Universidad Miguel Hernández Elche

Facultad de Ciencias Sociales y Jurídicas

UNIVERSITAS
Miguel Hernández

Subsetting the Linear Ordering Problem

Tutor: Mercedes Landete Ruiz
Cotutor: Juan Francisco Monge

Student: Rafael Antón Moya

June 10, 2024
2023-2024 Year

Contents

1 Introduction

The aim of this paper is to study a proposed modification of the Linear Ordering Problem (LOP), generally known in the field of optimisation by its acronym Linear Ordering Problem. Specifically, it is a combinatorial optimisation problem first studied in 1958 by Chenery and Watanabe. Since its appearance, it has been applied in many fields of study, including archaeology (Glover et al., 1972), economics (Leontief, 2008), graph theory (Charon and Hudry, 2007), translation (Tromble and Eisner, 2009) and mathematical psychology (Kemeny, 1959). It has also seen a wide variety of methods for its solution. Initially, it began to be solved using exact techniques, however, the amount of data it supported in certain investigations was of such a size that it caused a computational time that made it impossible to achieve an answer. This nuance was solved with the use of heuristic techniques, and later by applying the advances made in meteheuristic optimisation. Nowadays, the Memetic Algorithm (MA) and the Iterated Local Search (ILS) proposed by Schiavinotto and Stutzle in 2004, are the algorithms that represent the most avant-garde way of solving the LOP.

Combinatorial optimisation problems are certainly curious, because as we have seen for most of them, the complexity of the problem depends not only on the size of the instance, but also on a series of additional parameters that are generally unknown. The scientific community has been trying for years to understand what these characteristics are in order to guide algorithms to ensure their robustness. In 2004 Schiavinotto and Stutzle came up with certain properties characterising the robustness of LOP instances, the authors defined: disparity, coefficient of variation, skewness and fitness distance correlation.

The modification of the LOP to be addressed in this paper seeks to study a possible improvement of the algorithm by grouping the individuals, so that the most similar subjects are compared and there is no iteration between the disparate ones, thus obtaining internally homogeneous and externally heterogeneous groups, which would guarantee the quality of the response offered.

This paper is the result of the collaboration with the Department of Statistics of the Miguel Hernández University justified through the Collaboration Grant, code 998142, offered by the Ministry of Education. As a result of this collaboration we will deal with solutions of the LOP in real applications in the field of education, relevant modifications in the data of the problem to improve the accuracy of the resolution, the modification of the model mentioned above and a proposal for the application of the model in the system of academic recognition awarded in each academic year by the Miguel Hernández University of Elche to the best student of the promotion.

2 Linear Ordering Problem

2.1 Introduction

As we have seen in the first point, the LOP is a classical combinatorial problem that seeks to find the order that best classifies the individuals of a set. Let us suppose a set of elements to be ordered $V = \{1,...,n\}$ on which we apply a one-to-one pairwise comparison to obtain a square matrix D of order n. In this way, the row i of the matrix D, will contain the respective information to compare the element i with the rest of the elements of the initial set. Thus, each value of within the matrix D, such that d_{ij} will be the unit of goodness that will summarise the extent to which the element i comes before j.

The LOP is powered by the square preference matrix D, where each element of this matrix can be viewed as the *distance* from i to j. The larger the distance of the value d_{ij}, the greater the proof that i must precede j. The number of orders that we can create from a matrix of size n, is $n!$, that is, if we have a set with 5 individuals, we can establish 120 different orders. It is shown in an intuitive and graphic way in the $Table\ 2.2.1$, where it is intended to illustrate the following calculation: $5 * 4 * 3 * 2 * 1 * 1 = 120$.

Table 2.2.1: Graphical demonstration of the number of possible orders

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 Preference matrix

The calculation of the preference matrix D and of the individuals d_{ij} consists of counting the number of times that individual i is better than individual j. Although this is apparently a simple task, it can be complicated when the amount of information of each set of individuals is not of the same size. In such cases we will have to consider the possibility of making modifications to the data, which we will see clearly in the examples presented below, where we will first see the origin of the data and how they are treated in order to move on to their respective preference matrices.

Table 2.2.1: Voting list by judge

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

Table 2.2.2: Preference matrix from Table 2.2.1

	Judge 1	Judge 2	Judge 3	Judge 4	Judge 5
Judge 1	0	3	3	4	4
Judge 2	2	0	2	3	3
Judge 3	2	3	0	3	3
Judge 4	1	2	2	0	2
Judge 5	1	2	2	3	0

In $Table\ \ 2.2.1$, we see the order established by each of the five judges for participants a, b, c, d, and e. In this model, each of them is ranked from best to worst, so that $Judge$ 1 considers individual a to be the best and c the worst, and so on for the rest of the judges. The way to reflect this order in a matrix of preferences, as it is observed in the $Table\ 2.2.2$. As it can be intuited by everything explained up to this point, any position $i\ j$ of the matrix represents how many times i is better than j, in the example, the position $a\ b$ takes the value 3, representing that the individual a is better than b in three occasions. With respect to the problem posed we must have a series of questions in mind for the correct resolution of the same:

1. The maximum value that any position $i\ j$ in the matrix can take is 5, in which case it would mean that individual i would rank ahead of its corresponding individual in all 5 trials. On the other hand, the minimum value of any individual is 0, which would also imply a total consensus, as long as it ranks behind in all 5 judgements. So if we call D the matrix of the $Table\ 2.2.2$, it would be fulfilled that:
   $max\{d_{ij}\} <= 5\ ;\ min\{d_{ij}\} => 0\ \forall\ i, j \in \{1, 2, 3, 4, 5\}$

2. As the list of participants is closed and in this case there are 5 individuals, the value of position ij plus the value of position ji must be equal to 5. Using the above nomenclature, it is satisfied that:
   $d_{ij} + d_{ji} = 5\ \forall\ i, j \in \{1, 2, 3, 4, 5\}|\ i \neq j$

3. The element of the matrix corresponding to the diagonal of the matrix will always be 0, since at no time can we compare the position of an individual with itself. Using the above nomenclature, it is satisfied that:
   $d_{ij} = 0\ \forall\ i, j \in \{1, 2, 3, 4, 5\}|\ i = j$

