



UNIVERSIDAD NACIONAL DE INGENIERIA
FACULTAD DE CIENCIAS

CC412 PROYECTO DE TESIS I

ASESOR:
MG. JORGE ENRIQUE MAYTA GUILLERMO

Porqué Perú no le ganó a Australia, explicado por programación lineal en LINDO

AUTOR:
Kenjhy Javier Bazan Turín

12 de agosto de 2022

Agradecimientos

Dedicado a mi padre, a mi madre y mi familia por su apoyo incondicional a lo largo de la carrera.

A mi asesor Mg. Jorge Enrique Mayta Guillermo por su guía y compromiso para el desarrollo de la presente tesis.

A la Universidad Nacional de Ingeniería y a todos mis docentes, quienes me convirtieron en un profesional preparado para asumir todo tipo de desafíos que se me presenten.

Resumen

Tal como sucedió en noviembre de 2017 contra Nueva Zelanda, la Selección Peruana se alistaba para afrontar un nuevo repechaje de cara a su clasificación al Mundial de Qatar 2022. La consigna del equipo de todos era superar a Australia el lunes 13 de junio de 2022 en el majestuoso y moderno Ahmad Bin Ali Stadium en Doha. Este recinto albergó alrededor de 40 mil espectadores y en su mayoría peruanos.

Sin embargo, la blanquirroja, dirigida por Ricardo Gareca, no consiguió repetir la hazaña de clasificar al Mundial de Rusia 2018 vía repechaje.

Por tal razón. En el presente trabajo de investigación, abordamos un análisis totalmente distinto al deportivo o estadístico al cual solemos estar acostumbrados para así determinar el motivo por el cual la selección peruana de fútbol no logró conseguir la victoria ante el seleccionado australiano, todo esto explicado desde un punto de vista matemático y computacional, basado en la programación lineal usando como herramienta el lenguaje de programación LINDO.

Palabras Claves: *Fútbol, Perú, Australia, repechaje, Mundial, Qatar, LINDO*

Abstract

Just as it happened in November 2017 against New Zealand, the Peruvian National Team was getting ready to face a new playoff for its qualification for the Qatar 2022 World Cup. The motto of everyone's team was to beat Australia on Monday, June 13, 2022, in the majestic and modern Ahmad Bin Ali Stadium in Doha. This venue housed around 40 thousand spectators and mostly Peruvians.

However, the red and white team, directed by Ricardo Gareca, failed to repeat the feat of qualifying for the 2018 World Cup in Russia via playoffs.

For this reason. In the present research work, we approach a totally different analysis to the sports or statistical analysis to which we are usually accustomed in order to determine the reason why the Peruvian soccer team did not manage to win against the Australian team, all this explained from a point from a mathematical and computational point of view, based on linear programming using the LINDO programming language as a tool.

Keywords: *Footbal, Peru, Australia, playoff, World, Qatar, LINDO*

Índice

1. Introducción	5
1.1. Objetivos	5
1.2. Hipótesis	5
2. Conceptos previos	6
2.1. El problema de la programación lineal	6
2.1.1. Ejemplo	7
2.2. LINDO	8
2.2.1. ¿Qué es LINDO?	8
2.2.2. Editor de LINDO	8
2.3. EL problema del árbol de expansión mínima	8
2.4. Algoritmo de Kruskal	10
2.5. Algoritmo de Dijkstra - El camino mas corto	12
2.5.1. Ejemplo Aplicativo	15
2.6. Flujo Máximo	17
2.6.1. Ejemplo Aplicativo	17
3. Desarrollo	20
4. Experimentación	22
4.1. Ruta más corta	22
4.1.1. Gallese - Lapadula	22
4.1.2. Gallese - Carrillo	23
4.1.3. Gallese - Cueva	23
4.2. Flujo máximo	25
4.2.1. Gallese - Lapadula	28
4.2.2. Gallese - Carrillo	29
4.2.3. Gallese - Cueva	30
5. Resultados	31
5.1. Ruta más corta	31
5.1.1. Gallese - Lapadula	31
5.1.2. Gallese - Carrillo	32
5.1.3. Gallese - Cueva	33
5.2. Flujo máximo	34
5.2.1. Gallese - Lapadula	34
5.2.2. Gallese - Carrillo	35
5.2.3. Gallese - Cueva	37

6. Discusiones	39
7. Conclusiones	39
8. Referencias	40

1. Introducción

La programación lineal se ocupa de la optimización de una función lineal, sea minimización o maximización. Esto mientras se satisface un conjunto de restricciones de igualdad y/o desigualdad lineal. El problema de programación lineal fue concebido por primera vez por George B. Dantzig alrededor de 1947 mientras trabajaba como asesor matemático del Contralor de la Fuerza Aérea de los Estados Unidos en busca del desarrollo de una herramienta de planificación mecanizada para un programa de suministro logístico, capacitación y despliegue por etapas.

Sin embargo el matemático y economista soviético L V. Kantorovich formuló y resolvió un problema de este tipo relacionado con la organización y la planificación en 1939, su trabajo permaneció desconocido hasta 1959. Por lo tanto, la concepción de la clase general de problemas de programación lineal suele atribuirse a Dantzig debido a que la Fuerza Aérea se refiere a sus diversos planes y cronogramas para ser implementados como "programas", el primer artículo publicado de Dantzig abordó este problema como "Programación en una estructura lineal".

El término "programación lineal" como tal, fue declarado por el economista y matemático T. C. Koopmans en el verano de 1948 mientras él y Dantzig paseaban cerca de la playa de Santa Mónica en California.

1.1. Objetivos

En esta subsección se presentarán los objetivos del informe:

- i) Presentar la teoría de grafos relacionada al algoritmo de Dijkstra y Ford-Fulkerson.
- ii) Presentar un grafo o red que tenga la información de la selección de Perú de fútbol en el partido de repesca para el mundial Qatar 2022.
- iii) Aplicar los algoritmos mencionados para concluir cual es la jugada que debió generar el equipo para obtener la jugada trascendental que hubiera significado la victoria.

1.2. Hipótesis

"La programación lineal usando el lenguaje de programación Lindo como herramienta puede explicar el porqué Perú no le ganó a Australia"

2. Conceptos previos

2.1. El problema de la programación lineal

Consideramos el siguiente problema de programación lineal.

Donde:

Notación	Descripción
$c_1x_1 + c_2x_2 + \dots + c_nx_n$	función objetivo
$c_1 + c_2 + \dots + c_n$	coeficientes de costo
$x_1 + x_2 + \dots + x_n$	variables de decisión

Cuadro 1: Notaciones Básicas

Definimos $c_1x_1 + c_2x_2 + \dots + c_nx_n$ se define como la función objetivo a minimizar y se denotará por la letra "Z".

Los coeficientes $c_1 + c_2 + \dots + c_n$ se definen como los coeficientes de costo y $x_1 + x_2 + \dots + x_n$ se definen como las variables de decisión a determinar.

$$\left. \begin{array}{ll} \text{Funcion} & \text{Objetivo} \\ & Z = c_1x_1 + c_2x_2 + \dots + c_nx_n \\ \text{Sujeto} & \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \geq b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \geq b_2 \\ \vdots + \vdots + \dots + \vdots \geq \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \geq b_m \\ x_1, x_2, \dots, x_n \geq 0 \end{array} \end{array} \right\} \quad (1)$$

La desigualdad:

$$\sum_{j=1}^n a_{ij}x_j \geq b_i \quad (2)$$

Denota la i-ésima restricción funcional. Los coeficientes a_{ij} para $i = 1, \dots, n$ y $j = 1, \dots, n$ son llamados coeficientes funcionales. Estos coeficientes funcionales forman la matriz de restricciones A.

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

El vector columna cuya i-ésima componente es b_i , que se denomina vector del lado derecho, representa los requisitos mínimos que deben satisfacerse. Las restricciones

$x_1 + x_2 + \dots + x_n \geq 0$ son las restricciones de no negatividad. Un conjunto de valores de las variables x_1, \dots, x_n que satisfacen todas las restricciones se denomina punto factible o solución factible. El conjunto de todos estos puntos constituye la región factible o el espacio factible.

Usando la terminología anterior, el problema de programación lineal se puede enumerar de la siguiente manera: Entre todas las soluciones factibles, encuentre una que minimice o maximice la función objetivo.

2.1.1. Ejemplo

Consideremos el siguiente problema lineal:

$$\left. \begin{array}{l} \text{Minimizar} \\ \quad Z = 2x_1 - 3x_2 \\ \text{Sujeto a} \\ \quad x_1 + x_2 \geq 6 \\ \quad -x_1 - 2x_2 \geq -18 \\ \quad x_1, x_2 \geq 0 \end{array} \right\} \quad (3)$$

En este caso, tenemos dos variables de decisión x_1 y x_2 . La función objetivo a minimizar es $2x_1 - 3x_2$. Las restricciones y la región factible se ilustran en la Figura 1. El problema de optimización es, por lo tanto, encontrar un punto en la región factible que tenga el valor objetivo más pequeño posible.

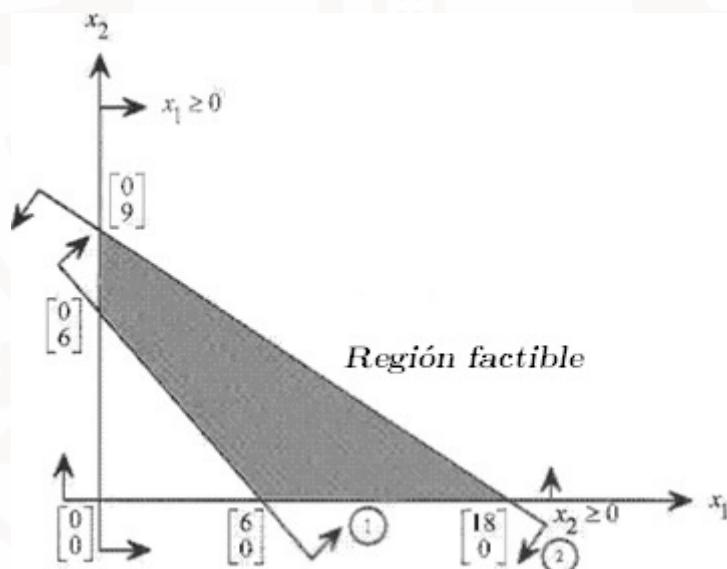


Ilustración de la región factible

2.2. LINDO

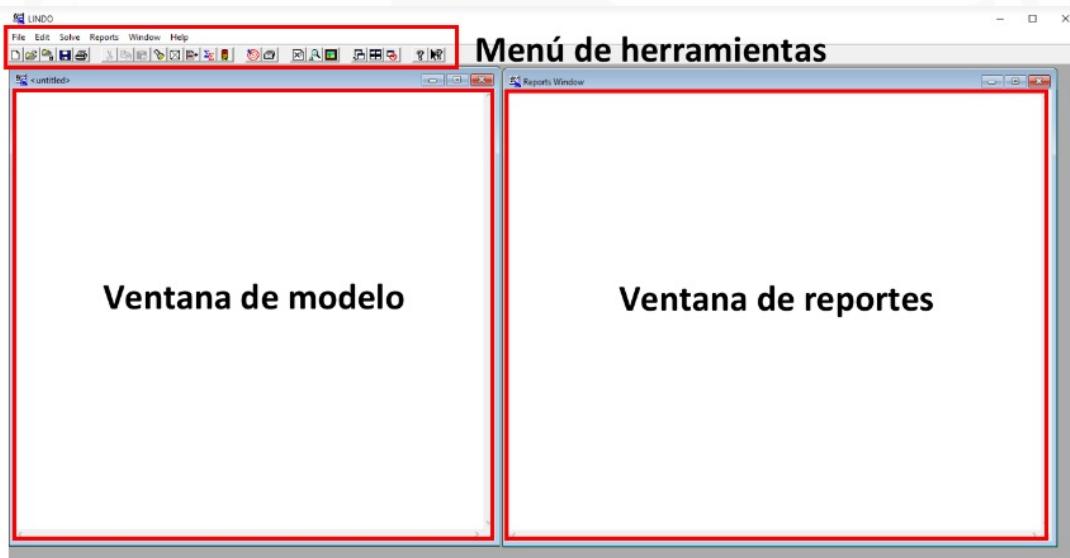
2.2.1. ¿Qué es LINDO?

