for j in I ] for i in I  ]
g =[[mdl2.binary_var(name="g_k%d_i%d"%(k,i)) for i in I ] for k in K  ]
#continuas
w =[[mdl2.continuous_var(name="w_i%d_j%d"%(i,j)) for j in I ] for i in I  ]
v =[[mdl2.continuous_var(name="v_i%d_j%d"%(i,j)) for j in I ] for i in I  ]
y = mdl2.continuous_var(name="y")



mdl2.add_constraints(  x[i][j] + x[j][i]   + z[i][j] +z[j][i]  == 1   for i in I
     for j in I if i!=j )
mdl2.add_constraints(  x[i][j] + x[j][k] + x[k][i] +z[i][j] + z[j][k] + z[k][i
   ]<= 2   for i in I for j in I for k in I  if i!=j and j!=k and k!= i    )
mdl2.add_constraint(  mdl2.sum( z[i][j]  for j in I for i in I if i!=j)  == 48
    )


mdl2.add_constraint(  mdl2.sum( (df[i][j])*v[i][j]  for j in I for i in I if i!=
    j)  ==1   )


mdl2.add_constraints(  w[i][j] - y <= 0  for i in I for j in I if i!=j )
mdl2.add_constraints(  w[i][j] - 1*x[i][j] <= 0  for i in I for j in I if i!=j )


mdl2.add_constraints(  v[i][j] - y <= 0  for i in I for j in I if i!=j )
mdl2.add_constraints(  v[i][j] - 1*z[i][j] <= 0  for i in I for j in I if i!=j )
mdl2.add_constraints(  v[i][j] -y - 1*z[i][j] >= -1  for i in I for j in I if i!
   =j )

#NUEVAS RESTRICCIONES
mdl2.add_constraints(mdl2.sum(g[k][i] for i in I) == 4 for k in K)
mdl2.add_constraints(mdl2.sum(g[k][i] for k in K) == 1 for i in I)


mdl2.add_constraints(  x[i][j] + x[j][i] <=2 - g[k][i]  - g[t][j]  for i in I
    for j in I for k in K for t in K if i!=j and k!=t)

#------------------------------------------------------------

mdl2.maximize( mdl2.sum( df[i][j]*w[i][j]   for j in I for i in I if i!=j)    )

mdl2.export_as_lp("p_fraccional.lp")


s = mdl2.solve(log_output=False)

print("Solucion del MODELO FRACCIONAL")
```

```
print(mdl2.solution)
```

8.5. Otro contenido

Tiempo de carga esperado para obtener la solución del Modelo para los datos PISA



Nota: 360 horas equivalen a 15 días esperando la respuesta

Apuntes brutos relevantes tomados durante la creación del Modelo.

(1)

$z_{ij}=0$

↓

z_{12}
z_{23}
z_{24}

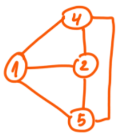
Deberíamos obtener 3 grupos y es uno solo } además, el 5 no lo ordena

(2)

$z_{ij}=0$

↓

z_{12}
z_{14}
z_{15}
z_{24}
z_{25}
z_{45}



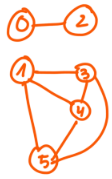
Debería aparecer 2 grupos con 3 individuos en cada uno y no se da el caso

(3)

$z_{ij}=0$

↓

z_{02}
z_{13}
z_{14}
z_{15}
z_{34}
z_{35}
z_{45}



Probando soluciones del modelo nuevo:

6 : 12 =

= CHL, COL, CZE, DEU, ESP, EST.

1) 2 grupos $\begin{cases} \to 3 & 1-2 \quad 2-3 \quad 1-3 \\ \to 3 & 4-5 \quad 5-6 \quad 4-6 \end{cases}$ $\sum_{i\in I}\sum_{j\in I} t_{ij} = 9.$ (1)

2) 3 grupos $\begin{cases} \to 2 & 1-2 \\ \to 2 & 3-4 \\ \to 2 & 5-6 \end{cases}$ $\sum_{i\in I}\sum_{j\in I} t_{ij} = 12$ (2)

3) 2 grupos $\begin{cases} \to 2 & 1-2 \\ \to 4 & 3-4 \quad 4-5 \quad 5-6 \quad 3-5 \quad 3-6 \quad 4-6 \end{cases}$ $\sum_{i\in I}\sum_{j\in I} t_{ij} = 8$ (n)

(1)

