

Parcial 3

Optimización 1

Thomas Martinod¹

¹Ingeniería Matemática, Universidad EAFIT, Colombia, tmartinods@eafit.edu.co

30 de octubre de 2023

Resumen

El presente documento corresponde a la entrega del tercer parcial del curso Optimización 1, impartido en el pregrado en Ingeniería Matemática en la Universidad EAFIT. En el presente, se proponen tres problemas clásicos de la optimización en redes, un problema de transporte, y dos problemas de flujo máximo. En la resolución de los problemas, se plantea el modelo matemático del problema y se emplea un algoritmo iterativo para llegar al óptimo global.

Nota: El archivo .xlsx referenciado y este documento se ubican en la carpeta **P3** del repositorio, <https://github.com/thomas-martinod/optimizacion-1>.

1. Problema de transporte

Problema 1. Suponga que una empresa de producción de helados está interesada en enviar desde sus tres plantas a sus cuatro franquicias los helados de la semana. La siguiente tabla relaciona la cantidad de helados que puede enviar de las diferentes plantas a sus franquicias.

Plantas ↓	Franquicia 1	Franquicia 2	Franquicia 3	Franquicia 4	Oferta
1	10	2	20	11	15
2	12	7	9	20	25
3	4	14	16	18	10
Demanda	5	15	15	15	

Cuadro 1: Distribución de plantas y franquicias de una empresa de producción de helados.

Construya el grafo que represente el problema, plantee el modelo de forma algebraica y determine las cantidades de helados que se deben enviar de cada planta a cada franquicia logrando el menor costo posible.

Solución.

Inicialmente, el grafo $G = (N, A)$ que representa este problema de transporte se expone en la figura 1, donde se definió P_i como la i -ésima planta de producción y F_j como la j -ésima franquicia.

En un problema genérico de transporte como el presentado en la figura 1, se considera un grafo $G = (N, A)$ con un conjunto de $n = |N|$ nodos de partida con ofertas $a_i \geq 0$, con $i \in \mathcal{O}$ (el conjunto

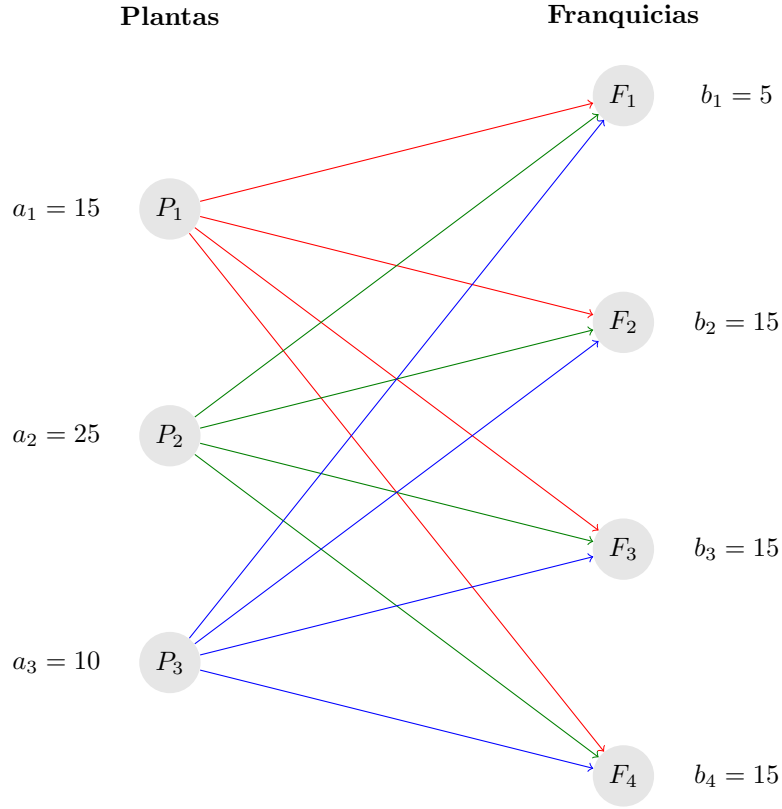


Figura 1: Grafo del problema de transporte asociado al cuadro 1.

de índices de las plantas) y m nodos de llegada con demandas $b_j \geq 0$ con $j \in \mathcal{D}$ (el conjunto de índices de las franquicias).