Linear, INteractive, and Discrete Optimizer viene a ser el acrónimo de LINDO, este es un lenguaje de programación usado para resolver problemas de programación lineal y no lineal.

La importancia de LINDO radica en la facilidad al ingresar el modelo y la sencillez para obtener la solución óptima y el análisis de sensibilidad.

2.2.2. Editor de LINDO

Link de descarga: <https://www.lindo.com/index.php/ls-downloads>



Interfaz Lindo

2.3. EL problema del árbol de expansión mínima

Introducción:

Imagine un mapa de su región favorita de campo con unos 30 o 40 pueblos. Algunos pares de aldeas están conectadas por caminos de grava. La municipalidad decide modernizar algunas de estas carreteras adecuadas para la conducción rápida de automóviles, pero quiere invertir la menor cantidad de dinero posible con la condición de que sea posible viajar entre dos pueblos a lo largo de una carretera. De esta manera se consigue un problema fundamental llamado árbol de expansión minimal.

Esta sección está dedicada a su solución.



Definición 1. Sea $G = (V, E)$ un grafo.

- 1) Para cada arista $e \in E(G)$ asignemos un número real no negativo $w(e)$ llamado el peso de la arista e
- 2) El grafo G junto con una función de peso $w : E \rightarrow \mathbb{R}$ es llamada una red o malla.

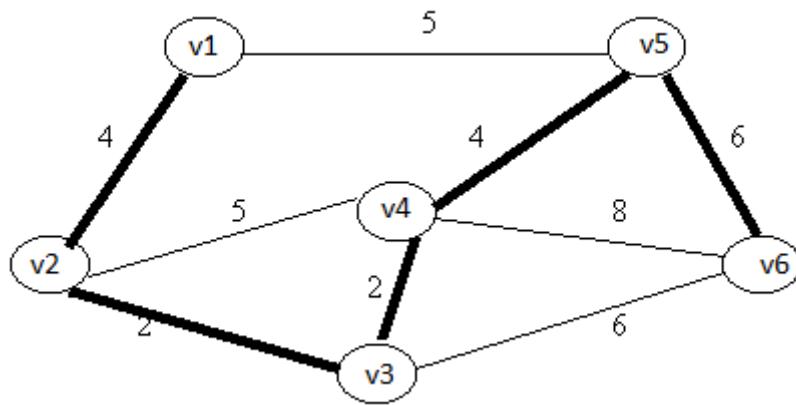
Observaciones:

- El problema a tratar se reformula matemáticamente, dado un grafo $G = (V, E)$ con función peso no negativa w en las aristas, encontrar un subgrafo conexo de expansión (V', E') tal que la suma

$$w(E') = \sum_{e \in E'} w(e)$$

posea el mínimo valor posible.

Ejemplo 1. La siguiente red muestra un árbol de expansión mínima T que con $V(T) = \{v_1, v_2, v_3, v_4, v_5\}$ y $E(T) = \{\{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_4\}, \{v_4, v_5\}, \{v_5, v_6\}\}$



Es decir $w(T) = 4 + 2 + 2 + 4 + 6 = 18$.

2.4. Algoritmo de Kruskal

La entrada es un grafo conexo $G = (V, E)$ con función de peso w para las aristas. Sea las aristas e_1, e_1, \dots, e_m tal que:

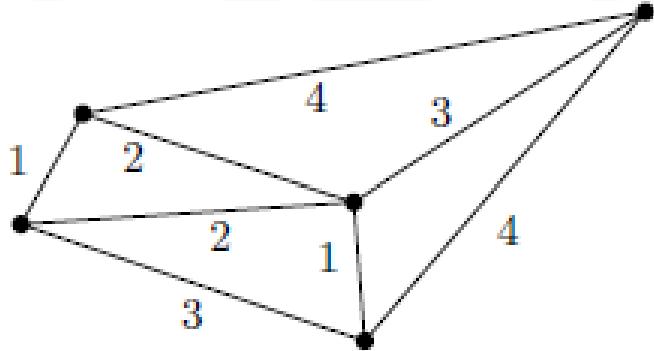
$$w(e_1) \leq w(e_2) \leq \dots \leq w(e_n).$$

Para el ordenamiento de aristas ejecutar el algoritmo de expansión:

- 1) Se crea un bosque B (un conjunto de árboles), donde cada vértice del grafo es un árbol separado.
- 2) Se crea un conjunto C que contenga a todas las aristas del grafo.
- 3) Mientras C es no vacío, suprimir una arista de peso mínimo de C .
- 4) Si esa arista conecta dos árboles diferentes se añade al bosque, combinando los dos árboles en un solo árbol.
- 5) En caso contrario, se desecha la arista.

Al acabar el algoritmo, el bosque tiene un solo componente, el cual forma un árbol de expansión mínima del grafo.

Ejemplo 2. Dada la siguiente red o malla:



Una posible ejecución del algoritmo de Kruskal se muestra en el siguiente diagrama:



Teorema 1. El algoritmo de Kruskal resuelve el problema de árbol expansión mínima.

Demostración. Sea T un árbol de expansión encontrado por el algoritmo y sea \check{T} otro árbol de expansión del grafo $G = (V, E)$. Por demostrar que $w(E(T)) \leq w(E(\check{T}))$. Denotemos las aristas de T por $e'_1, e'_2, \dots, e'_{n-1}$ tal que $w(e'_1) \leq w(e'_2) \leq \dots \leq w(e'_{n-1})$ (la arista $e_i = e_j$ para algún j). De igual modo, sea $\check{e}_1, \dots, \check{e}_{n-1}$ las aristas de \check{T} ordenados en orden creciente por pesos. Vamos a probar que para $i = 1, \dots, n-1$ se tiene que:

$$w(e'_i) \leq w(\check{e}_i).$$

Esto prueba que T es un árbol de expansión mínima, procedamos por contradicción supongamos que la desigualdad anterior no se cumple, entonces $w(e'_i) > w(\check{e}_i)$. Al considerar los conjuntos

$$\begin{aligned} E' &= \{e'_1, e'_2, \dots, e'_{i-1}\}, \\ \check{E} &= \{\check{e}_1, \check{e}_2, \dots, \check{e}_i\}. \end{aligned}$$

Los grafos (V, E') y (V, \check{E}) no contienen ciclos y $|E'| = i - 1$, $|\check{E}| = i$. Para llegar a una contradicción es suficiente probar que existe una arista $e \in \check{E}$ tal que el grafo $(V, E' \cup \{e\})$ no contiene ciclos, así obtenemos que $w(e) \leq w(\check{e}_i) < w(e'_i)$ y esto significa que al elegir la arista e en el algoritmo nosotros cometimos un error,

pero no hay necesidad de eliminar la arista e ya que podemos seleccionar a e'_i en su reemplazo. Por tanto, es suficiente probar que, si $E', E \subseteq \binom{V}{2}$ son dos conjuntos de aristas tal que el grafo (V, \check{E}) no tiene ciclos y $|E'| < |\check{E}|$, entonces alguna arista $e \in \check{E}$ conecta vértices de componentes distintas del grafo V, E' . Esto se puede hacer mediante un simple argumento de conteo. Sea V_1, \dots, V_s los conjuntos de vértices de las componentes del grafo (V, E') , luego se tiene que:

$$|E' \cap \binom{V_j}{2}| \geq |V_j| - 1,$$

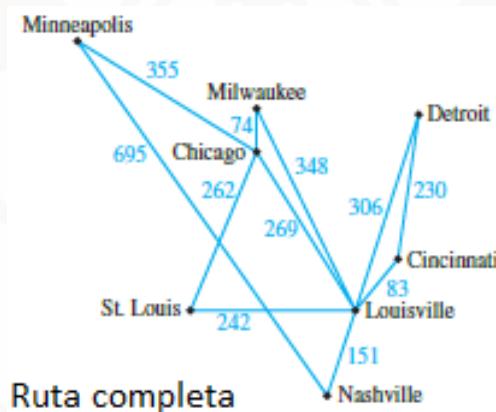
y al sumar estas desigualdades sobre j se tiene que $|E'| \geq n - s$. Por otro lado, como \check{E} no tiene ciclos, se tiene:

$$|\check{E} \cap \binom{V_j}{2}| \leq |V_j| - 1,$$

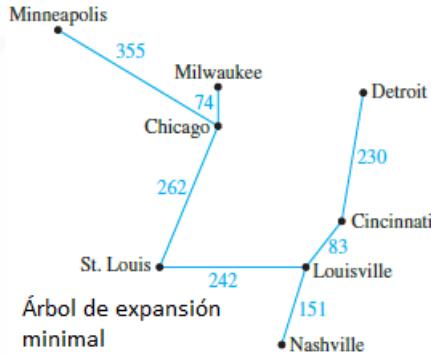
y por lo tanto a lo más $n - s$ aristas de \check{E} están contenidos en alguna de las componentes V_j , pero como asumimos que $|\check{E}| > |E'|$, existiría una arista $e \in \check{E}$ que pertenece a dos componentes distintas. \square

2.5. Algoritmo de Dijkstra - El camino mas corto

Aunque los árboles producidos por el algoritmo de Kruskal tiene el menor peso posible total en comparación con todos los demás árboles de expansión para los grafos dados, no siempre muestra la distancia más corta entre dos puntos en el grafo. Por ejemplo la siguiente red muestra la ruta completa de viajes entre ciudades con sus respectivas distancias en millas



Se puede probar que al aplicar del algoritmo de Kruskal en la red anterior se tiene el siguiente árbol de expansión minimal:



De acuerdo con el sistema de ruta completo, se puede volar directamente de Nashville a Minneapolis por una distancia de 695 millas, mientras que si usa el árbol de expansión mínimo la única forma de volar desde Nashville a Minneapolis es pasando por Louisville, St. Louis y Chicago, lo que da distancia total de $151+242+262+355 = 1010$ millas.