Table 2.2.3: Average mathematics score by school ID and country

ID	Country	Score
001	Spain	9,30
002	Spain	7,50
003	Spain	8,00
004	Spain	9,00
005	Spain	5,50
001	France	7,10
002	France	5,40
003	France	6,00
004	France	8,20
001	Germany	7,00
002	Germany	6,20
003	Germany	5,30
004	Germany	7,20
001	Brazil	5,90
002	Brazil	7,90
003	Brazil	7,30
004	Brazil	4,50
005	Brazil	6,90

Table 2.2.4: Preference matrix from Table 2.2.3

	Spain	France	Germany	Brazil
Spain	0	15	17	20
France	5	0	9	11
Germany	3	7	0	9
Brazil	5	9	11	0

As we have already mentioned, in this second example we see the creation of a preference matrix from original data where the comparisons do not have the same size, $Table\ 2.2.3$. In the case of Spain and Brazil we have 5 individuals while in France and Germany we have only 4. In this case, as we can see in the preference matrix, the sum of the values $d_{ij} + d_{ji}$ is still a value that we can calculate, and that has a logic. In order to explain it, let us define $V_i\ forall\ i \in \{Spain, French, Germany, Brazil\}$, which will represent the number of individuals of each element, so $V_{Spain} = V_{Brazil} = 5$ and $V_{France} = V_{Germany} = 4$. Once we have defined these concepts, we can already ensure that:

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

As we can see with the established formula, for the comparison $Spain - Germany$, we would have $15 + 5 = 5 * 4 = 20$. Moreover, as in the other example, the two properties mentioned above are fulfilled, now with the nuance that the size is different for the second of them, so we would define:

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

- $max\{d_{ij}\} <= V_i * V_j\ ;\ mind_{ij} => 0$ $(Property\ 2.2.3)$

As has been intuited in the procedure of creating the preference matrix, there is a small disadvantage in the comparisons when the sizes are different. When comparing $Spain - Brazil$, we obtain that $Spain$ is 20 times larger than $Brazil$, while when compared to $Germany$, this number is reduced to 17. Without taking into account the size of the groups of countries, we could think that $Spain$ has more superiority when competing against $Brazil$, than when competing against $Germany$, however, it is only a question of totals, because if we translate it into percentages, we would obtain the following result:

Table 2.2.5: Transformed preference matrix

	Spain	France	Germany	Brazil
Spain	0	0.75	0.85	0.8
France	0.25	0	0.56	0.55
Germany	0.15	0.44	0	0.45
Brazil	0.2	0.45	0.55	0

As is clearly reflected in the example of $Table\ 2.2.5$, actually the greatest superiority that $Spain$ finds is with Germany, being better in 85% of the occasions. The normalisation of the preference matrix data is what we have to do if we want to guarantee a good solution in the LOP, as we have seen, this simple change has transformed the way we look at the preference matrix and we are able to offer a much more accurate conclusion. If we call T the matrix that is reflected in $Table\ 2.2.5$, the transformation that is applied to $Table\ 2.2.4$ to obtain this matrix is:

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

2.3 Problem explanation

From what we have pointed out so far, we know that the LOP is a problem that tries to decipher the best order to classify an indefinite number of individuals. We know that the greater the number of individuals, the greater the complexity of the system to be solved and, therefore, the greater the processing capacity required in our computer systems. We also know that to find a solution to this problem we need a preference matrix, which we have already learned to calculate in the point *2.2 Preference matrix*.

If the preference matrix is reflecting for each pair $i - j$, the number of times that individual i is better than individual j, then, in the correct order, the first individual should have a higher sum total than the rest. To understand this concept, we will use the $Table\ 2.2.5$ on which we will apply permutations in the rows to show different results.

Table 2.3.1: Permuted preference matrix from Table 2.5.5 (1)

	Spain	France	Germany	Brazil
Spain	0	0.8	0.75	0.85
France	0.2	0	0.45	0.55
Germany	0.25	0.55	0	0.56
Brazil	0.15	0.45	0.44	0

Table 2.3.2: Permuted preference matrix from Table 2.5.5 (2)

	Spain	France	Germany	Brazil
Spain	0	0.75	0.8	0.85
France	0.25	0	0.55	0.56
Germany	0.2	0.45	0	0.55
Brazil	0.15	0.44	0.45	0

If we carry out the sum by rows of the matrix of preferences, we will be obtaining the total number of times that an individual is better than the rest, so, if we call P to the permuted matrix represented in the $Table$ 2. 3. 1,, and we calculate the following expression:

- $\sum_{j=Spain}^{Germany} p_{ij} \ \ \forall i \in \{Spain, Brazil, France, Germany\}$

We obtain the following results for each of the four countries:

- $\sum_{j=Spain}^{Germany} p_{Spain\,j} = 2.4$
- $\sum_{j=Spain}^{Germany} p_{Brazil\,j} = 1.2$
- $\sum_{j=Spain}^{Germany} p_{France\,j} = 1.36$
- $\sum_{j=Spain}^{Germany} p_{Germany\,j} = 1.04$

As we can see in these results, the first individual will always be the one that obtains, as we have said, the largest sum of all the comparisons, however, the same will not happen if we keep looking at the rest of the individuals in the matrix. This first case results in such a way, because we are using for the LOP all the individuals of the row, on the contrary, as we go down the ranks, the number of possible individuals to sum is reduced, so that the solution of the system strictly requires the application of the algorithm that we study in this work, and that basically optimises the remaining spaces to obtain the optimal order.

The LOP as we already know is a classical combinatorial problem, so we have to be able to solve what has been captured, in a mathematical programming model. If we look at the $Tables$ 2. 2. 5, 2. 3. 1, *and* 2. 3. 2., we will find that we can study the improvement of the solution (the order) by looking at the individuals above the diagonal. If we add up the elements above the matrix of the tables we get the following: $Table$ 2. 2. 5

- $Table\ 2.2.5 = 3.96$
- $Table\ 2.3.1 = 3.96$
- $Table\ 2.3.2 = 4.06$

The problem of the LOP is precisely to solve the action we have just explained, we start from a preference matrix, in our case the $Table$ 2. 2. 5, and we carry out a series of permutations that make us arrive at the matrix whose upper diagonal is the maximum possible one ($Table$ 2. 3. 2).