```
Solución del MODELO FRACCIONAL
solution for: docplex_model4
objective: 1.46587
status: OPTIMAL_SOLUTION(2)
x_i2_j1=1
x_i4_j1=1
x_i4_j2=1
x_i5_j1=1
x_i5_j2=1
x_i5_j4=1
z_i0_j1=1
z_i0_j2=1
z_i0_j3=1
z_i0_j4=1
z_i0_j5=1
z_i3_j1=1
z_i3_j2=1
z_i3_j4=1
z_i3_j5=1
w_i2_j1=0.449
w_i4_j1=0.449
w_i4_j2=0.449
w_i5_j1=0.449
w_i5_j2=0.449
w_i5_j4=0.449
v_i0_j1=0.449
v_i0_j2=0.449
v_i0_j3=0.449
v_i0_j4=0.449
v_i0_j5=0.449
v_i3_j1=0.449
v_i3_j2=0.449
v_i3_j4=0.449
v_i3_j5=0.449
y=0.449
```

t_{ij} = i y j están en el mismo grupo e i va por delante de j

$$\begin{bmatrix} 0 & 01 & 02 & 03 & 04 & 05 \\ & 0 & 12 & 13 & 14 & 15 \\ & & 0 & 23 & 24 & 25 \\ & & & 0 & 34 & 35 \\ & & & & 0 & 45 \\ & & & & & 0 \end{bmatrix}$$

(n)

```
Solución del MODELO FRACCIONAL
solution for: docplex_model3
objective: 1.0627
status: OPTIMAL_SOLUTION(2)
x_i0_j2=1
x_i1_j5=1
x_i3_j1=1
x_i3_j5=1
x_i4_j1=1
x_i4_j3=1
x_i4_j5=1
z_i0_j1=1
z_i0_j3=1
z_i0_j4=1
z_i0_j5=1
z_i2_j1=1
z_i2_j3=1
z_i2_j4=1
z_i2_j5=1
w_i0_j2=0.294
w_i1_j5=0.294
w_i3_j1=0.294
w_i3_j5=0.294
w_i4_j1=0.294
w_i4_j3=0.294
w_i4_j5=0.294
v_i0_j1=0.294
v_i0_j3=0.294
v_i0_j4=0.294
v_i0_j5=0.294
v_i2_j1=0.294
v_i2_j3=0.294
v_i2_j4=0.294
v_i2_j5=0.294
y=0.294
```

0 − 2
1 − 2
1 − 3
1 − 4
1 − 5
3 − 4
3 − 5
4 − 5

$$\begin{array}{c|cccc|cc} & 0 & 1 & 2 & 3 & 4 & 5 \\ \hline 0 & 0 & 01 & 02 & 03 & 04 & 05 \\ 1 & & & 12 & 13 & 14 & 15 \\ 2 & & & & 23 & 24 & 25 \\ 3 & & & & & 34 & 35 \\ \hline 4 & & & & & & \boxed{45} \\ 5 & & & & & & \end{array}$$

~~ambos~~ 2 4 − 5
1 − 2 ~~0 − 3~~
2 − 3 1 − 3
1 − 2

(2)