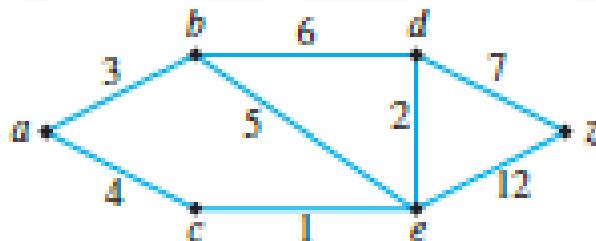
Algoritmo de Dijkstra

ALGORITHM 1 Dijkstra's Algorithm.

```

procedure Dijkstra( $G$ : weighted connected simple graph, with
    all weights positive)
    { $G$  has vertices  $a = v_0, v_1, \dots, v_n = z$  and lengths  $w(v_i, v_j)$ 
     where  $w(v_i, v_j) = \infty$  if  $\{v_i, v_j\}$  is not an edge in  $G$ }
    for  $i := 1$  to  $n$ 
         $L(v_i) := \infty$ 
     $L(a) := 0$ 
     $S := \emptyset$ 
    {the labels are now initialized so that the label of  $a$  is 0 and all
     other labels are  $\infty$ , and  $S$  is the empty set}
    while  $z \notin S$ 
         $u :=$  a vertex not in  $S$  with  $L(u)$  minimal
         $S := S \cup \{u\}$ 
        for all vertices  $v$  not in  $S$ 
            if  $L(u) + w(u, v) < L(v)$  then  $L(v) := L(u) + w(u, v)$ 
            {this adds a vertex to  $S$  with minimal label and updates the
             labels of vertices not in  $S$ }
    return  $L(z)$  { $L(z) =$  length of a shortest path from  $a$  to  $z$ }
```

Ejemplo 3. Dada la red



Veamos los pasos en la ejecución del algoritmo de Dijkstra para hallar la ruta más corta de a hacia z .

Step	$V(T)$	$E(T)$	F	$L(a)$	$L(b)$	$L(c)$	$L(d)$	$L(e)$	$L(z)$
0	{a}	\emptyset	{a}	0	∞	∞	∞	∞	∞
1	{a}	\emptyset	$\{b, c\}$	0	3	4	∞	∞	∞
2	{a, b}	{(a, b)}	{c, d, e}	0	3	4	9	8	∞
3	{a, b, c}	{(a, b), (a, c)}	{d, e}	0	3	4	9	5	∞
4	{a, b, c, e}	{(a, b), (a, c), (c, e)}	{d, z}	0	3	4	7	5	17
5	{a, b, c, e, d}	{(a, b), (a, c), (c, e), (e, d)}	{z}	0	3	4	7	5	14
6	{a, b, c, e, d, z}	{(a, b), (a, c), (c, e), (e, d), (e, z)}							

El problema de ruta más corta se formula como un modelo que conecta a un nodo de origen y un nodo de destino Para lo cual determina un subconjunto de arcos y un subconjunto de nodos que garanticen la mínima distancia

$$\min \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$$

Sujeto a:

$$\sum_{j=1}^n x_{ij} - \sum_{k=1}^l x_{ki} = \begin{cases} 1 & \text{si } i \text{ nodo inicio} \\ -1 & \text{si } i \text{ nodo final} \\ 0 & \text{si } i \text{ en los nodos intermedios} \end{cases}$$

$$\forall x_{ij} \geq 0$$

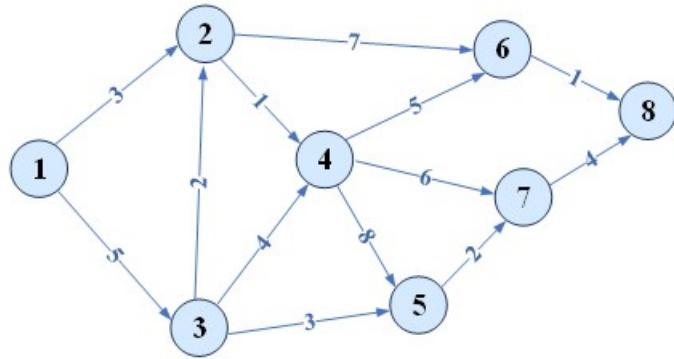
donde:

c_{ij} : es el peso de cada arista dirigida $e_{ij} = (i, j)$.

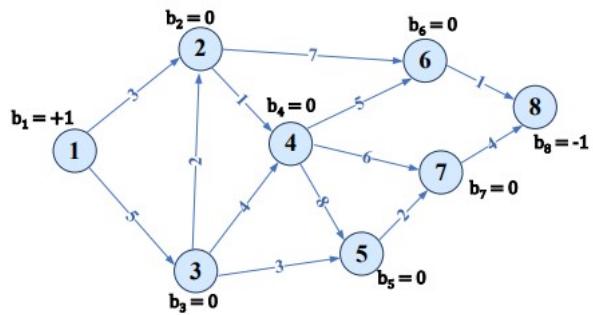
x_{ij} : la decisión de tomar la arista $e_{ij} = (i, j)$

2.5.1. Ejemplo Aplicativo

El vuelo **ACRF2020** está a punto de despegar desde el nodo 1 hacia el nodo 8. La siguiente red describe las posibles rutas consideradas para poder emprender con éxito el vuelo a destino. El viento a lo largo de cada arco afecta de manera considerable el tiempo de vuelo, y por ende, el consumo de combustible. La red muestra los tiempos de vuelo (en horas). **Para que el vuelo optimice costos, usted debe elegir la ruta que minimiza el tiempo total de vuelo.**



Modelo matemático



$$\min \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$$

$$\sum_{j=1}^n x_{ij} - \sum_{k=1}^l x_{ki} = \begin{cases} 1, & i = \text{nodo inicio} \\ -1, & i = \text{nodo fin} \\ 0, & \text{en los nodos intermedios} \end{cases}$$

$$\forall x_{ij} \geq 0$$

$$\begin{aligned} \min = & 3x_{12} + 5x_{13} + \\ & 1x_{24} + 7x_{26} + \\ & 2x_{32} + 4x_{34} + 3x_{35} + \\ & 5x_{46} + 6x_{47} + 8x_{45} + \\ & 2x_{57} + \\ & 1x_{68} + \\ & 4x_{78} \end{aligned}$$

$$\text{st: } \left\{ \begin{array}{l} N_1: x_{12} + x_{13} = 1 \\ N_8: -x_{68} - x_{78} = -1 \\ N_2: x_{24} + x_{26} - x_{12} - x_{32} = 0 \\ N_3: x_{32} + x_{34} - x_{13} = 0 \\ N_4: x_{46} + x_{47} + x_{45} - x_{24} - x_{34} = 0 \\ N_5: x_{57} - x_{35} - x_{45} = 0 \\ N_6: x_{68} - x_{26} - x_{46} = 0 \\ N_7: x_{78} - x_{47} - x_{57} = 0 \end{array} \right.$$

$$\begin{aligned} & x_{12}, x_{13}, x_{24}, x_{26}, x_{32}, x_{34}, x_{35}, x_{46}, x_{47}, \\ & x_{45}, x_{57}, x_{68}, x_{78} \geq 0 \end{aligned}$$

Modelo Computacional

```
MIN 3 X12 + 5 X13 + X24 + 7 X26 + 2 X32 + 4 X34 + 3 X35 + 5 X46 + 6 X47  
+ 8 X45 + 2 X57 + X68 + 4 X78  
SUBJECT TO  
2) X12 + X13 = 1  
3) - X68 - X78 = - 1  
4) - X12 + X24 + X26 - X32 = 0  
5) - X13 + X32 + X34 = 0  
6) - X24 - X34 + X46 + X47 + X45 = 0  
7) - X35 - X45 + X57 = 0  
8) - X26 - X46 + X68 = 0  
9) - X47 - X57 + X78 = 0  
END
```

Resultado

OBJECTIVE FUNCTION VALUE

1) 10.00000

VARIABLE	VALUE	REDUCED COST
X12	1.000000	0.000000
X13	0.000000	0.000000
X24	1.000000	0.000000
X26	0.000000	1.000000
X32	0.000000	4.000000
X34	0.000000	5.000000
X35	0.000000	9.000000
X46	1.000000	0.000000
X47	0.000000	4.000000
X45	0.000000	8.000000
X57	0.000000	0.000000
X68	1.000000	0.000000
X78	0.000000	0.000000

2.6. Flujo Máximo

El problema de flujo máximo se formula como un modelo que conecta a un nodo de origen(fuente) y un nodo de destino (sumidero). Para lo cual determina un flujo F que hace uso de un subconjunto de aristas y un subconjunto de vértices los cuales forman cadenas de flujo que garantizan el máximo flujo

$$\max F$$

Sujeto a:

$$\sum_{j=1}^n x_{ij} - \sum_{k=1}^l x_{ki} = \begin{cases} F & \text{si } i \text{ nodo inicio} \\ -F & \text{si } i \text{ nodo final} \\ 0 & \text{si en los nodos intermedios} \end{cases}$$

Además:

$x_{ij} \leq U_{ij}$, el flujo de tránsito (i, j) no debe superar la capacidad U_{ij} .

$x_{ij} \geq L_{ij}$, el flujo de tránsito (i, j) debe suprir por lo menos la capacidad de L_{ij} .

$$\forall x_{ij} \geq 0$$

donde:

U_{ij} : la capacidad máxima de la arista (i, j) .

L_{ij} : la capacidad mínima de la arista (i, j) .

F : el flujo máximo de toda la red

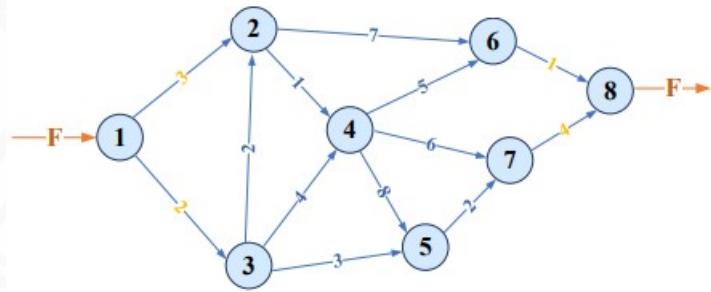
x_{ij} : la cantidad de flujo que transita en el arco (i, j)

2.6.1. Ejemplo Aplicativo

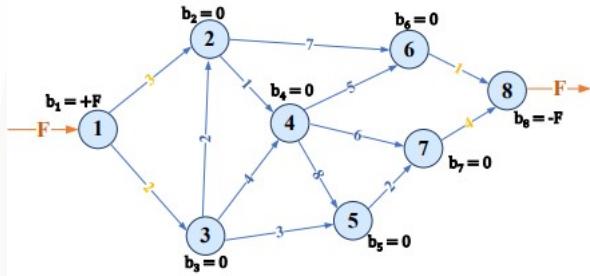
En la aldea de YoKoha se han incorporado dos clanes (nodo 6, nodo 7). Cada clan cuenta con un área de terreno delimita, no obstante existe la problemática que no disponen de agua, para lo cual se construirán acueductos que transporten agua.