2.4 Mathematical model

After what we saw at the end of the previous section, we know that the mathematical model we use to solve the LOP problem, and to obtain the **maximum** possible value in the upper diagonal of the preference matrix, has the obvious objective of maximising, and is expressed as follows:

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

In this model we take into account that M is the indicative for the corresponding preference matrix, so m_{rs}, is the individual of the preference matrix that we have explained in the previous sections, and that reflects the number of times that the individual r is ahead of s. On the other hand, the variable x_{rs}, is a binary variable, which takes the value of 1, if we affirm that the individual r is ahead in the ranking than s. Thus, the constraint (1), states that either r is preferable to s, or vice versa, but at no time can both happen. The condition (2), indicates the logical order when we have three individuals, where if r is better than s, and this better than t, the sum will always be less than or equal to two. To understand this condition we will give an example of an ordered ranking and show how the condition is fulfilled:

Table 2.4.1: Ranking

Position	Ranking
1st	A
2nd	B
3rd	C
4th	D

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

For this order, the value of the variables shown accompanying the $Table\ 2.4.1$ should be as follows:

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

With the proposed example and the value of the variables given, the simplicity and usefulness of the restriction is clearly explained. In this example, we can also see how the restriction (1) acts, since, for example, $x_{AB}+x_{BA}=1$. The last restriction that remains to be resolved is (3), which is more than a restriction, it is an indication that the variable x is a binary variable between 1 and 0.

2.5 LOP Solution

Although the resolution of the LOP was carried out with a computer programme, without which it would be impossible to study the matrices that will be seen later, we must emphasise the correct interpretation of the results in order to be able to obtain solutions.

The method we will use will be the extraction of the software, mostly Lingo, although we will also use Python, and its import into Excel to be treated. The steps that lead to the resolution of any matrix using Lingo and the LOP algorithm will be detailed below.