```
Solución del MODELO FRACCIONAL
solution for: docplex_model2
objective: 0.321189
status: OPTIMAL_SOLUTION(2)
x_i1_j2=1
x_i3_j2=1
x_i4_j2=1
z_i0_j1=1
z_i0_j2=1
z_i0_j3=1
z_i0_j4=1
z_i0_j5=1
z_i1_j3=1
z_i1_j4=1
z_i1_j5=1
z_i2_j5=1
z_i3_j4=1
z_i3_j5=1
z_i4_j5=1
w_i1_j2=0.176
w_i3_j2=0.176
w_i4_j2=0.176
v_i0_j1=0.176
v_i0_j2=0.176
v_i0_j3=0.176
v_i0_j4=0.176
v_i0_j5=0.176
v_i1_j3=0.176
v_i1_j4=0.176
v_i1_j5=0.176
v_i2_j5=0.176
v_i3_j4=0.176
v_i3_j5=0.176
v_i4_j5=0.176
y=0.176
```

UNIONES

1 − 2
2 − 3
2 − 4

MODELO MONGE

$$\text{Máx} \sum_{i\in I}\sum_{\substack{j\in I\\ i\neq j}} a_{ij}\cdot x_{ij}$$

si $z_{ij}=1$, no pertenecen al mismo grupo, entonces no lo comparamos.

s.a $x_{ij}+x_{ji}+z_{ij}+\cancel{z_{ji}}\leq 1 \quad \forall i,j\in I,\ \cancel{i\neq j}\ i<j$

$x_{ij}+x_{jk}+x_{ki}\leq 2 \quad \forall i,j,k\in I,\ i\neq j,\ j\neq k,\ k\neq i$

$\sum_{i\in I}\sum_{j\in I} z_{ij}=4$ $\quad \cancel{i\neq j}\ i<j$

$\sum_{\substack{i\in I\\ i\neq j}} x_{ij}+x_{ji}\geq 1 \quad \forall j\in I$

$\rightarrow \quad z_{ij}+z_{jk}=2\cdot z_{ki} \quad \forall i,j,k\in I,\ i<j,\ j<k$

· Creo que no hay coherencia entre grupos.

Modelo 39x39

Grupos = 4 → 10 45 indiv; 10 45 indiv; 10 45 indiv; 9 36 indiv

171 indiv con 0 en la diagonal superior

$$\frac{39\cdot 39-39}{2}=741-171=\underline{570}$$

1 - 3	2 - 3	3 - 5
1 - 4	2 - 4	3 - 6

grupo 1	grupo 2
1	3
3	5
4	6
2	

a_{12} a_{13} a_{14}

a_{23} a_{24}

a_{34}

$z_{ij}+z_{jk}=2z_{ki}$

1-2 2-3 3-1

2-3 3-4 4-2

16 indiv

$z_{ij}=0$ i y j van en el mismo grupo

$i\rightarrow j$ $\quad$ 1 - 2

$j\rightarrow k$ $\quad$ 2 - 3

$k\rightarrow i$

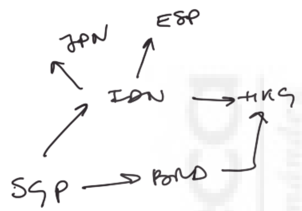
[$z_{ij}=z_{jk}=z_{ki}$

└ Integridad de los grupos.

$z_{ij}+z_{jk}=2\cdot z_{ki} \quad \forall i,j,t$

0 $\quad$ 0 = 0

1 $\quad$ 1 = 2



grupo 1	grupo 2
1 - 2	4 - 5
2 - 3	5 - 6

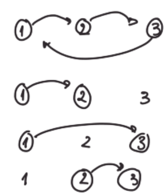
$z_{12}=0$ $\quad z_{45}=0$

$z_{23}=0$ $\quad z_{56}=0$

$$\begin{array}{c} 1\\2\\3\\4\\5\\6 \end{array}\begin{bmatrix} 0 & a_{12} & a_{13} & a_{14} & a_{15} & a_{16}\\ a_{21} & 0 & a_{23} & a_{24} & a_{25} & a_{26}\\ a_{31} & a_{32} & 0 & a_{34} & a_{35} & a_{36}\\ a_{41} & a_{42} & a_{43} & 0 & a_{45} & a_{46}\\ a_{51} & a_{52} & a_{53} & a_{54} & 0 & a_{56}\\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & 0 \end{bmatrix}$$

grupo 1	grupo 2
1 2 3	4 5 6
z_{12} z_{13}	z_{45} z_{56}
z_{23}	z_{46}

6 elementos



NUEVO MODELO