El agua es obtenida desde los clanes que tiene experiencia en técnicas “Suiton” (nodo 2, nodo 3) y será transportada por los acueductos que transitaran por las tierras de los clanes nodo 4 y nodo 5 hasta llegar a las tierras de nodo 6 y nodo 7.

En la red adjunta se observa como serán construidos los canales, los arcos representan la máxima cantidad de agua en miles de m^3 , y el nodo representa la oferta o demanda de agua. Determine si se podrá abastecer de agua a los nuevos clanes.



Modelo matemático



$$\max F$$

$$\sum_{j=1}^n x_{ij} - \sum_{k=1}^l x_{ki} = \begin{cases} F, & i = \text{nodo inicio} \\ -F, & i = \text{nodo fin} \\ 0, & \text{en los nodos intermedios} \end{cases}$$

$$x_{ij} \leq U_{ij}$$

$$x_{ij} \geq L_{ij},$$

$$\forall x_{ij} \geq 0$$

$$\max = F$$

$$st \begin{cases} N_1: x_{12} + x_{13} = F \\ N_8: -x_{68} - x_{78} = -F \\ N_2: x_{24} + x_{26} - x_{12} - x_{32} = 0 \\ N_3: x_{32} + x_{34} + x_{35} - x_{13} = 0 \\ N_4: x_{45} + x_{46} + x_{47} - x_{24} - x_{34} = 0 \\ N_5: x_{57} - x_{35} - x_{45} = 0 \\ N_6: x_{68} - x_{26} - x_{46} = 0 \\ N_7: x_{78} - x_{47} - x_{57} = 0 \end{cases}$$

$$st \begin{cases} x_{12} \leq 3; & x_{13} \leq 2; & x_{24} \leq 1 \\ x_{26} \leq 7; & x_{32} \leq 2; & x_{34} \leq 4 \\ x_{35} \leq 3; & x_{45} \leq 8; & x_{46} \leq 5 \\ x_{47} \leq 6; & x_{57} \leq 2; & x_{68} \leq 1 \\ & & x_{78} \leq 4 \end{cases}$$

$$x_{12}, x_{13}, x_{24}, x_{26}, x_{32}, x_{34}, x_{35}, x_{46}, x_{47}, \\ x_{45}, x_{57}, x_{68}, x_{78} \geq 0$$

Modelo Computacional

```
MAX F
SUBJECT TO
 2) - F + X12 + X13 = 0
 3) F - X68 - X78 = 0
 4) - X12 + X24 + X26 - X32 = 0
 5) - X13 + X32 + X34 + X35 = 0
 6) - X24 - X34 + X46 + X47 + X45 = 0
 7) - X35 - X45 + X57 = 0
 8) X68 - X26 - X46 = 0
 9) X78 - X47 - X57 = 0
10) X12 <= 3
11) X13 <= 2
12) X24 <= 1
13) X26 <= 7
14) X32 <= 2
15) X34 <= 4
16) X35 <= 3
17) X46 <= 5
18) X47 <= 6
19) X45 <= 8
20) X57 <= 2
21) X68 <= 1
22) X78 <= 4
END
```

Resultado

LP OPTIMUM FOUND AT STEP 2		
OBJECTIVE FUNCTION VALUE		
1)	4.000000	
VARIABLE	VALUE	REDUCED COST
P	4.000000	0.000000
X12	2.000000	0.000000
X13	2.000000	0.000000
X68	1.000000	0.000000
X78	3.000000	0.000000
X24	1.000000	0.000000
X26	1.000000	0.000000
X32	0.000000	1.000000
X34	0.000000	0.000000
X35	2.000000	0.000000
X46	0.000000	1.000000
X47	1.000000	0.000000
X45	0.000000	0.000000
X57	2.000000	0.000000

3. Desarrollo

Vamos a analizar el partido de Perú vs Australia, por lo cual vamos a contabilizar los pases incorrectos, Teniendo en cuenta que consideramos si el pase de un jugador peruano a otro jugador peruano es correcto siempre en cuando este no sea un pase errado o comprometido, de tal forma que si el jugador receptor recibe el pase y al cabo de un intervalo menor de 3 segundos pierde la posesión del balón se considerará como un pase incorrecto.



Ejemplo de pase incorrecto

Además se considerará los pesos de las aristas de la forma siguiente:

- Arista para la ruta más corta

$$c_{ij} = \frac{\text{Números de pases incorrectos del jugador } i \text{ al jugador } j}{\text{Número de pases totales del jugador } i \text{ al jugador } j}$$

- Arista para el flujo máximo

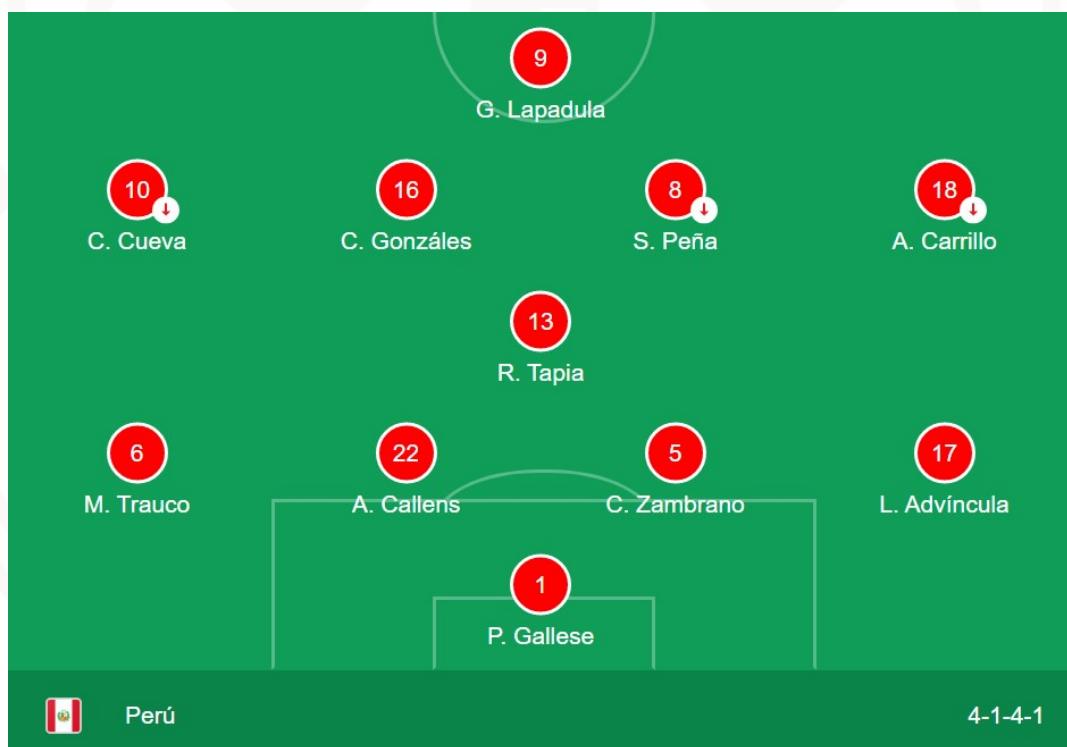
$$c_{ij} = \frac{\text{Números de pases correctos del jugador } i \text{ al jugador } j}{\text{Número de pases totales del jugador } i \text{ al jugador } j}$$

También se considerará para la ruta más corta y el flujo máximo como el nodo inicial al portero *Gallese* y como nodo final a los delanteros; es decir, *Carrillo*, *Lapadula* y *Cueva*.



Delanteros del seleccionado peruano

Se representará la alineación inicial del equipo peruano con el dorsal de cada jugador de la forma siguiente:



11 inicial Perú vs Australia Repesca Mundial Qatar 2022

4. Experimentación

En esta sección presentaremos los algoritmos que emplearemos para el análisis del partido de repesca por un cupo al Mundial Qatar 2022 entre Perú vs Australia.

4.1. Ruta más corta

Debemos de tener en cuenta que vamos a evaluar la ruta más corta en el mismo grafo, es por ello que la función a minimizar será la misma para todo los procesos, teniendo como variación el nodo inicial y el nodo final, entonces como consecuencia las restricciones también varían.

```
MIN 0 x1_6 + 0 x1_22 + 0 x1_5 + 0 x1_17 + 0 x1_13 + 0.33 x1_10 + 1  
    x1_16 + 1 x1_8 + 0.67 x1_18 + 0 x1_9  
+ 0 x6_1 + 0 x6_22 + 0 x6_5 + 0.5 x6_17 + 0.5 x6_13 + 0.22 x6_10 + 0  
    x6_16 + 0 x6_8 + 0 x6_18 + 1 x6_9  
+ 0 x22_1 + 0 x22_6 + 0 x22_5 + 0 x22_17 + 0 x22_13 + 0 x22_10 + 0  
    x22_16 + 0 x22_8 + 0.5 x22_18 + 0 x22_9  
+ 0 x5_1 + 0 x5_6 + 0 x5_22 + 0.06 x5_17 + 0 x5_13 + 0 x5_10 + 0 x5_16  
    + 0.5 x5_8 + 0 x5_18 + 1 x5_9  
+ 0 x17_1 + 0 x17_6 + 0 x17_22 + 0 x17_5 + 0 x17_13 + 0 x17_10 + 0  
    x17_16 + 0.13 x17_8 + 0.5 x17_18 + 1 x17_9  
+ 0 x13_1 + 0 x13_6 + 0 x13_22 + 0 x13_5 + 0 x13_17 + 0.25 x13_10 + 0  
    x13_16 + 0 x13_8 + 1 x13_18 + 0.33 x13_9  
+ 0 x10_1 + 0 x10_6 + 0 x10_22 + 0 x10_5 + 0.5 x10_17 + 0.2 x10_13 +  
    0.2 x10_16 + 0.25 x10_8 + 0.33 x10_18 + 0.4 x10_9  
+ 0 x16_1 + 0 x16_6 + 0 x16_22 + 0 x16_5 + 0 x16_17 + 0 x16_13 + 0  
    x16_10 + 0 x16_8 + 0.33 x16_18 + 1 x16_9  
+ 0 x8_1 + 0 x8_6 + 0 x8_22 + 0 x8_5 + 0.11 x8_17 + 0 x8_13 + 0.44  
    x8_10 + 0 x8_16 + 0 x8_18 + 1 x8_9  
+ 0 x18_1 + 0 x18_6 + 0 x18_22 + 0 x18_5 + 0.3 x18_17 + 0 x18_13 + 0.2  
    x18_10 + 0.25 x18_16 + 0.33 x18_8 + 0.6 x18_9  
+ 0 x9_1 + 0 x9_6 + 0 x9_22 + 0 x9_5 + 0 x9_17 + 0 x9_13 + 0 x9_10 + 0  
    x9_16 + 0 x9_8 + 0 x9_18
```

Grafo de la Ruta más corta

4.1.1. Gallese - Lapadula

En primer lugar vamos a considerar como nodo inicial a Gallese y como nodo final a Lapadula.