Además, a cada arco $(i, j) \in A$ se le asocian un costo unitario de transporte c_{ij} constante, y un flujo neto x_{ij} como la cantidad de producto que va desde la fuente de índice $i \in \mathcal{O}$ hasta la franquicia de índice $j \in \mathcal{D}$. Como se asume que un producto va de planta a franquicia y no al revés, debe cumplirse que $x_{ij} \geq 0$ para todo origen y destino.

Como observación, los costos unitarios, que representan la inversión que conlleva llevar una unidad de flujo desde la planta i hasta la franquicia j se pueden observar en el cuadro 1, como la intersección de la i -ésima fila con la j -ésima columna.

Otro aspecto crucial del problema de transporte es que este tipo de problema se caracteriza por mantener un equilibrio entre la oferta y la demanda. Matemáticamente, esta igualdad se expresa de la siguiente manera:

$$\sum_{i \in \mathcal{O}} a_i = \sum_{j \in \mathcal{D}} b_j \quad (1)$$

Esta ecuación se interpreta como una “ley de conservación de la masa”, lo que significa que ningún producto debe perderse ni generarse espontáneamente. En este problema en particular, podemos comprobar que esto se cumple, ya que $15 + 25 + 10 = 50 = 5 + 15 + 15 + 15$.

Ahora, teniendo en cuenta los nm arcos y sus flujos respectivos x_{ij} , que son las variables de decisión, como el problema se basa en minimizar el costo total del transporte, en su forma general, el problema de transporte se puede escribir como:

$$\begin{aligned}
 \text{mín} \quad & \sum_{i \in \mathcal{O}} \sum_{j \in \mathcal{D}} c_{ij} x_{ij} \\
 \text{s.a.} \quad & \sum_{j \in \mathcal{D}} x_{ij} = a_i, \quad i \in \mathcal{O} \\
 & \sum_{i \in \mathcal{O}} x_{ij} = b_j, \quad j \in \mathcal{D} \\
 & x_{ij} \geq 0, \quad \forall i \in \mathcal{O}, \quad \forall j \in \mathcal{D}.
 \end{aligned} \tag{2}$$

Donde la primera y segunda restricción corresponden a que en los nodos origen y destino se respete la oferta y demanda que los caracteriza respectivamente.

Aplicando esta definición general, en el caso del problema propuesto en 1, se tiene que el conjunto de índices de los orígenes es $\mathcal{O} = \{1, 2, 3\}$, mientras que el conjunto de índices de los destinos es $\mathcal{D} = \{1, 2, 3, 4\}$ como se aprecia en el grafo 1. Además, la matriz de costos unitarios viene dada por:

$$\mathbf{c} = [c_{ij}] = \begin{bmatrix} 10 & 2 & 20 & 11 \\ 12 & 7 & 9 & 20 \\ 4 & 14 & 16 & 18 \end{bmatrix}$$

y por ende, la dimensión de $\mathbf{x} = [x_{ij}]$ también es 3×4 . Similarmente, las ofertas a_i y demandas b_j se encuentran en el cuadro 1, y son los vectores:

$$\mathbf{a} = \begin{bmatrix} 15 \\ 25 \\ 20 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 5 \\ 15 \\ 15 \\ 15 \end{bmatrix}$$

luego, reemplazando todos estos datos en el problema 2, se obtiene el problema a solucionar:

$$\begin{aligned}
 (P) \quad \text{mín} \quad & Z = 10x_{11} + 2x_{12} + 20x_{13} + 11x_{14} + 12x_{21} + 7x_{22} \\
 & + 9x_{23} + 20x_{24} + 4x_{31} + 14x_{32} + 16x_{33} + 18x_{34} \\
 \text{s.a.} \quad & \sum_{i=1}^3 x_{i1} = 5, \quad \sum_{i=1}^3 x_{i2} = 15, \quad \sum_{i=1}^3 x_{i3} = 15, \quad \sum_{i=1}^3 x_{i4} = 15 \\
 & \sum_{j=1}^4 x_{1j} = 15, \quad \sum_{j=1}^4 x_{2j} = 25, \quad \sum_{j=1}^4 x_{3j} = 10 \\
 & x_{ij} \geq 0, \quad \forall i \in \{1, 2, 3\}, \quad \forall j \in \{1, 2, 3, 4\}
 \end{aligned} \tag{3}$$