$$\text{Máx} \sum_{i\in I}\sum_{j\in I} a_{ij}\cdot x_{ij}\cdot \frac{1}{\sum_{i\in I}\sum_{j\in I} a_{ij}\cdot z_{ij}}$$

s.a $x_{ij}+x_{ji}+z_{ij}+z_{ji}\leq 1 \quad \forall i,j\in I$

$x_{ij}+x_{jk}+x_{ki}+z_{ij}+z_{jk}+z_{ki}\leq 2 \quad \forall i,j\in I$

$\sum_{i\in I}\sum_{j\in J} z_{ij}=4$

$x_{ij}\in\{0,1\}$

$z_{ij}\in\{0,1\}$

||

$$\text{Máx} \sum_{i\in I}\sum_{j\in I} a_{ij}\cdot w_{ij}$$

$\sum\sum a_{ij}\cdot w_{ij}$

$\sum\sum a_{ij}\cdot x_{ij}\cdot y$

$\hookrightarrow y=\frac{1}{a_{ij}z_{ij}}$

s.a $x_{ij}+x_{ji}+z_{ij}+z_{ji}\leq 1 \quad \forall i,j\in I$

$x_{ij}+x_{jk}+x_{ki}+z_{ij}+z_{jk}+z_{ki}\leq 2 \quad \forall i,j\in I$

$\sum_{i\in I}\sum_{j\in J} z_{ij}=4$

$x_{ij}\in\{0,1\}$

$z_{ij}\in\{0,1\}$

$\sum\sum a_{ij}\cdot v_{ij}=1$

$w_{ij}=x_{ij}\cdot y$

$w_{ij}\leq y$

$v_{ij}=z_{ij}\cdot y$

$v_{ij}\leq y$

$v_{ij}\geq y-M(1-z_{ij})$

$$\text{Máx} \sum_{i\in I}\sum_{\substack{j\in I\\ i\neq j}} c_{ij}\cdot w_{ij}$$

s.a $x_{ij}+x_{ji}+z_{ij}+z_{ji}=1 \quad \forall i,j\in I / i\neq j$

$x_{ij}+x_{jk}+x_{ki}+z_{ij}+z_{jk}+z_{ki}\leq 2 \quad \forall i,j\in I / i\neq j$

$$\sum_{i\in I}\sum_{\substack{j\in I\\ i\neq j}} z_{ij}=9$$

Hasta aquí, tenemos:

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & a_{12} & a_{13} & a_{14} \\ 2 & a_{21} & 0 & a_{23} & a_{24} \\ 3 & a_{31} & a_{32} & 0 & a_{34} \\ 4 & a_{41} & a_{42} & a_{43} & 0 \end{array}$$

MODELO NORMAL

$$\sum_{i\in I}\sum_{j\in I} a_{ij}\cdot x_{ij}$$

$x_{ij}+x_{ji}=1 \quad \forall i,j\in I / i\neq j$

$x_{ij}+x_{jk}+x_{ki}\leq 2 \quad \forall i,j,k\in I / i\neq j, j\neq k, k\neq i$

MODELO 2 [QUEREMOS CREAR GRUPOS]

$$\sum_{i\in I}\sum_{\substack{j\in I\\ i\neq j}} a_{ij}\cdot x_{ij}$$

$x_{ij}+x_{ji}+z_{ij}\leq 1 \quad \forall i,j\in I / i\neq j$

$x_{ij}+x_{jk}+x_{ki}\leq 2$

$$\sum_{i\in I}\sum_{\substack{j\in I\\ i\neq j}} z_{ij}=4$$

$$\sum_{j\in I} x_{ij}+x_{ji}\geq 1 \quad \forall i\in I / i\neq j$$

$z_{ij}=1$ cuando preferimos i a j y no están en el mismo grupo. Si tenemos $4\times 4=16-4=\frac{12}{2}=\boxed{6}$

6 indiv por encima de la matriz.

Si tenemos dos grupos para 4 individuos z_{12} y $z_{34}=0$, entonces hay solo 4 z_{ij} que no sean parejas.

En general no veo este modelo porque → $x_{12}+x_{21}+z_{12}\leq 1$

si x_{12} es que prefiero el 1 al 2
si x_{21} es que prefiero el 2 al 1
si z_{12} es que prefiero el 1 al 2 y ambos no están en el mismo grupo.
No puedo no preferir ninguno.