SUBJECT TO

- 2) $X1_{10} + X1_{16} + X1_8 + X1_{18} = 1$
- 3) $X6_{17} + X6_{13} + X6_{10} + X6_9 = 0$
- 4) $X22_{18} = 0$

- EQUACIONES DIFERENCIALES
-
- 5) $X_{5_17} + X_{5_8} + X_{5_9} = 0$
 6) $X_{17_8} + X_{17_18} + X_{17_9} - X_{6_17} - X_{5_17} - X_{10_17} - X_{8_17} - X_{18_17} = 0$
 7) $X_{13_10} + X_{13_18} + X_{13_9} - X_{6_13} - X_{10_13} = 0$
 8) $X_{10_17} + X_{10_13} + X_{10_16} + X_{10_8} + X_{10_18} + X_{10_9} - X_{1_10} - X_{6_10} - X_{13_10} - X_{8_10} - X_{18_10} = 0$
 9) $X_{16_18} + X_{16_9} - X_{1_16} - X_{10_16} - X_{18_16} = 0$
 10) $X_{8_17} + X_{8_10} + X_{8_9} - X_{1_8} - X_{5_8} - X_{17_8} - X_{10_8} - X_{18_8} = 0$
 11) $X_{18_17} + X_{18_10} + X_{18_16} + X_{18_8} + X_{18_9} - X_{1_18} - X_{22_18} - X_{17_18} - X_{13_18} - X_{10_18} - X_{16_18} = 0$
 12) $-X_{6_9} - X_{5_9} - X_{17_9} - X_{13_9} - X_{10_9} - X_{16_9} - X_{8_9} - X_{18_9} = -1$

END

4.1.2. Gallese - Carrillo

En segundo lugar vamos a considerar como nodo inicial a Gallese y como nodo final a Carrillo.

SUBJECT TO

- 2) $X_{1_10} + X_{1_16} + X_{1_8} + X_{1_18} = 1$
 3) $X_{6_17} + X_{6_13} + X_{6_10} + X_{6_9} = 0$
 4) $X_{22_18} = 0$
 5) $X_{5_17} + X_{5_8} + X_{5_9} = 0$
 6) $X_{17_8} + X_{17_18} + X_{17_9} - X_{6_17} - X_{5_17} - X_{10_17} - X_{8_17} - X_{18_17} = 0$
 7) $X_{13_10} + X_{13_18} + X_{13_9} - X_{6_13} - X_{10_13} = 0$
 8) $X_{10_17} + X_{10_13} + X_{10_16} + X_{10_8} + X_{10_18} + X_{10_9} - X_{1_10} - X_{6_10} - X_{13_10} - X_{8_10} - X_{18_10} = 0$
 9) $X_{16_18} + X_{16_9} - X_{1_16} - X_{10_16} - X_{18_16} = 0$
 10) $X_{8_17} + X_{8_10} + X_{8_9} - X_{1_8} - X_{5_8} - X_{17_8} - X_{10_8} - X_{18_8} = 0$
 11) $X_{18_17} + X_{18_10} + X_{18_16} + X_{18_8} + X_{18_9} - X_{1_18} - X_{22_18} - X_{17_18} - X_{13_18} - X_{10_18} - X_{16_18} = -1$
 12) $-X_{6_9} - X_{5_9} - X_{17_9} - X_{13_9} - X_{10_9} - X_{16_9} - X_{8_9} - X_{18_9} = 0$

END

4.1.3. Gallese - Cueva

Finalmente vamos a considerar como nodo inicial a Gallese y como nodo final a Cueva.

SUBJECT TO

- 2) $X_{1_10} + X_{1_16} + X_{1_8} + X_{1_18} = 1$
 3) $X_{6_17} + X_{6_13} + X_{6_10} + X_{6_9} = 0$

-
- 4) $X_{22_18} = 0$
 5) $X_{5_17} + X_{5_8} + X_{5_9} = 0$
 6) $X_{17_8} + X_{17_18} + X_{17_9} - X_{6_17} - X_{5_17} - X_{10_17} - X_{8_17} - X_{18_17} = 0$
 7) $X_{13_10} + X_{13_18} + X_{13_9} - X_{6_13} - X_{10_13} = 0$
 8) $X_{10_17} + X_{10_13} + X_{10_16} + X_{10_8} + X_{10_18} + X_{10_9} - X_{1_10} - X_{6_10} - X_{13_10} - X_{8_10} - X_{18_10} = -1$
 9) $X_{16_18} + X_{16_9} - X_{1_16} - X_{10_16} - X_{18_16} = 0$
 10) $X_{8_17} + X_{8_10} + X_{8_9} - X_{1_8} - X_{5_8} - X_{17_8} - X_{10_8} - X_{18_8} = 0$
 11) $X_{18_17} + X_{18_10} + X_{18_16} + X_{18_8} + X_{18_9} - X_{1_18} - X_{22_18} - X_{17_18} - X_{13_18} - X_{10_18} - X_{16_18} = 0$
 12) $-X_{6_9} - X_{5_9} - X_{17_9} - X_{13_9} - X_{10_9} - X_{16_9} - X_{8_9} - X_{18_9} = 0$

END

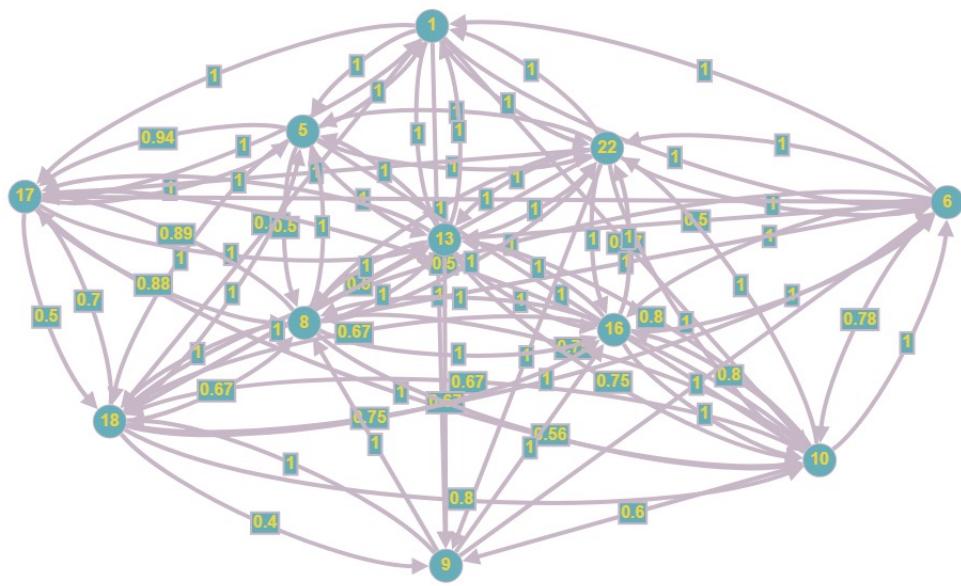
4.2. Flujo máximo

Análogamente al camino más corto, vamos a evaluar el flujo máximo en el mismo grafo, es por ello que la función a maximizar será la misma para todo los procesos, teniendo como variación el nodo inicial y el nodo final, entonces como consecuencia las restricciones también varían; sin embargo, estas se dan en las primeras 11 restricciones numeradas del número "2.^al número "12correspondientes a cada nodo, siendo el resto de restricciones constantes al evaluar el mismo grafo.

MAX	F
SUBJECT TO	
2) Inicio de restricciones variables	
...	
12) Final de restricciones variables	
13) X1_6 <= 1	
14) X1_22 <= 1	
15) X1_5 <= 1	
16) X1_17 <= 1	
17) X1_13 <= 1	
18) X1_10 <= 0.67	
19) X1_18 <= 0.33	
20) X1_9 <= 1	
21) X6_1 <= 1	
22) X6_22 <= 1	
23) X6_17 <= 0.5	
24) X6_13 <= 0.5	
25) X6_10 <= 0.78	
26) X6_16 <= 1	
27) X6_8 <= 1	
28) X6_18 <= 1	
29) X22_1 <= 1	
30) X22_6 <= 1	
31) X22_5 <= 1	
32) X22_13 <= 1	
33) X22_10 <= 1	
34) X22_16 <= 1	
35) X22_8 <= 1	
36) X22_18 <= 0.5	
37) X22_9 <= 1	
38) X5_1 <= 1	
39) X5_22 <= 1	
40) X5_17 <= 0.94	
41) X5_13 <= 1	
42) X5_10 <= 1	
43) X5_8 <= 0.5	
44) X5_18 <= 1	
45) X17_1 <= 1	

```
46) X17_22 <= 1
47) X17_5 <= 1
48) X17_13 <= 1
49) X17_10 <= 1
50) X17_8 <= 0.88
51) X17_18 <= 0.5
52) X13_1 <= 1
53) X13_22 <= 1
54) X13_5 <= 1
55) X13_17 <= 1
56) X13_10 <= 0.75
57) X13_16 <= 1
58) X13_8 <= 1
59) X13_9 <= 0.67
60) X10_6 <= 1
61) X10_22 <= 1
62) X10_17 <= 0.5
63) X10_13 <= 0.8
64) X10_16 <= 0.8
65) X10_8 <= 0.75
66) X10_18 <= 0.67
67) X10_9 <= 0.6
68) X16_6 <= 1
69) X16_22 <= 1
70) X16_13 <= 1
71) X16_10 <= 1
72) X16_8 <= 1
73) X16_18 <= 0.67
74) X8_22 <= 1
75) X8_5 <= 1
76) X8_17 <= 0.89
77) X8_13 <= 1
78) X8_10 <= 0.56
79) X8_16 <= 1
80) X8_18 <= 1
81) X18_6 <= 1
82) X18_5 <= 1
83) X18_17 <= 0.7
84) X18_13 <= 1
85) X18_10 <= 0.8
86) X18_16 <= 0.75
87) X18_8 <= 0.67
88) X18_9 <= 0.4
89) X9_6 <= 1
90) X9_16 <= 1
91) X9_8 <= 1
92) X9_18 <= 1
```

END



Grafo del Flujo Máximo

4.2.1. Gallese - Lapadula

En primer lugar vamos a considerar como nodo inicial a Gallese y como nodo final a Lapadula.