Para solucionar este problema se hará uso de dos métodos iterativos vistos en clase. Al tratarse de un *toy problem* (de dimensión $nm = 12$), puede hallarse una solución factible cercana a la óptima usando el método de Vogel. Posteriormente se comparará este resultado con el obtenido mediante el método simplex en el solver de Excel.

1.1. Método de aproximación de Vogel

El método de Vogel es un método heurístico empleado para resolver problemas de asignación eficiente en la distribución de productos desde múltiples fuentes a varios destinos. Su objetivo es minimizar los

costos de transporte y logra hallar soluciones básicas factibles.

Sin embargo, este método tiene muchas limitaciones, como no garantizar siempre la convergencia hacia una solución óptima, ni ser adecuado para grandes conjuntos de datos. A continuación se empleará para hallar una solución básica factible al problema (P) y se describirá brevemente cada momento del algoritmo.

Como anotación, no se hace una descripción analítica del algoritmo en yuxtaposición a la extensiva descripción del algoritmo FF que se mostrará en la segunda sección, puesto que el método de Vogel no es de interés, en cuanto a que ser solución de este no es condición suficiente para ser óptimo global. Este método simplemente se usa para ejemplificar los algoritmos vistos durante el curso.

Paso 0. Equilibrio entre oferta y demanda: Dado que las cantidades ofrecidas por las plantas coinciden con las cantidades solicitadas por las franquicias (como se aprecia en la ecuación 1), no es necesario introducir plantas o franquicias artificiales. Por lo tanto, la tabla inicial, que muestra las ofertas, las demandas y los costos correspondientes, se presenta en la figura 2.

		Destinos				Oferta
		Franquicia 1	Franquicia 2	Franquicia 3	Franquicia 4	
Orígenes	Planta 1	10	2	20	11	15
	Planta 2	12	7	9	20	25
	Planta 3	4	14	16	18	10
Demanda		5	15	15	15	50

Figura 2: Tabla inicial del algoritmo de Vogel.

Paso 1.1. Cálculo de la primera penalización (P1) y asignación de costos: En el primer paso, se calculan las penalizaciones iniciales. Luego, se selecciona el costo unitario más bajo de la fila o columna con la penalización más alta. Posteriormente, se compara este costo con el valor menor entre la oferta y la demanda en la celda correspondiente. Luego, se asigna este valor a la celda y se descuenta de la oferta y la demanda para lograr un equilibrio.

Por último, se elimina la fila o columna que se haya agotado tras realizar la resta (pueden ser ambas). La tabla resultado de este primer proceso se aprecia en la figura 3.

		Destinos				Oferta
		Franquicia 1	Franquicia 2	Franquicia 3	Franquicia 4	
Orígenes	Planta 1	10	2	20	11	15
	Planta 2	12	7	9	20	25
	Planta 3	4	14	16	18	10
Demanda		5	15	15	15	45

P1	10 - 4 = 6	7 - 2 = 5	16 - 9 = 7	18 - 11 = 7
-----------	------------	-----------	------------	-------------

P1	10 - 2 = 8	9 - 7 = 2	14 - 4 = 10
-----------	------------	-----------	-------------

Figura 3: Cálculo de la primera penalización y primera iteración.

Paso 1.2. Iteración: Como la suma de ofertas y la suma de demandas son ambas no nulas, se debe seguir iterando el proceso descrito en el paso anterior, reduciendo la matriz en consideración. Particularmente, solo se considera la submatriz resultante después de eliminar la columna resaltada, como se muestra en la figura 4.