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & a_{12} & a_{13} & a_{14} \\ 2 & a_{21} & 0 & a_{23} & a_{24} \\ 3 & a_{31} & a_{32} & 0 & a_{34} \\ 4 & a_{41} & a_{42} & a_{43} & 0 \end{array}$$

Grupos
1 – 2
3 – 4

$x_{12}+x_{21}+z_{12}\leq 1$

$x_{21}+x_{12}+z_{21}\leq 1$

$z_{12}+z_{13}+z_{14}+z_{23}+z_{24}+z_{34}=4$

de momento mejoro: $x_{ij}+x_{ji}+z_{ij}+z_{ji}=1$ si pertenecen al mismo grupo z_{ij}

$z_{01}=1$
$z_{02}=1$] $z_{03}=0$
$z_{13}=1$
$z_{23}=1$] $z_{12}=0$

$\boxed{1}-\boxed{2}$
$\boxed{0}-\boxed{3}$

así funciona, pero si aumentamos el número de grupos no

→ 1 – 2
0 – 3

además no establecemos el comportamiento de los grupos

2 individuos

$$\sum_{\substack{j\in I\\ j\neq I}}\left[(1-z_{ij})+(1-z_{ji})\right]=1 \quad \forall i\in I$$

$1-z_{01}+1-z_{02}+\boxed{1-z_{03}}+1-z_{10}+1-z_{20}+1-z_{30}=1$

$1-z_{10}+\boxed{1-z_{12}}+1-z_{13}+1-z_{01}+1-z_{21}+1-z_{31}=1$

$1-z_{20}+1-z_{21}+1-z_{23}+1-z_{02}+\boxed{1-z_{12}}+1-z_{32}=1$

$1-z_{30}+1-z_{31}+1-z_{32}+\boxed{1-z_{03}}+1-z_{13}+1-z_{23}=1$

√

$(1-z_{01})+(1-z_{02})+(1-z_{03})=1$

$(1-z_{12})+(1-z_{13})+(1-z_{10})=1$

$(1-z_{23})+(1-z_{20})+(1-z_{21})=1$

$(1-z_{30})+(1-z_{31})+(1-z_{32})=1$

• por qué?
→ porque quiero que el 0 solo se enlace con uno

6x6
↳ 2 grupos ↗ 3 ↘ 3

0 – 1 0 – 2 1 – 2

3 – 4 3 – 5 4 – 5

Objetivo

· Crear grupos para resolver el problema.

· Planteamiento: es necesario crear grupos? Mejora el LOP? Mejoran los resultados?

· Suponemos que sí

Proceso

1) Individuos

· Comparamos y obtenemos una matriz de comparación

$$\begin{array}{c} 1\\2\\3\\4 \end{array}\begin{bmatrix} 0 & a_{12} & a_{13} & a_{14} \\ a_{21} & 0 & a_{23} & a_{24} \\ a_{31} & a_{32} & 0 & a_{34} \\ a_{41} & a_{42} & a_{43} & 0 \end{bmatrix}$$

LOP normal

$$\text{Máx} \sum_{i\in I}\sum_{\substack{j\in I\\ i\neq j}} a_{ij}\cdot x_{ij}$$

s.a $x_{ij}+x_{ji}=1 \quad \forall i,j\in I, i\neq j$

$x_{ij}+x_{jk}+x_{ki}\leq 2 \quad \forall i,j,k\in I, i\neq j\neq k$

$x_{ij}=\text{bin}$ → 1 i es preferible a j; → 0 contrario

Planteamiento Rafa

· ¿Qué es el LOP?

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4\\ \hline 1 & 0 & a_{12} & a_{13} & a_{14} \\ 2 & a_{21} & 0 & a_{23} & a_{24} \\ 3 & a_{31} & a_{32} & 0 & a_{34} \\ 4 & a_{41} & a_{42} & a_{43} & 0 \end{array}$$

• Modelo que maximiza la diagonal superior.

· $\text{Máx} \sum\sum a_{ij}\cdot x_{ij}$ siendo x_{ij} los valores de i, j que preferimos

· Si creamos una variable x_{ijk} → i es preferible a j ambas pertenecen al

a_{13} preferimos 1 a 3 pero no son del mismo grupo