MAX	F
SUBJECT TO	
2)	$-F + X1_6 + X1_22 + X1_5 + X1_17 + X1_13 + X1_10 + X1_18 + X1_9 - X22_1 - X5_1 - X17_1 - X13_1 = 0$
3)	$X6_1 + X6_22 + X6_17 + X6_13 + X6_10 + X6_16 + X6_8 + X6_18 - X1_6 - X22_6 - X10_6 - X16_6 - X18_6 - X9_6 = 0$
4)	$X22_1 + X22_6 + X22_5 + X22_13 + X22_10 + X22_16 + X22_8 + X22_18 + X22_9 - X1_22 - X6_22 - X5_22 - X17_22 - X13_22 - X10_22 - X16_22 - X8_22 = 0$
5)	$X5_1 + X5_22 + X5_17 + X5_13 + X5_10 + X5_8 + X5_18 - X1_5 - X22_5 - X17_5 - X13_5 - X8_5 - X18_5 = 0$
6)	$X17_1 + X17_22 + X17_5 + X17_13 + X17_10 + X17_8 + X17_18 - X1_17 - X6_17 - X5_17 - X13_17 - X10_17 - X8_17 - X18_17 = 0$
7)	$X13_1 + X13_22 + X13_5 + X13_17 + X13_10 + X13_16 + X13_8 + X13_9 - X1_13 - X6_13 - X22_13 - X5_13 - X17_13 - X10_13 - X16_13 - X8_13 - X18_13 = 0$
8)	$X10_6 + X10_22 + X10_17 + X10_13 + X10_16 + X10_8 + X10_18 + X10_9 - X1_10 - X6_10 - X22_10 - X5_10 - X17_10 - X13_10 - X16_10 - X8_10 - X18_10 = 0$
9)	$X16_6 + X16_22 + X16_13 + X16_10 + X16_8 + X16_18 - X6_16 - X22_16 - X13_16 - X10_16 - X8_16 - X18_16 - X9_16 = 0$
10)	$X8_22 + X8_5 + X8_17 + X8_13 + X8_10 + X8_16 + X8_18 - X6_8 - X22_8 - X5_8 - X17_8 - X13_8 - X10_8 - X16_8 - X18_8 - X9_8 = 0$
11)	$X18_6 + X18_5 + X18_17 + X18_13 + X18_10 + X18_16 + X18_8 + X18_9 - X1_18 - X6_18 - X6_18 - X5_18 - X17_18 - X10_18 - X16_18 - X8_18 - X9_18 = 0$
12)	$F + X9_6 + X9_16 + X9_8 + X9_18 - X1_9 - X22_9 - X13_9 - X10_9 - X18_9 = 0$
13)	Inicio de restricciones constantes
	...
92)	Final de restricciones constantes
END	

4.2.2. Gallese - Carrillo

En segundo lugar vamos a considerar como nodo inicial a Gallese y como nodo final a Carrillo.

MAX F SUBJECT TO

- 2) $-F + X1_6 + X1_22 + X1_5 + X1_17 + X1_13 + X1_10 + X1_18 + X1_9 - X22_1 - X5_1 - X17_1 - X13_1 = 0$
- 3) $X6_1 + X6_22 + X6_17 + X6_13 + X6_10 + X6_16 + X6_8 + X6_18 - X1_6 - X22_6 - X10_6 - X16_6 - X18_6 - X9_6 = 0$
- 4) $X22_1 + X22_6 + X22_5 + X22_13 + X22_10 + X22_16 + X22_8 + X22_18 + X22_9 - X1_22 - X6_22 - X5_22 - X17_22 - X13_22 - X10_22 - X16_22 - X8_22 = 0$
- 5) $X5_1 + X5_22 + X5_17 + X5_13 + X5_10 + X5_8 + X5_18 - X1_5 - X22_5 - X17_5 - X13_5 - X8_5 - X18_5 = 0$
- 6) $X17_1 + X17_22 + X17_5 + X17_13 + X17_10 + X17_8 + X17_18 - X1_17 - X6_17 - X5_17 - X13_17 - X10_17 - X8_17 - X18_17 = 0$
- 7) $X13_1 + X13_22 + X13_5 + X13_17 + X13_10 + X13_16 + X13_8 + X13_9 - X1_13 - X6_13 - X22_13 - X5_13 - X17_13 - X10_13 - X16_13 - X8_13 - X18_13 = 0$
- 8) $X10_6 + X10_22 + X10_17 + X10_13 + X10_16 + X10_8 + X10_18 + X10_9 - X1_10 - X6_10 - X22_10 - X5_10 - X17_10 - X13_10 - X16_10 - X8_10 - X18_10 = 0$
- 9) $X16_6 + X16_22 + X16_13 + X16_10 + X16_8 + X16_18 - X6_16 - X22_16 - X13_16 - X10_16 - X8_16 - X18_16 - X9_16 = 0$
- 10) $X8_22 + X8_5 + X8_17 + X8_13 + X8_10 + X8_16 + X8_18 - X6_8 - X22_8 - X5_8 - X17_8 - X13_8 - X10_8 - X16_8 - X18_8 - X9_8 = 0$
- 11) $F + X18_6 + X18_5 + X18_17 + X18_13 + X18_10 + X18_16 + X18_8 + X18_9 - X1_18 - X6_18 - X6_18 - X5_18 - X17_18 - X10_18 - X16_18 - X8_18 - X9_18 = 0$
- 12) $X9_6 + X9_16 + X9_8 + X9_18 - X1_9 - X22_9 - X13_9 - X10_9 - X18_9 = 0$
- 13) Inicio de restricciones constantes
...
92) Final de restricciones constantes

4.2.3. Gallese - Cueva

Finalmente vamos a considerar como nodo inicial a Gallese y como nodo final a Cueva.

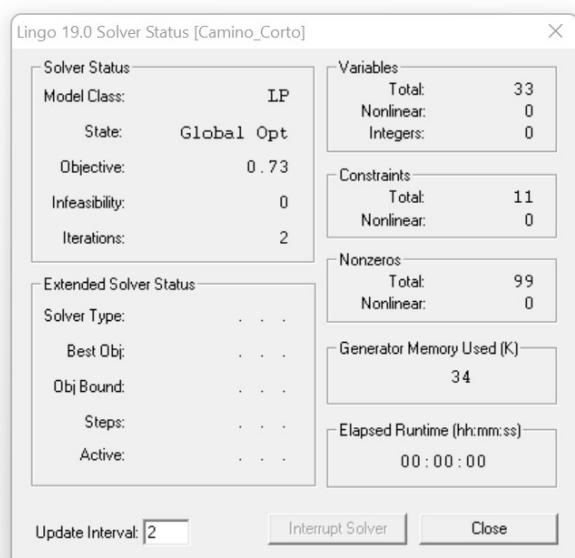
MAX	F
	SUBJECT TO
	<ol style="list-style-type: none"> 2) $-F + X1_6 + X1_22 + X1_5 + X1_17 + X1_13 + X1_10 + X1_18 + X1_9 - X22_1 - X5_1 - X17_1 - X13_1 = 0$ 3) $X6_1 + X6_22 + X6_17 + X6_13 + X6_10 + X6_16 + X6_8 + X6_18 - X1_6 - X22_6 - X10_6 - X16_6 - X18_6 - X9_6 = 0$ 4) $X22_1 + X22_6 + X22_5 + X22_13 + X22_10 + X22_16 + X22_8 + X22_18 + X22_9 - X1_22 - X6_22 - X5_22 - X17_22 - X13_22 - X10_22 - X16_22 - X8_22 = 0$ 5) $X5_1 + X5_22 + X5_17 + X5_13 + X5_10 + X5_8 + X5_18 - X1_5 - X22_5 - X17_5 - X13_5 - X8_5 - X18_5 = 0$ 6) $X17_1 + X17_22 + X17_5 + X17_13 + X17_10 + X17_8 + X17_18 - X1_17 - X6_17 - X5_17 - X13_17 - X10_17 - X8_17 - X18_17 = 0$ 7) $X13_1 + X13_22 + X13_5 + X13_17 + X13_10 + X13_16 + X13_8 + X13_9 - X1_13 - X6_13 - X22_13 - X5_13 - X17_13 - X10_13 - X16_13 - X8_13 - X18_13 = 0$ 8) $F + X10_6 + X10_22 + X10_17 + X10_13 + X10_16 + X10_8 + X10_18 + X10_9 - X1_10 - X6_10 - X22_10 - X5_10 - X17_10 - X13_10 - X16_10 - X8_10 - X18_10 = 0$ 9) $X16_6 + X16_22 + X16_13 + X16_10 + X16_8 + X16_18 - X6_16 - X22_16 - X13_16 - X10_16 - X8_16 - X18_16 - X9_16 = 0$ 10) $X8_22 + X8_5 + X8_17 + X8_13 + X8_10 + X8_16 + X8_18 - X6_8 - X22_8 - X5_8 - X17_8 - X13_8 - X10_8 - X16_8 - X18_8 - X9_8 = 0$ 11) $X18_6 + X18_5 + X18_17 + X18_13 + X18_10 + X18_16 + X18_8 + X18_9 - X1_18 - X6_18 - X6_18 - X5_18 - X17_18 - X10_18 - X16_18 - X8_18 - X9_18 = 0$ 12) $X9_6 + X9_16 + X9_8 + X9_18 - X1_9 - X22_9 - X13_9 - X10_9 - X18_9 = 0$ 13) Inicio de restricciones constantes ... 92) Final de restricciones constantes
END	

5. Resultados

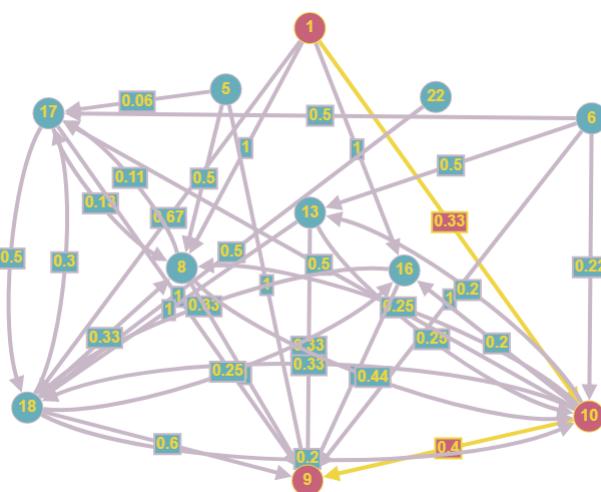
5.1. Ruta más corta

5.1.1. Gallese - Lapadula

Variable	Value	Reduced Cost
X1_10	1.000000	0.000000
X1_16	0.000000	0.6700000
X1_8	0.000000	0.6700000
X1_18	0.000000	0.3400000
X6_17	0.000000	0.5000000
X6_13	0.000000	0.3000000
X6_10	0.000000	0.2200000
X6_9	0.000000	0.6000000
X22_18	0.000000	0.0000000
X5_17	0.000000	0.6000000E-01
X5_8	0.000000	0.5000000
X5_9	0.000000	0.6000000
X17_8	0.000000	0.1300000
X17_18	0.000000	0.5000000
X17_9	0.000000	0.6000000
X13_10	0.000000	0.4500000
X13_18	0.000000	1.200000
X13_9	0.000000	0.1300000
X10_17	0.000000	0.5000000
X10_13	0.000000	0.0000000
X10_16	0.000000	0.2000000
X10_8	0.000000	0.2500000
X10_18	0.000000	0.3300000
X10_9	1.000000	0.0000000
X16_18	0.000000	0.3300000
X16_9	0.000000	0.6000000
X8_17	0.000000	0.1100000
X8_10	0.000000	0.4400000
X8_9	0.000000	0.6000000
X18_17	0.000000	0.3000000
X18_10	0.000000	0.2000000
X18_16	0.000000	0.2500000
X18_8	0.000000	0.3300000
X18_9	0.000000	0.2000000