		Destinos				Oferta
		Franquicia 1	Franquicia 2	Franquicia 3	Franquicia 4	
Orígenes	Planta 1	10	2	20	11	15
	Planta 2	12	7	9	20	25
	Planta 3	4	14	16	18	10
Demanda		5	15	15	15	30

P1	P2
$10 - 2 = 8$	$11 - 2 = 9$
$9 - 7 = 2$	$9 - 7 = 2$
$14 - 4 = 10$	$16 - 14 = 2$

P1	$10 - 4 = 6$	$7 - 2 = 5$	$16 - 9 = 7$	$18 - 11 = 7$
P2		$7 - 2 = 5$	$16 - 9 = 7$	$18 - 11 = 7$

Figura 4: Cálculo de la segunda penalización y segunda iteración.

Nótese que, tras esta iteración, tanto la columna como la fila resaltadas se eliminan, ya que ambas se hacen cero después de realizar la iteración. Este hecho, reducirá en 1 la cantidad de asignaciones dispuestas al final del proceso, que ahora será $m + n - 2$.

Paso 1.3: Se continua con el proceso iterativo del algoritmo empleando la tercera penalización. Se obtiene la siguiente tabla:

		Destinos				Oferta
		Franquicia 1	Franquicia 2	Franquicia 3	Franquicia 4	
Orígenes	Planta 1	10	2	20	11	15
	Planta 2	12	7	9	20	25
	Planta 3	4	14	16	18	10
Demanda		5	15	15	15	15

P1	P2	P3
$10 - 2 = 8$	$11 - 2 = 9$	
$9 - 7 = 2$	$9 - 7 = 2$	$20 - 9 = 11$
$14 - 4 = 10$	$16 - 14 = 2$	$18 - 16 = 2$

P1	$10 - 4 = 6$	$7 - 2 = 5$	$16 - 9 = 7$	$18 - 11 = 7$
P2		$7 - 2 = 5$	$16 - 9 = 7$	$18 - 11 = 7$
P3			$16 - 9 = 7$	$20 - 18 = 2$

Figura 5: Cálculo de la tercera penalización y tercera iteración.

Paso 1.4: Se prosigue con la cuarta iteración del algoritmo usando, calculando la cuarta penalización. El tablero de Vogel luego de esta iteración se aprecia en la figura 6.

Paso 2: Se lleva a cabo la última asignación de acuerdo a la única celda que aún tiene penalización para completar la tabla. La tabla que muestra la última asignación y por ende la asignación de transporte final se muestra en la figura 7.

Paso 3: Finalmente se calcula el costo factible de Vogel Z_V calculado por el algoritmo como:

$$Z_V = 15 \cdot 2 + 15 \cdot 9 + 10 \cdot 20 + 5 \cdot 4 + 5 \cdot 18 = 475 \quad (4)$$

		Destinos				Oferta
		Franquicia 1	Franquicia 2	Franquicia 3	Franquicia 4	
Orígenes	Planta 1	10	15	20	11	15
	Planta 2	12	7	9	20	25
	Planta 3	4	14	16	18	10
Demanda		5	15	15	15	5

P1	P2	P3	P4
10 - 2 = 8	11 - 2 = 9		
9 - 7 = 2	9 - 7 = 2	20 - 9 = 11	20
14 - 4 = 10	16 - 14 = 2	18 - 16 = 2	18

P1	10 - 4 = 6	7 - 2 = 5	16 - 9 = 7	18 - 11 = 7
P2		7 - 2 = 5	16 - 9 = 7	18 - 11 = 7
P3			16 - 9 = 7	20 - 18 = 2
P4				20 - 18 = 2

Figura 6: Cálculo de la cuarta penalización y cuarta iteración.

		Destinos				Oferta
		Franquicia 1	Franquicia 2	Franquicia 3	Franquicia 4	
Orígenes	Planta 1	10	15	20	11	15
	Planta 2	12	7	9	20	25
	Planta 3	4	14	16	18	10
Demanda		5	15	15	15	5

Figura 7: Última asignación forzada del algoritmo y asignaciones finales.