MODELO MONGE

$$\text{Máx} \sum_{i\in I}\sum_{\substack{j\in I\\ i\neq j}} a_{ij}\cdot x_{ij}$$

si $z_{ij}=1$, no pertenecen al mismo grupo, entonces no lo comparamos.

s.a $x_{ij}+x_{ji}+z_{ij}+\cancel{z_{ji}}\leq 1 \quad \forall i,j\in I, \cancel{i\neq j}\ i<j$

$x_{ij}+x_{jk}+x_{ki}\leq 2 \quad \forall i,j,k\in I, i\neq j, j\neq k, k\neq i$

$$\sum_{i\in I}\sum_{j\in I} z_{ij}=4 \quad \cancel{i\neq j}\ i<j$$

$$\sum_{\substack{i\in I\\ i\neq j}} x_{ij}+x_{ji}\geq 1 \quad \forall j\in I$$

· $x_{ij}=\text{bin}$ → 1 es preferible i a j; → 0 contrario

$z_{ij}=\text{bin}$ → 1 i y j no pertenecen al mismo grupo; → 0 contrario

Problemas

· $z_{ij}+z_{ji}\leq 1$ } si $z_{12}=1$, 1 y 2 no pertenecen al mismo grupo, por lo que 2 y 1 también; es lo mismo.

· Cómo se el número de grupos?

$z_{12}+z_{13}+z_{14}+z_{21}+z_{23}+z_{24}+z_{31}+z_{32}+z_{34}+z_{41}+z_{42}+z_{43}=4$

→ Con esto sabemos que hay 2 grupos

Por qué?

$z_{12}+z_{13}+z_{14}=3$

$z_{41}=1$

1 → 2, 3, 4   4 → 1

	SGP	HKG	ION	BRD	JPN	ESP
SGP	0	13223	35725	142660	20776	131706
HKG	12233	0	30806	121701		105303
ION	223	426	0	91819		9481
BRD	1648	2471	83532	0		44927
JPN	12076					211887
ESP	23438	28193		179037		

Matriz Pisa

$$\begin{array}{c} 1\\2\\3\\4\\5\\6 \end{array}\begin{bmatrix} 0 & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & 0 & a_{23} & a_{24} & a_{25} & a_{26} \\ a_{31} & a_{32} & 0 & a_{34} & a_{35} & a_{36} \\ a_{41} & a_{42} & a_{43} & 0 & a_{45} & a_{46} \\ a_{51} & a_{52} & a_{53} & a_{54} & 0 & a_{57} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & 0 \end{bmatrix}$$

3 grupos

$a_{12}=0$

$a_{34}=0$

$a_{54}=0$

$$\frac{n\cdot n(-1)}{2}=\frac{6\cdot 5}{2}=15$$

$15-3=\boxed{12}$ → nº parejas posibles = 3

2 grupos

a_{12}, a_{13}, a_{14}, a_{23}, a_{24}, a_{34} | a_{56}

$15-6=\boxed{9}$ } como sabemos el nº de parejas?

3 indiv. 3!

nº parejas =

$$=\frac{n!}{m!\,(n-m)!}=\frac{n!}{2!\,(n-2)!}$$

n = indiv/grupo

m = parejas (2)

39x39 → grupo 1 = 10 indiv → 45 parejas; grupo 2 = 10 indiv → 45 parejas; grupo 3 = 10 indiv → 45 parej; grupo 4 = 9 indiv → 36 parej

$$\frac{(39\cdot 39)-39}{2}=\boxed{570}$$

171 parejas

8.6. Enlaces

A continuación, se adjuntan enlaces que dirigen a carpetas de Google Drive, donde se podrán descargar los archivos utilizados para la consecución de este trabjo.

- Acceso a todas las matrices formato CSV utilizadas durante el trabajo: Acceso
- Acceso a código Lingo utilizado durante el documento: Acceso