Table 2.5.1: Preference matrix to solve

	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
```

Starting from the *Table* 2.5.1, the first thing we do is to dump the information into Lingo, the program we will use to solve the model, although as we have mentioned, it can be Python or any other program that generates confidence in what we do. Once we have the matrix, we express the model by means of the corresponding programming. Due to the inconvenience of explaining it in this work, we are not going to go into detail at any point about the programming we use or how it works, but we will attach it in the Annex. When we calculate the result, the solution of the model will be a single value that will reflect the maximum found; however, what we are really interested in is the combination of the variables that give rise to that maximum.

In this case we do not use Excel to save space in the document and to be more concise and practical, although in solutions with more data, we must undoubtedly resort to this software to find solutions quickly. As we can see, we obtain 1 and 0, but since we know what this result means after the previous sections, we are able to interpret it. As we see, x_{12}, x_{13}, x_{14}, x_{23}, x_{24}, $x_{34} = 1$, which means the following result:

Table 2.5.2: Result

Position	Ranking
1st	1
2nd	2
3rd	3
4th	4

Up to this point, all the indications have been given to be able to obtain the solution of the LOP on any dataset, being now able to start from a raw database and step by step obtain the necessary elements to solve the optimal order. From this point onwards, the complete realisation of the LOP on PISA data will be presented, to later explain the possible improvements to the model studied so far.

3 LOP Resolution for PISA report

3.1 Problem context

The idea of solving the LOP problem with PISA data is the starting point of the research carried out, and is discussed later in this paper. As a step towards the complete understanding of the functioning of the LOP problem, the resolution of the PISA database was proposed, as it had been used by the tutor Mercedes Landete and the co-tutor Juan Francisco Monge in the article published in 2019 entitled “A linear ordering problem of sets”, added in the Bibliography of this paper. It should be noted that in this publication we can only see the result of the application of the LOP to this database, so it has served as a learning guide. In order to offer a much wider range of knowledge regarding the resolution of this type of situation, the following points detail exactly all the steps followed.

3.2 PISA report

First, we will describe what the Programme for International Student Assessment (PISA) report is all about. It reflects the OECD’s global survey, which measures the academic performance of 15-year-old students in three different domains: mathematics, science and reading. The exams are promoted in each country at the institutional level, i.e. the state is in charge of conducting them. To guarantee a representative sample of the data, the state requires 4,500 to 10,000 students per country to take the exams, although in some cases, such as Mexico or the United States, the number rises to more than 30,000 students.

The data file that we have used to solve the problem is a txt type divided into 6 columns:

- CNT: Indicates the country from which the study is collected. It is a nominal qualitative variable.
- UNIT: Records the school number being evaluated among all countries. It is an ordinal qualitative variable.
- SCHOOLID: Records the school number being evaluated within the same country, starting from one for each new country. It is an ordinal qualitative variable.
- PVMATH: Records the score obtained by that school in the mathematics exam. It is a continuous quantitative variable.
- PVREAD: Records the score obtained by that school in the reading exam. It is a continuous quantitative variable.
- PVSCIE: Records the score obtained by that school in the science exam. It is a continuous quantitative variable.

Table 3.2.1 below shows a visualization that seeks to reflect the format of this file. As can be seen, and taking into account the previous description of the variables, the last value in the UNIT column, which corresponds to USA (in alphabetical order in the CNT variable), shows the total number of schools used for the study 13,494.

Table 3.2.1: Preference matrix to solve

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 Problem target

Once we know which is the database we are going to treat, the variables that compose it and the type of each one, we only need to know what objective we want to achieve in order to establish the steps to follow. In this case the objective of the application of the LOP is to obtain the order of countries ordered from best to worst according to the score in the mathematics exam in order to show the differences with respect to other types of sorting criteria.

3.4 Transforming database to preference matrix

To apply all the required transformations to the data that facilitate the resolution of the problem, we will import the txt database in our statistical computing software, R-Studio, hereinafter R. Before continuing, it should be noted that all the programming related to what is discussed in this section is reflected in the annex, being able to be consulted for a better understanding. In R we will eliminate the columns that we do not need, these are $UNIT$, $SCHOOLID$, $PVREAD$, and $PVSCIE$, so only CNT, and $PVMATH$ will remain.

To obtain the preference matrix, we will separate the original database into as many tables as there are countries in the database, then we will create a loop that can count how many times one country is better than another, comparing individual to individual in each of the 39 countries.

The following tables attempt to simulate the step from entering the software to exiting in the form of a preference matrix. In the $Table\ 3.4.1$,, the first column is a representation of the display in the software, and is indicative of the size of each table, which is why this column does not have a header. With the $Tables\ 3.4.1\ and\ 3.4.2$, we seek to represent the separation of the countries in different tables, as there are 39 different countries we will represent only the first two to avoid saturating the document.

In $Table\ 3.4.3$ we see the matrix of preferences obtained from the comparison of all the tables, as we can see, in the case of the $Australia - Bulgaria$ comparison, we have that $Australia$ is superior to $Bulgaria$ in 108,433 occasions, while on the contrary we only add up to 37.257. With the representation of the $Tables\ 2.4.1\ and\ 2.4.2$, we can prove the veracity of $Propiedad\ 2.2.1$, since if we call D the preference matrix of $Table\ 3.4.3$, then $D_{AUS-BGR}+D_{BGR-AUS}$ = 145.700, which is the equivalent of making all possible combinations, i.e. $775 * 188 = 145,700$.

Table 3.4.1: Representative table of mathematics scores in Australia

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

Table 3.4.2: Representative table of mathematics scores in Bulgaria

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

Table 3.4.3: Preference matrix obtained

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

As can be seen in the tables shown above, the number of individuals in each country is not the same, so following the explanations in the previous sections, the procedure to follow in this case is to normalize the preference matrix to avoid problems caused by an excessively large or small size in the comparisons. The new, now normalized, preference matrix is the one reflected in the$Table\ 3.4.4$

Table 3.4.4: Normalized preference matrix

	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

We can check the adequacy of the executed calculation, first of all, because the sum of all the pairs fulfills the following: $N_{ij} + Nji = 1$, assuming that N is the matrix of the $Table$ 3.4.4. Furthermore, we see that there are significant differences between the two tables, if we take USA as an example, we see how $D_{USA-AUS} = 55388$, and $D_{USA-BGR} = 22142$, however, when these are normalized we have $N_{USA-AUS} = 0.4439$, and $N_{USA-BGR} = 0.7315$.

With this matrix we can finish the data processing stage in R-Studio. Each time we process the algorithm that compares the different tables to find the preference matrix we have just studied, the software takes about 30 minutes for a data size equivalent to the given one. For this reason we export the obtained matrix to a CSV file that can be further processed.

3.5 Optimization model solution

In this section we will work in another software, LINGO/LINDO, commonly used in mathematics for the resolution of linear systems. The complete code can be consulted in the appendix, although it is exactly the same as the one used in $Block$ 2.5 modifying the data matrix used and its size.

Changing these two details, the calculation of the problem rises to the resolution of 1,521 variables, and 55.576 restrictions, whose resolution time in this program is insignificant. Evidently, because of the size of the problem, it is impossible to interpret the solution without external help, for this we will resort to Excel, where with a simple formula we will obtain the solution.

The $Table$ 3.5.1 represents the solution obtained that reflects the order generated by the LOP model to rank the countries from best to worst. Although the table contains only two variables $RANKING$, and $COUNTRY$, it is necessary to define what each one reflects, this is represented in three columns to save space.

Table 3.5.1: LOP solution with PISA data

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 Modifying the Linear Ordering Problem

4.1 Target

Once we know how to proceed correctly with the calculation and obtaining the solution of the LOP, we can start thinking about possible modifications of the problem that give rise to a much more robust solution. In our case, we suggest improving the model by comparing individuals in subgroups, obviously of a smaller size than the initial one. The objective is to find internally homogeneous and externally heterogeneous groups, with the idea of establishing comparisons only with the most similar individuals, and to avoid comparing between very mismatched countries that simply create a greater weight, often unbalanced in the preference matrix.