Este punto factible se alcanza cuando las variables de decisión tienen los valores en rojo en la figura 7:

$$\mathbf{x}_V = [x_{ij}] = \begin{bmatrix} 0 & 15 & 0 & 0 \\ 0 & 0 & 15 & 10 \\ 5 & 0 & 0 & 5 \end{bmatrix}$$

Ahora bien, como se discutió en clase, la solución \mathbf{x}_V que alcanza este método es factible y como buen método heurístico, suele ser cercana a un óptimo local. En particular, para verificar la validez del resultado obtenido empleando el método de Vogel, como el problema es de dimensión pequeña, se puede verificar mediante el método simplex el óptimo absoluto de (P) .

Para ello, en el archivo `.xlsx` dispuesto en el github [1], se empleó el solver de Excel (método simplex) contando con las doce variables de decisión, las siete restricciones de igualdad y todas las condiciones de no negatividad. Luego, conociendo que el método simplex bajo condiciones de acotamiento y factibilidad siempre obtiene el óptimo global, puede ser afirmado que la solución al mínimo costo del problema (P) se da en:

$$\mathbf{x}^* = [x_{ij}^*] = \begin{bmatrix} 0 & 5 & 0 & 10 \\ 0 & 10 & 15 & 0 \\ 5 & 0 & 0 & 5 \end{bmatrix}$$

teniendo un valor de la función objetivo $Z^* = 435$, alcanzado al cabo de 11 iteraciones. Claramente se observó que el método de Vogel es aproximado, pero que su solución puede ser no tan desviada de la solución del problema. No obstante, en general se recomienda para solucionar este problema emplear el método generalizado primal-dual aplicado al problema de transporte, como se enseña en [2], o el método de los multiplicadores de Lagrange.

2. Problemas de flujo máximo

Problema 2. Usted ha sido contratado para revisar la viabilidad de una ruta que necesita garantizar que 12.000 carros por hora pueden ir del nodo 1 al nodo 7.

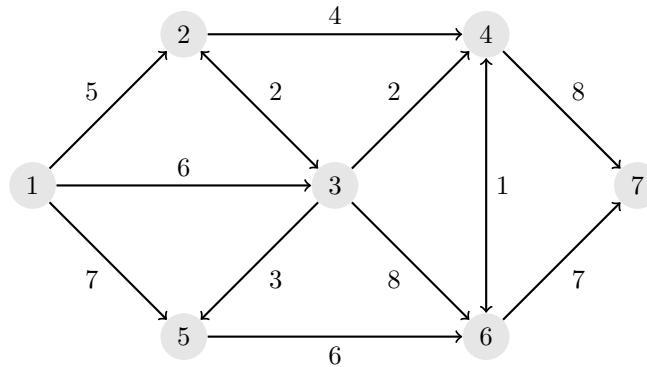


Figura 8: Grafo G del problema 2.

Los nodos intermedios hacen referencia a las vías por donde puede pasar un carro y el peso de las aristas a la cantidad en miles de carros que pueden ir por esa vía.

- ¿Puede usted garantizar que esta vía permite el flujo de 12 mil carros por hora?
- ¿Cuál es el flujo máximo de carros que pueden ir del nodo 1 al nodo 7 por hora?
- ¿Cuál es la ruta que permite más cantidad de vehículos por hora?

Solución.

En este problema y el siguiente, se empleará el enfoque del algoritmo de Ford y Fulkerson (algoritmo FF). La idea central de este algoritmo es construir de manera iterativa el flujo máximo en una red de flujo, partiendo de un flujo inicial factible. En cada iteración, se busca un camino que permita aumentar el flujo con respecto al flujo actual. Este aumento se basa en las capacidades residuales (incrementales) de los arcos que componen el camino.