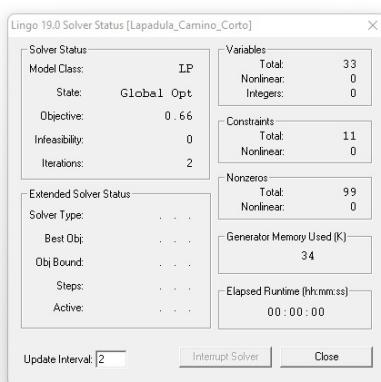
Reporte



Ruta más corta Gallese - Lapadula

5.1.2. Gallese - Carrillo

Variable	Value	Reduced Cost
X1_10	1.000000	0.000000
X1_16	0.000000	0.670000
X1_8	0.000000	0.670000
X1_18	0.000000	0.100000E-01
X6_17	0.000000	0.500000
X6_13	0.000000	0.300000
X6_10	0.000000	0.220000
X6_9	0.000000	1.000000
X22_18	0.000000	0.000000
X5_17	0.000000	0.000000
X5_8	0.000000	0.440000
X5_9	0.000000	0.940000
X17_8	0.000000	0.130000
X17_18	0.000000	0.170000
X17_9	0.000000	1.000000
X13_10	0.000000	0.450000
X13_18	0.000000	0.870000
X13_9	0.000000	0.530000
X10_17	0.000000	0.500000
X10_13	0.000000	0.000000
X10_16	0.000000	0.200000
X10_8	0.000000	0.250000
X10_18	1.000000	0.000000
X10_9	0.000000	0.400000
X16_18	0.000000	0.000000
X16_9	0.000000	1.000000
X8_17	0.000000	0.110000
X8_10	0.000000	0.440000
X8_9	0.000000	1.000000
X18_17	0.000000	0.630000
X18_10	0.000000	0.530000
X18_16	0.000000	0.580000
X18_8	0.000000	0.660000
X18_9	0.000000	0.930000



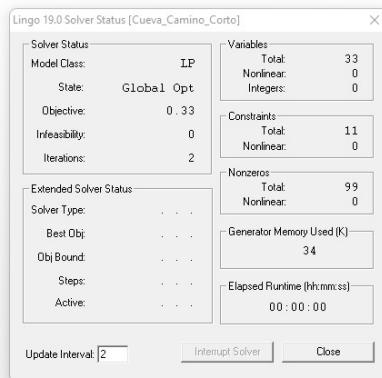
Reporte



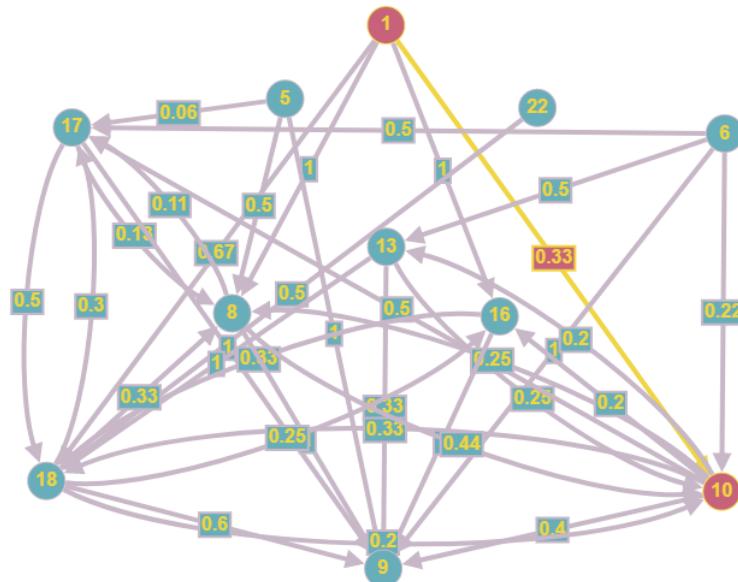
Ruta más corta Gallese - Carrillo

5.1.3. Gallese - Cueva

Variable	Value	Reduced Cost
X1_10	1.000000	0.000000
X1_16	0.000000	1.200000
X1_8	0.000000	0.870000
X1_18	0.000000	0.540000
X6_17	0.000000	0.500000
X6_13	0.000000	0.100000
X6_10	0.000000	0.200000E-01
X6_9	0.000000	1.000000
X22_18	0.000000	0.000000
X5_17	0.000000	0.000000
X5_8	0.000000	0.440000
X5_9	0.000000	0.840000
X17_8	0.000000	0.130000
X17_18	0.000000	0.500000
X17_9	0.000000	1.000000
X13_10	0.000000	0.450000
X13_18	0.000000	1.400000
X13_9	0.000000	0.730000
X10_17	0.000000	0.700000
X10_13	0.000000	0.000000
X10_16	0.000000	0.730000
X10_8	0.000000	0.450000
X10_18	0.000000	0.530000
X10_9	0.000000	0.600000
X16_18	0.000000	0.000000
X16_9	0.000000	0.670000
X8_17	0.000000	0.110000
X5_10	0.000000	0.240000
X8_9	0.000000	1.000000
X18_17	0.000000	0.300000
X18_10	0.000000	0.000000
X18_16	0.000000	0.580000
X18_8	0.000000	0.330000
X18_9	0.000000	0.600000



Reporte



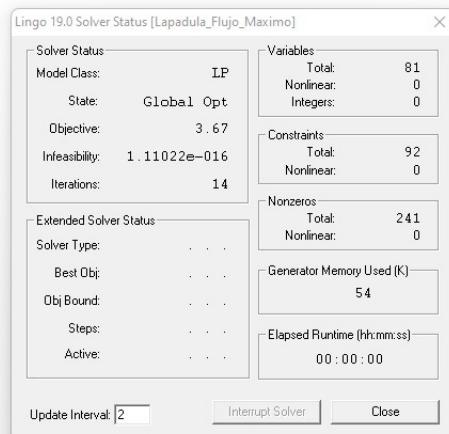
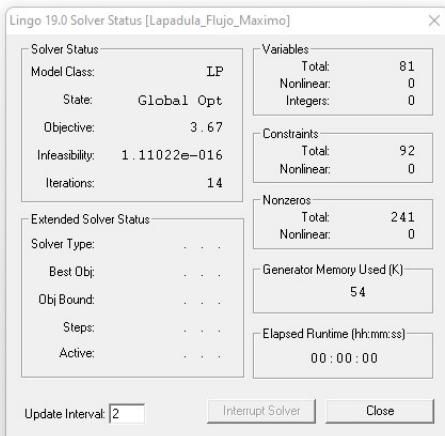
Ruta más corta Gallese - Cueva

5.2. Flujo máximo

5.2.1. Gallese - Lapadula

Variable	Value	Reduced Cost
F	3.670000	0.000000
X1_6	0.000000	0.000000
X1_22	1.000000	0.000000
X1_5	1.000000	0.000000
X1_17	1.000000	0.000000
X1_13	1.000000	0.000000
X1_10	0.000000	0.000000
X1_18	0.000000	0.000000
X1_9	1.000000	0.000000
X22_1	0.000000	0.000000
X5_1	0.750000	0.000000
X17_1	0.000000	0.000000
X13_1	0.580000	0.000000
X6_1	0.000000	0.000000
X6_22	1.000000	0.000000
X6_17	0.000000	0.000000
X6_13	0.000000	0.000000
X6_10	0.000000	0.000000
X6_16	1.000000	0.000000
X6_8	0.000000	0.000000
X6_18	0.000000	0.000000
X22_6	0.000000	0.000000
X10_6	0.870000	0.000000
X16_6	0.130000	0.000000
X18_6	1.000000	0.000000
X5_6	0.000000	1.000000
X22_5	1.000000	0.000000
X22_13	1.000000	0.000000
X22_10	1.000000	0.000000
X22_16	0.000000	0.000000
X22_8	0.000000	0.000000
X22_18	0.000000	0.000000
X22_9	1.000000	0.000000
X5_22	0.000000	0.000000
X17_22	0.000000	0.000000
X13_22	0.000000	0.000000
X10_22	0.000000	0.000000
X16_22	1.000000	0.000000

X8_22	1.000000	0.000000
X5_17	0.940000	0.000000
X5_13	1.000000	0.000000
X5_10	0.000000	0.000000
X5_8	0.000000	0.000000
X5_18	1.000000	0.000000
X17_5	0.000000	0.000000
X13_5	0.000000	0.000000
X8_5	1.000000	0.000000
X10_5	0.690000	0.000000
X17_13	1.000000	0.000000
X17_10	1.000000	0.000000
X17_8	0.880000	0.000000
X17_18	0.000000	0.000000
X13_17	0.000000	0.000000
X10_17	0.000000	0.000000
X8_17	0.630000	0.000000
X18_17	0.310000	0.000000
X13_10	0.750000	0.000000
X13_16	1.000000	0.000000
X13_8	1.000000	0.000000
X13_9	0.670000	0.000000
X10_13	0.000000	0.000000
X16_13	0.000000	0.000000
X8_13	0.000000	0.000000
X16_13	0.000000	0.000000
X10_16	0.800000	0.000000
X10_8	0.750000	0.000000
X10_18	0.000000	0.000000
X10_9	0.600000	0.000000
X16_10	0.000000	0.000000
X8_10	0.000000	0.000000
X18_10	0.270000	0.000000
X16_8	1.000000	0.000000
X16_19	0.670000	0.000000
X8_16	0.000000	0.000000
X18_16	0.000000	0.000000
X9_16	0.000000	1.000000
X8_18	1.000000	0.000000
X10_8	0.000000	0.000000
X9_8	0.000000	1.000000
X10_9	0.400000	0.000000
X9_18	0.000000	1.000000