In order to clarify the purpose of this paper, we will again use the PISA database, which has been explained in detail in the previous section, including the transformation of the data. We start from the matrix corresponding to the $Table\ 3.4.4$. Once we know this, we can specify that with the application of such a model we want to get 3 groups of 13 individuals each with the above mentioned characteristics, and we also want to internally sort the groups.

4.2 Model

To understand the model, we have to keep in mind the indications presented in the previous section; in the case of the original LOP model, we seek to study all the values that lie above the upper diagonal, however, now we have to study only the comparisons between individuals belonging to the same group.

In addition, this model is modified according to the objective we are looking for, which is why we have specified in the previous paragraphs the number of groups and individuals we want to achieve. It will be observed in the model the use of the letters A for the constraint (6), and B for (7), being these the only ones that find variations. Before presenting the new model, we collect in the following points the explanation of the indexes and variables that interact in this one and that will be useful to understand the mathematical mechanics:

- P: Set of individuals to be compared r,s $\in$ {1,2,...,39}.
- K: Set of groups to be carried out k,t $\in$ {1,2,3}.
- x_{rs}: Binary variable, takes a value of 1 when we prefer individual r to s, and both are in the same group.
- z_{rs}: Binary variable, takes a value of 1 when we prefer individual r to s, and both are not in the same group.
- g_{kr}: Binary variable, takes value of 1 when we assign individual r to group k.

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

As we have mentioned, we make use of both z and x to study those individuals that are inside the groups and look for that combination whose proportion of values inside and outside all the groups is the maximum possible. Thus, if we look at the objective function, we maximize the values that we are going to introduce in the groups, and minimize those that are outside; this is the sense of the objective function, and that allows us to establish internally homogeneous groups, which are therefore externally heterogeneous.

With the constraint (4), we are indicating the preference regarding the comparison of two individuals r, and s of the matrix, whether or not they are within the same group. The constraint (5) seeks the same goal of establishing preferences now with three individuals r, s, and t, and that these comply with the logical order, again interacting with individuals from the same or different groups. The condition (6), characterized as we have commented above by the use of the letter A, seeks to count how many individuals do not belong to the same group, this calculation we obtain from the following reasoning explained with the *Table* 4.2.1:

Table 4.2.1: LOP solution with PISA data

	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

In the hypothetical case of establishing the following groups $G1 = \{1, 2\},\ y\ G2 = \{3, 4\}$, that is, individuals 1 and 2 belong to group 1, and individuals 3 and 4 to group 2, the only comparisons that would be established would be those of the same group, leaving out the values m_{13},m_{14},m_{23}, and m_{24}, and giving rise to the restriction:

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

This value can be calculated by hand as we have done, however, it would be very tedious to repeat it for the matrix of order 39 that we have to study, so we can obtain the value with a calculation. We advance that the value of A for the initial assumptions we have taken equals 507. To understand the formula described below it will be necessary to know that n_i refers to the size of the group i, and N is the order of the preference matrix. We note that we use the summation for the case in which the number of individuals per group is not always the same, a case which we could apply to the model very simply. Thus we have the following calculation:

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

The restriction (7), is another restriction that varies according to the objective, in this case the calculation is reduced to the logic, giving to B the value of 13 for this case, because it is the number of individuals of each group, in case we wanted that the groups did not have the same number of individuals we could force it manually restricting for each group, the number of individuals that compose it. The restriction (8), indicates that each individual must belong to only one group, that is to say, we cannot leave any individual without a group, nor that the same individual belongs to more than one, because the first would be a problem and the second an inconsistency. Finally, the restriction (9) serves to associate the individuals to be compared for being in the same group; x, with the group to which each individual belongs; g, and that there are no divergences between both, to simplify the clarification, that the individuals i, j that are compared both belong to the same group.

For the realization of this model, many brutes have been studied before, which have not given a good result, but have helped to obtain the solution that we will discuss later. Therefore, these will be included in the appendix so that they can serve as a guide to which paths have no solution, and serve as a help and idea for future work and research.

If we have looked at the model we want to study, we will realize that we are not dealing with a linear model, therefore, we will have to apply a series of modifications to it so that it can be perfectly studied by the available software. With these modifications, we observe an important transformation, to which we need to add a series of variables; y, v, and w that allow its linearization. In this work, we will not explain the procedures followed to linearize the model, we will simply offer the final result.

If we have looked at the model we want to study, we will realize that we are not dealing with a linear model, therefore, we will have to apply a series of modifications to it so that it can be perfectly studied by the available software. With these modifications, we observe an important transformation, to which we need to add a series of variables; y, v, and w that allow its linearization. In this work, we will not explain the procedures followed to linearize the model, we will simply offer the final result.

$$\max \quad \sum_{r\in P}\sum_{s\in P: r\neq s} m_{rs}w_{rs}$$

$$s.a \quad x_{rs}+x_{sr}+z_{rs}+z_{sr}=1 \qquad \forall\, r,s\in P: r<s \tag{13}$$

$$x_{rs}+x_{st}+x_{tr}+z_{rs}+z_{st}+z_{tr}\leq 1 \qquad \forall\, r,s,t\in P: r,s,t \text{ pwd} \tag{14}$$

$$\sum_{r\in P}\sum_{s\in P: r\neq s} z_{rs}=A \tag{15}$$

$$\tag{16}$$

$$\sum_{r\in P}\sum_{s\in P: r\neq s} m_{rs}v_{rs}=1 \tag{17}$$

$$w_{rs}-y\leq 0 \qquad \forall\, r,s\in P: r<s \tag{18}$$

$$w_{rs}-Mx_{rs}\leq 0 \qquad \forall\, r,s\in P: r<s \tag{19}$$

$$\tag{20}$$

$$v_{rs}-y\leq 0 \qquad \forall\, r,s\in P: r<s \tag{21}$$

$$v_{rs}-Mz_{rs}\leq 0 \qquad \forall\, r,s\in P: r<s \tag{22}$$

$$v_{rs}-y-Mz_{rs}\geq -M \qquad \forall\, r,s\in P: r<s \tag{23}$$

$$\sum_{r\in P} g_{kr}=B \qquad \forall\, k\in K \tag{24}$$

$$\sum_{k\in K} g_{kr}=1 \qquad \forall\, r\in P \tag{25}$$

$$x_{rs}+x_{sr}\leq 2-g_{kr}+g_{ts} \qquad \forall\, r,s\in P; k,t\in K \tag{26}$$

$$x_{rs}\in\{0,1\} \qquad \forall\, r,s\in P: r\neq s \tag{27}$$

$$z_{rs}\in\{0,1\} \qquad \forall\, r,s\in P: r\neq s \tag{28}$$

$$g_{kr}\in\{0,1\} \qquad \forall\, r\in P, k\in K \tag{29}$$

4.3 Application to PISA data

Before commenting on the results of the application of the model presented on the PISA data, it is worth mentioning that this is not the most optimal possible solution, since the model loading time extends indefinitely, in fact, the longest processing that has been performed has exceeded 358 hours, a capture of this load is attached in the Annex, i.e., about 15 days loading to obtain a solution.

Due to the obstacle encountered, we have decided to limit the processing time of the program to a few hours, which has resulted in the following solution; shown in the *Table* 4.3.1.

Table 3.5.1: Model solution with PISA data

Ranking	Group 1	Ranking	Group 2	Ranking	Group 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

We have observed that the longer the loading time, the better the solution we obtain, at least in the sense of approximation to the initial objective. That is, the more homogeneous the groups are internally, and the more heterogeneous externally, however, this matrix is not useful for comparing the algorithms used.

Another very important point to comment on the algorithm is the current importance of this type of models. During the past month of May 2024, in the congress *International Symposium on Combinatorial Optimization*, the paper ***Cycle of Clusters*** was presented, added to the bibliography of the work, a research whose objective is to generate clusters of individuals (as it is understood, internally homogeneous and heterogeneous with respect to the others), and then to order them among themselves. As we can see, the scope of this research is very similar to ours, adding that we also sort the clusters internally, thus adding a point of complexity and sophistication to the result.

Thanks to this research, we get ideas on how to shorten the processing time of the model; for example, by means of inequality constraints, whether triangular, partitioning, or subtraversion, which are detailed in the research paper. Thanks to the attendance to this congress of the tutor Mercedes, we can understand how relevant is the model we are creating, and how current is this topic in Operations Research.

4.4 Algorithm comparison

Due to the processing time required by the new model formulated to solve any large matrix, we have trimmed the PISA data matrix to 12 individuals in order to test and compare how our model performs with respect to the LOP.

In order to have effective comparisons, we must compare what solution the LOP offers us for that matrix, what solution our model offers, and how good the solution is if we force the LOP result into our model, which is equivalent to splitting the LOP solution into as many groups as we have tested in our model. In the following $Table\ 4.4.1$, we will be able to consult the result by sorting the individuals with the three criteria mentioned above. We also emphasize that, in order to save space in the document, we will attach the Lingo code, together with the matrices used in the Annex of the work.

Table 3.5.1: New model solution using PISA data

LOP ranking		Model ranking		LOP applicating Model	
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

For these results it should be noted that the model we have programmed considers 3 groups with 4 individuals for each one, so the last two columns of the $Table\ 4.4.1$, $Classification\ Model$, and *LOP applied to the* $Model$, are divided into groups every 4 individuals, this we have not represented because it is not relevant for the analysis of the results.

As we can see, there are differences between the result offered by the LOP and the one offered by the Model, changing the order of the countries $ESP - DEU - ROU - COL - THA$. What we seek to find now is whether the model we have created improves the solution of the LOP, for this we will focus on the objectives of the models, which are the following:

- LOP: 52,11
- LOP applicated to the Model: 1,17989
- Model: 1,276524

Having seen the results, and knowing that the objective of the Model is to maximize the objective, we can conclude that there are differences between the LOP (compare all possible individuals) and our model (compare only those individuals that we classify in the same group), and that this difference is advantageous on the one hand, in terms of the result obtained, and on the other hand, in terms of the diversity of the response. Specifically, the Model solution improves the LOP fit by 8.2%, so there is a significant increase. We emphasize in addition to these three points mentioned above, that the LOP response cannot be compared with the other two because it has a completely different objective, but it is useful as information reflecting the three solutions.

5 Application proposal for Universidad Miguel Hernández

In this section we will describe the proposal we intend to make for the Miguel Hernández University, in which we will detail how the Model we have invented can be applied to offer an alternative to improve a classification system that uses a not very robust criterion. The focus of the proposal is none other than to give value to the work done by highlighting that it is not simply a theoretical advance in the field of Operations Research, but also has relevant practical implications, and can begin to be applied in internal systems of the University.

5.1 Proposal Context

The Miguel Hernández University annually honors the best students of each university degree, for this recognition, the selection criterion is based on the overall average of the total number of subjects taken during the four years of study, and the student with the highest average is awarded for his or her work. While we were working on the creation of the algorithm, the idea arose internally to offer another criterion for the student ranking system.

5.2 Target

With this proposal we seek to demonstrate that the classification criterion that uses the mean to reflect the order of the individuals is not the most robust, especially in this type of cases where we have a large amount of data to work with, and where there are also cases in which the graduate student has taken a different number of subjects than his or her classmates because they have been validated.

With this demonstration, we intend to be able to work together with internal departments of the Miguel Hernández University to help choose the best file of each promotion in a more sensible way, and taking into account the information of the total number of subjects taken.

5.3 Data & Model

Although we would have liked to be able to work with real data to make the application on an existing case, it has not been possible to have access to these in any way, so we have had to simulate the 40 grades of 12 students to be able to work. This simulation has been carried out in *Excel*, and the data have been worked in $R - Studio$ to then pass them to *Lingo* in order to obtain results. The data used can be consulted in the Annex in the description.

The data have been created randomly using values between 5 and 10, this is because no student can be considered for the extraordinary award if he/she has not finished the degree, and for this it is required to have passed all the subjects. Regarding the simulation of the data, the 12 observations have been very centered around 8.5, so the results will not be totally faithful to the reality of the universities.

Regarding our model, it is important to note that we are going to work as we have done during the previous applications, with 3 groups of 4 individuals each, so that in the *Table* 5.4.1, it will be necessary to take into account that there is a separation of groups every 4 individuals.

5.4 Solution

For the database we have chosen for these 12 individuals, we do not see any noticeable changes in the top positions. This is due firstly to the fact that the database does not faithfully represent the distribution of the students' grades, and there is no way to find such a distribution because we do not have any original data to guide us. Secondly, and removing the effect of the database, if there were no changes or modifications in the ranking order, it could be a mere coincidence in the data, since the ranking criteria are completely different.

As we can appreciate with the following results, evident in $Table\ 5.4.1$, just as there are changes in the lower positions, this could have occurred in the higher ones, obtaining a totally different result of the extraordinary prizes.

To understand the $Table\ 5.4.1$, we emphasize that when we indicate, $A.X$, being X a value between one and twelve, we refer to $A.X$ the number that identifies it in order to keep track of its variation as we apply the different results.

Table 3.5.1: Solution for new model with PISA data

Average ranking	LOP ranking	Model ranking
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

As mentioned before, the lowest positions have the most changes, but this is due to a mere coincidence, since the data have been calculated in a random way, and not following any kind of distribution, so the only way to test the change that such an algorithm would produce in practice would be to obtain the real data.

6 Conclusion

In this last section, we will discuss the conclusions we have drawn after many months of work on this algorithm, as well as the assumptions we can make for the future.

First of all, it should be noted that thanks to this research, there is an advance in the classification of individuals and an improvement over the original LOP model, because now we are dividing into internally homogeneous and externally heterogeneous groups, which in turn are classified internally from largest to smallest. We know that there is an improvement, because firstly we obtain a different solution to the LOP, i.e., it is not the same to classify by comparing all individuals with each other, than to do so by comparing only those that are part of the same group and, secondly, because doing so is, as we have seen in the paper, better.

On the other hand, it is important to comment as a negative point about the algorithm that we have made, the loading time required to find solutions. It is very counterproductive to have to wait the amount of hours seen in the document to be able to conclude and advance with the investigations, this fact has slowed down the advances. This can occur because, one, in our model (for a matter of time), we cannot add unequal constraints that speed up the search for the objective, and with which we would advance many hours of work, and/or two, because the software we use is not really powerful for the calculation of problems of this dimension, which consequently require a really high number of constraints.

Supporting this negative point, it would be very interesting to continue this research by adding the inequality constraints mentioned above to be able to study much larger matrices with much less loading time, it would be at this moment when the Model would achieve its maximum potential, reaching surprising results for many fields.

Rambling on about these scopes, we could create ranks of all kinds, with unrestricted group sizes to study other types of information that require a larger volume of data, for example, cities in a country, restaurants in a province, sports rankings, and many other options that would appear as the problem is explored further.

Another point that can be further worked on in this Model is the assignment of individuals per group. Currently, we are the ones who must manually indicate the equal distribution (or not) of individuals in each group, however, it could be very interesting to obtain the optimal value by letting the algorithm choose the distribution it requires. This problem has been hovering in our heads during all these months, and now more than ever before it seems very feasible thanks to the high quality of the model that we have generated, however, and as we have been commenting, for a question of temporary budget we have had to abandon the task.

Just as we want to add in the Annex those models and tests that have not been successful, we also consider it appropriate to comment, as we have done in the previous paragraph, on those points that have been left pending, with the aim of returning as soon as possible to the document and being able to continue where we left off without losing the ideas we have, as well as being able to help future researchers who want to be enlightened with new ideas.

7 Bibliography

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 Appendix

8.1 Databases used in the thesis

Table 8.8.1: Raw PISA database

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

Access to full table clicking on the following link: Link

Access to csv file clicking on the following link: Link

Table 8.8.1: Raw university database (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

Table 8.8.1: Raw university database (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 R-Studio code used to get the matrices

Code used to obtain PISA preference matrices