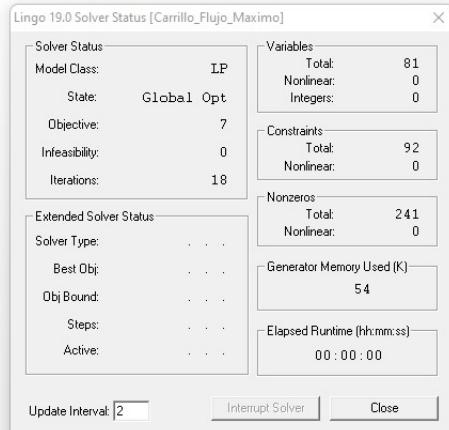
Reporte



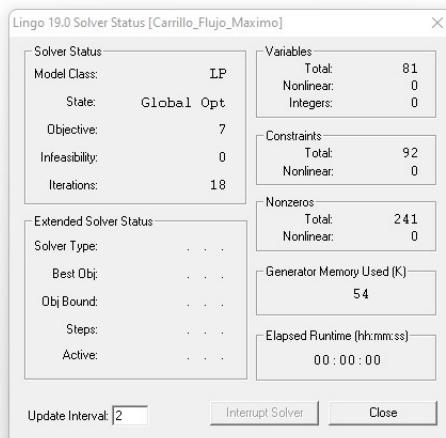
Flujo máximo Gallese - Lapadula

5.2.2. Gallese - Carrillo

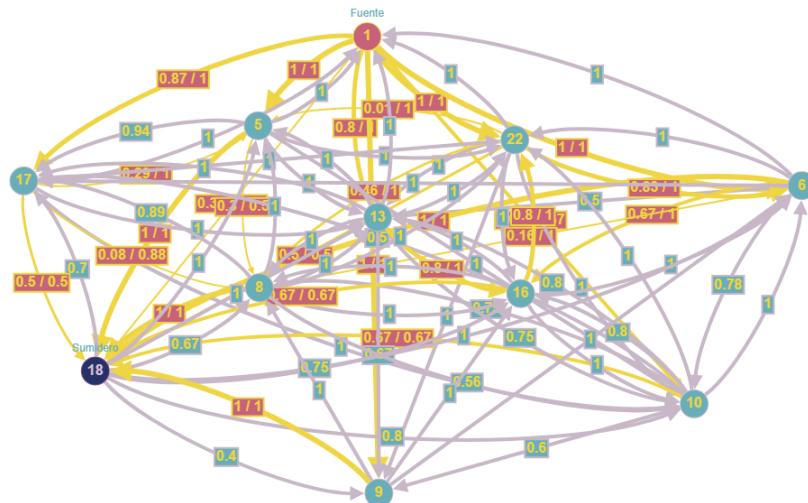
Variable	Value	Reduced Cost
F	7.000000	0.000000
X1_6	1.000000	0.000000
X1_22	1.000000	0.000000
X1_5	1.000000	0.000000
X1_17	1.000000	0.000000
X1_13	1.000000	0.000000
X1_10	0.6700000	0.000000
X1_18	0.3300000	0.000000
X1_9	1.000000	0.000000
X22_1	0.000000	1.000000
X5_1	0.000000	1.000000
X17_1	0.000000	1.000000
X13_1	0.000000	1.000000
X6_1	0.9150000	0.000000
X2_22	1.000000	0.000000
X6_17	0.000000	0.000000
X6_13	0.000000	0.000000
X6_10	0.000000	0.000000
X6_16	0.000000	0.000000
X6_8	0.000000	0.000000
X6_18	0.9150000	0.000000
X22_6	0.000000	0.000000
X10_6	1.000000	0.000000
X16_6	0.8300000	0.000000
X18_6	0.000000	0.000000
X9_6	0.000000	0.000000
X22_5	1.000000	0.000000
X22_13	1.000000	0.000000
X22_10	1.000000	0.000000
X22_16	1.000000	0.000000
X22_8	0.000000	0.000000
X22_18	0.000000	0.000000
X22_9	0.000000	0.000000
X5_22	0.000000	0.000000
X17_22	0.000000	0.000000
X13_22	0.000000	0.000000
X10_22	0.2000000	0.000000
X16_22	0.8000000	0.000000
X8_22	1.000000	0.000000
X5_17	0.9400000	0.000000
X5_13	1.000000	0.000000
X5_10	0.000000	0.000000
X5_8	0.000000	0.000000



X5_18	1.000000	0.000000
X17_5	0.000000	0.000000
X13_5	0.000000	0.000000
X8_5	0.9400000	0.000000
X18_5	0.000000	0.000000
X17_13	0.000000	0.000000
X17_10	1.000000	0.000000
X17_8	0.8800000	0.000000
X17_18	0.5000000	0.000000
X13_17	0.2500000	0.000000
X10_17	0.000000	0.000000
X8_17	0.1900000	0.000000
X18_17	0.000000	0.000000
X13_10	0.7500000	0.000000
X13_16	1.000000	0.000000
X13_8	1.000000	0.000000
X13_9	0.000000	0.000000
X10_13	0.000000	0.000000
X16_13	0.000000	0.000000
X8_13	0.000000	0.000000
X18_13	0.000000	0.000000
X10_16	0.8000000	0.000000
X10_8	0.7500000	0.000000
X10_18	0.6700000	0.000000
X10_9	0.000000	0.000000
X16_10	0.000000	0.000000
X8_10	0.000000	0.000000
X18_10	0.000000	0.000000
X16_8	1.000000	0.000000
X16_18	0.6700000	0.000000
X8_16	0.5000000	0.000000
X18_16	0.000000	0.000000
X9_16	0.000000	0.000000
X8_18	1.000000	0.000000
X18_8	0.000000	0.000000
X9_8	0.000000	0.000000
X18_9	0.000000	0.000000
X9_18	1.000000	0.000000



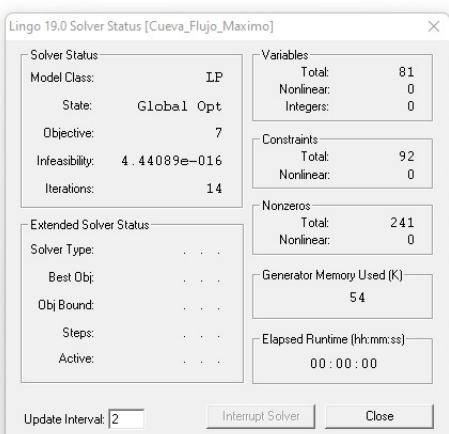
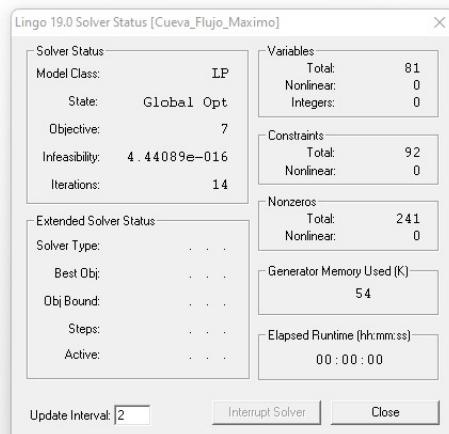
Reporte



Flujo máximo Gallese - Carrillo

5.2.3. Gallese - Cueva

Variable	Value	Reduced Cost
F	7.000000	0.000000
X1_6	1.000000	0.000000
X1_22	1.000000	0.000000
X1_5	1.000000	0.000000
X1_17	1.000000	0.000000
X1_13	1.000000	0.000000
X1_10	0.6700000	0.000000
X1_18	0.3300000	0.000000
X1_9	1.000000	0.000000
X2_1	0.000000	1.000000
X5_1	0.000000	1.000000
X17_1	0.000000	1.000000
X13_1	0.000000	1.000000
X6_1	0.000000	0.000000
X6_22	1.000000	0.000000
X6_17	0.000000	0.000000
X6_13	0.000000	0.000000
X6_10	0.7800000	0.000000
X6_16	1.000000	0.000000
X6_8	0.000000	0.000000
X6_18	0.000000	0.000000
X2_6	0.000000	0.000000
X10_6	0.000000	0.000000
X16_6	0.000000	0.000000
X18_6	0.7800000	0.000000
X9_6	1.000000	0.000000
X22_5	1.000000	0.000000
X22_13	1.000000	0.000000
X22_10	1.000000	0.000000
X22_16	1.000000	0.000000
X22_8	0.000000	0.000000
X22_18	0.000000	0.000000
X22_9	0.000000	0.000000
X5_22	0.000000	0.000000
X17_22	0.000000	0.000000
X13_22	0.000000	0.000000
X10_22	0.000000	0.000000
X16_22	1.000000	0.000000
X8_22	1.000000	0.000000
X5_17	0.9400000	0.000000
X5_13	1.000000	0.000000
X5_10	0.4400000	0.000000
X5_8	0.5000000	0.000000
X5_18	0.000000	0.000000
X17_5	0.000000	0.000000
X13_5	0.000000	0.000000
X8_5	0.5500000	0.000000
X18_5	0.3300000	0.000000
X17_13	0.000000	0.000000
X17_10	1.000000	0.000000
X17_8	0.8800000	0.000000
X17_18	0.5000000	0.000000
X13_17	0.2500000	0.000000
X10_17	0.000000	0.000000
X8_17	0.000000	0.000000
X18_17	0.1900000	0.000000
X13_10	0.7500000	0.000000
X13_16	1.000000	0.000000
X13_8	1.000000	0.000000
X13_9	0.000000	0.000000
X10_13	0.000000	0.000000
X16_13	0.000000	0.000000
X8_13	0.000000	0.000000
X18_13	0.000000	0.000000
X10_16	0.000000	0.000000
X10_8	0.000000	0.000000
X10_18	0.000000	0.000000
X10_9	0.000000	0.000000
X16_10	1.000000	0.000000
X8_10	0.5600000	0.000000
X18_10	0.8000000	0.000000
X16_8	1.000000	0.000000
X16_18	0.6700000	0.000000
X8_16	0.2700000	0.000000
X18_16	0.000000	0.000000
X9_16	0.4000000	0.000000
X8_18	1.000000	0.000000
X18_8	0.000000	0.000000
X5_8	0.000000	0.000000
X18_9	0.4000000	0.000000
X9_18	0.000000	0.000000



Reporte



Flujo máximo Gallese - Cueva

6. Discusiones

- De los resultados se obtuvieron 3 rutas cortas las cuales significan jugadas de peligro, lo cual conlleva probablemente un gran ocasión de gol, estas son:
 - i) Gallese, Cueva y Carrillo.
 - ii) Gallese y Cueva.
 - iii) Gallese, Cueva y Lapadula.
- De las jugadas mencionadas anteriormente, durante el partido no se presento dichas combinaciones. también, que en el jugador **Cueva** participa en las 3 jugadas, esto nos quiere decir que el jugador influyente en el partido para Perú fue **Cueva**. Bajo esta lógica, cabe recalcar que **Gallese** no es tomado en cuenta ya que antes fué definido como nodo inicial.
- Sin embargo, el jugador **Cueva** durante el partido Perú v Australia presentó un rendimiento inferior a lo usual, lo cual explica porqué no se pudieron concretar dichas jugadas, esto se puede ver con los datos obtenidos y la evaluación correspondiente al jugador en la pagina web <https://www.sofascore.com/peru-australia/QUbsPVb>.

7. Conclusiones

- Es correcto afirmar que la programación lineal puede explicar el porqué Perú no le ganó a Australia usando el lenguaje de programación Lindo como herramienta.
- Concluimos que el fracaso del combinado peruano se debe a que no se evidenció las jugadas antes mencionadas durante el partido de repesca por el Mundial Qatar 2022 entre Perú vs Australia.
- Así mismo, concluimos que **Cueva** fue el jugador más influyente para Perú.

8. Referencias

- Mokhtar S. Bazaraa, John J. Jarvis, Hanif D. Sherali *Linear Programming and Network Flows-John Wiley and Sons (2009)*
- Edsger W. Dijkstra *numerische mathematikc 1 (1959)*
- L. R. Ford, JR. y D. R. Fulkerson *Maximal Flow Through a Network (1956)*
- Jonatán W. Rojas *Teoría de redes, ruta más corta (2020)*
- Jonatán W. Rojas, Wilmer Atoche, Eduardo Carbajal y Miguel Fernandez *Teoría de redes, flujo máximo (2020)*
- LINDO system Inc. <https://ampl.com/>