```
#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)
```

```r
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)
```

Code used to obtain university preference matrices

```
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 LINGO code used to get the optimals

Code used to obtain LOP PISA data optimals

```
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
```

Code used to obtain Model optimal using former PISA solution

```
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
```

Note: In order to calculate the Model without restricting the PISA solution, we will need to delete the addtional restrictions

8.4 R-Studio code used to get the optimals

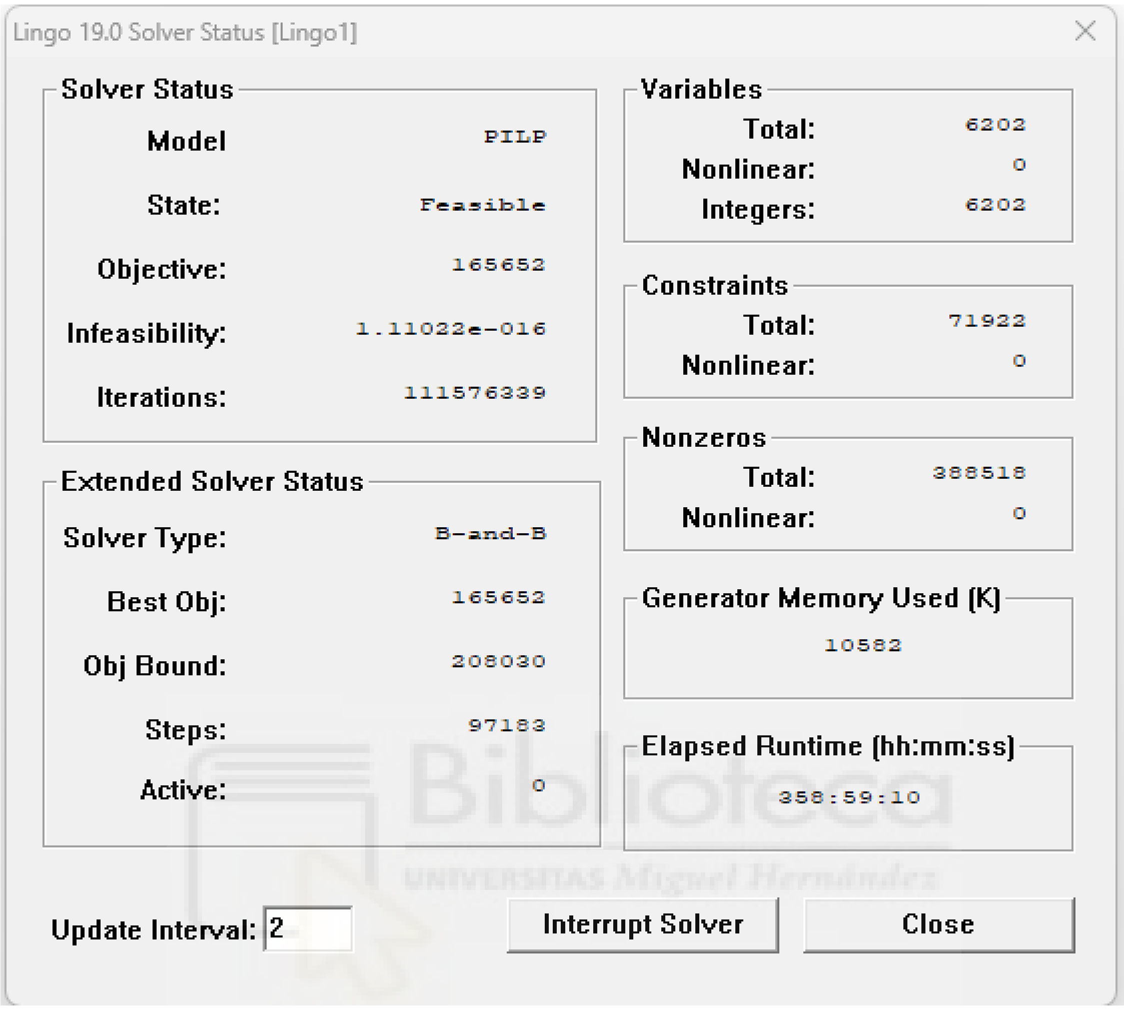
Code used to obtain Model optimals

```
#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")
```

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

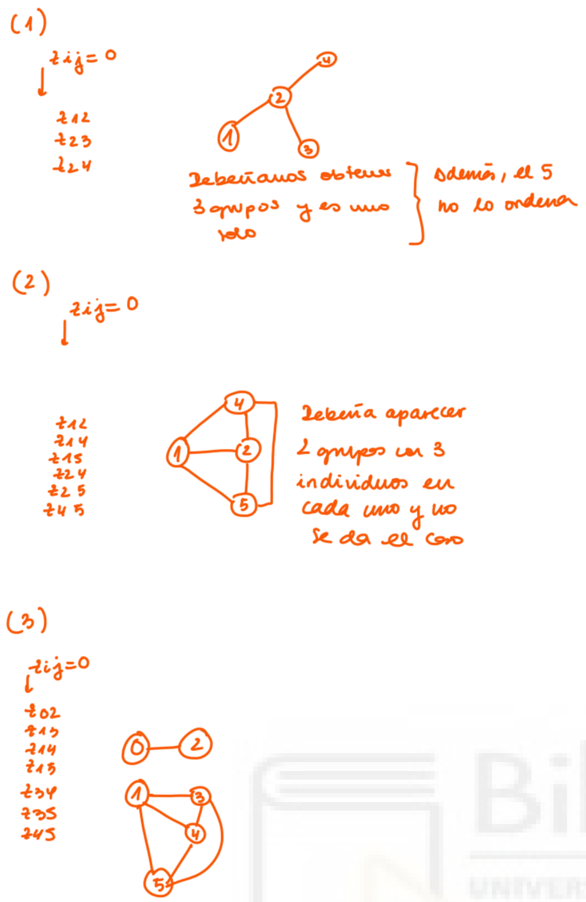
8.5 Other content

Expected loading time to obtain the Model solution for PISA data



Note: 360 hours equals 15 days waiting for a response.

Relevant raw notes taken during the creation of the Model.



Probando soluciones del modelo nuevo:

6 : 12 =

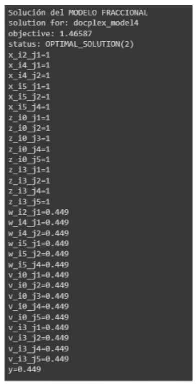
= CHL, COL, CZE, DEU, ESP, EST.

1) 2 grupos → 3   1-2   2-3   1-3 ; → 3   4-5   5-6   4-6 } $\sum_{i\in I}\sum_{j\in I} t_{ij} = 9$. (1)

2) 3 grupos → 2   1-2 ; → 2   3-4 ; → 2   5-6 } $\sum_{i\in I}\sum_{j\in I} t_{ij} = 12$ (2)

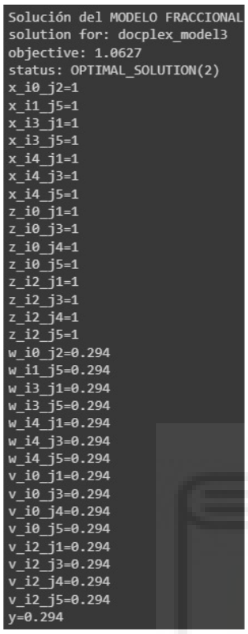
3) 2 grupos → 2   1-2 ; → 4   3-4   4-5   5-6   3-5   3-6   4-6 } $\sum_{i\in I}\sum_{j\in I} t_{ij} = 8$ (n)

(1)

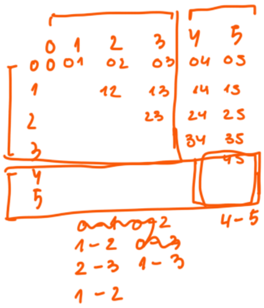

```
Solución del MODELO FRACCIONAL
solution for: docplex_model4
objective: 1.46587
status: OPTIMAL_SOLUTION(2)
x_i2_j1=1
x_i4_j1=1
x_i4_j2=1
x_i5_j1=1
x_i5_j2=1
x_i5_j4=1
z_i0_j1=1
z_i0_j2=1
z_i0_j3=1
z_i0_j4=1
z_i0_j5=1
z_i3_j1=1
z_i3_j2=1
z_i3_j4=1
z_i3_j5=1
w_i2_j1=0.449
w_i4_j1=0.449
w_i4_j2=0.449
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

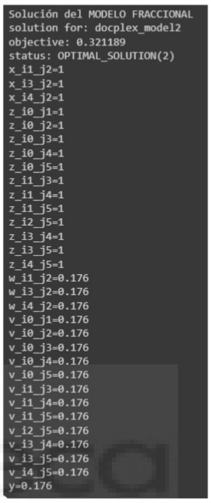
(n)


```
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



(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 MONGF

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

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

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

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

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

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

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

· 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

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

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

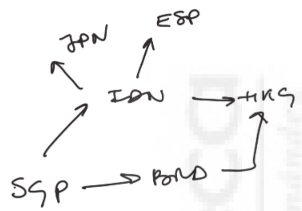
i → j, j → k, k → i; 1-2, 2-3

[$z_{ij}=z_{jk}=z_{ki}$
Integridad de los grupos.

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

0 0 = 0

1 1 = 2



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

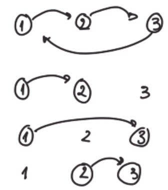
$z_{12}=0$ $z_{45}=0$

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

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

grupo 1	grupo 2
1 2 3	4 5 6
z_{12} z_{13} z_{23}	z_{45} z_{56} 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 $x_{ij}+x_{ji}+z_{ij}+z_{ji}\le 1 \quad \forall i,j\in I$

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

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

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

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

||

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

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

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

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

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

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

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

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

$w_{ij}\le y$

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

$v_{ij}\le y$

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

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

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

↳ $y=\frac{1}{a_{ij}z_{ij}}$

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

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:

	1	2	3	4
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

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]

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

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

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

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

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

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

	1	2	3	4
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

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 sugiero: $x_{ij}+x_{ji}+z_{ij}+z_{ji}=1$ → si pertenecen al mismo grupo z_{ij}

$z_{01}=1$, $z_{02}=1$] $z_{03}=0$

$z_{13}=1$, $z_{23}=1$] $z_{12}=0$

[1]–[2], [0]–[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 queremos que el 0 solo se enlace con uno

6x6

↳ 2 grupos → 3: 0 – 1, 0 – 2, 1 – 2

→ 3: 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 →

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

LOP normal

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

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?

	1	2	3	4
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

- 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 MONSF

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

↗ si $x_{ij} = 1$, no pertenecen al mismo grupo, ahora no lo comparamos.

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

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

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

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

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

Problema

· $z_{ij} + z_{ji} \leq 1$ { si $z_{12} = 1$, 1 y 2 no pertenecen al mismo grupo, por lo que 2 y 1 tampoco; 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	BRO	JPN	ESP
SGP	0	13223	33725	142660	20776	131706
HKG	12233	0	30806	121701		105303
ION	223	426	0	91819		9481
BRO	1648	2471	83532	0		44927
JPN	12076					211851
ESP	23438	28193		179037		

Matriz Pisa

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

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}

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

3 indiv.   3!

nº parejas =

$= \frac{n!}{m!\,(n-m)!} =$ n = indiv / grupo; m = parejas (2)

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

39×39 → 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 Links

Below are links to Google Drive folders, where you can download the files used for this work.

- Access to all CSV format matrices used during the work: Access
- Access to Lingo code used during the document: Access