Este proceso se repite hasta que ya no sea posible encontrar un camino de aumento de flujo entre el nodo fuente y el nodo destino. Cada iteración del algoritmo se divide en dos partes fundamentales:

1. Búsqueda de un camino de aumento de flujo: En esta etapa, se busca un camino en la red que permita incrementar el flujo en la dirección deseada. Este camino se elige teniendo en cuenta las capacidades residuales de los arcos.
2. Determinación de la cantidad de flujo mínimo: Una vez encontrado el camino de aumento, se calcula la cantidad máxima de flujo que puede ser transportada por este camino, limitada por las capacidades residuales de los arcos que lo componen.

Antes de pasar a la siguiente iteración, el flujo máximo determinado en la etapa anterior se descuenta de los caminos de aumento desarrollados, lo que garantiza que no se excedan las capacidades de los arcos.

Ahora formalmente, consideremos $G = (N, A)$ como el grafo al que se le desea hallar el flujo máximo. Además, sean f_{ij} el flujo que pasa por la arista $(i, j) \in A$, u_{ij} el flujo máximo que puede pasar por la misma arista, l_{ij} su respectivo flujo mínimo, F el flujo exógeno (neto) que entra al nodo 1 o sale por el nodo $n = 7$ (que se desea maximizar), y $f := (\{f_{ij}\}_{(i,j) \in A}, F)$ un **flujo factible**, el algoritmo FF se describe a continuación:

Algoritmo FF

- **Paso 0. Flujo inicial:** Hallar un flujo factible f inicial. Como observación importante, cuando $l_{ij} = 0$, para toda arista $(i, j) \in A$ (los flujos inferiores son nulos), un flujo factible consiste en tomar todos los flujos f_{ij} como nulos, y por ende $F = 0$.
- **Paso 1. Creación de un grafo auxiliar:** Para el grafo G se crea un grafo incremental $G' = (N', A')$ donde $N = N'$ y se definen $B_1 = B_2 = \emptyset$. Para determinar A' se examinan los arcos (i, j) de G de la siguiente forma:
 - Si $f_{ij} < u_{ij}$ en G , la arista (i, j) se incorpora a A' con capacidad incremental $\Delta_{ij} = u_{ij} - f_{ij}$, y se define $B_1 = B_1 \cup (i, j)$ recursivamente.
 - Si $f_{ij} > l_{ij}$ en G , la arista (j, i) se incorpora a A' con capacidad incremental $\Delta_{ij} = f_{ij} - l_{ij}$ y se define $B_2 = B_2 \cup (i, j)$ recursivamente,
 - Nótese que estas dos situaciones podrían ocurrir simultáneamente, creando 2 arcos en G' .
- **Paso 2. Criterio de parada:** Se busca un camino C' en G' desde el nodo 1 hasta el nodo n . Si no existe ningún camino en G' , entonces detener el algoritmo. La solución es óptima.
- **Paso 3. Camino y Aumento:** En caso que este camino C' exista (si son más de uno, se selecciona alguno), se determina el incremento ε de la siguiente forma:

$$\varepsilon = \min \left\{ \min_{(i,j) \in C \cap B_1} (u_{ij} - f_{ij}), \min_{(i,j) \in C \cap B_2} (f_{ij} - l_{ij}) \right\}$$

- **Paso 4. Nuevo flujo:** Para repetir el algoritmo desde el paso 1, se redefine:

$$\begin{aligned} F &\leftarrow F + \varepsilon \\ f_{ij} &\leftarrow f_{ij} + \varepsilon, \quad \text{si } (i, j) \in C \cap B_1 \\ f_{ij} &\leftarrow f_{ij} + \varepsilon, \quad \text{si } (i, j) \in C \cap B_2 \end{aligned}$$

Luego, repetir el paso 1 y continuar este proceso hasta la parada.

Sea $G = (N, A)$ el grafo expuesto en la figura 8, veamos la aplicación del algoritmo FF por iteraciones (solo se describe a profundidad la primera iteración, pero todas siguen la estructura del recuadro encima).

Como observaciones, se usará el superíndice entre paréntesis para denotar el número de iteración de cualquier objeto que se actualice durante el algoritmo, comenzando desde $k = 0$. Además, las flechas que entran al nodo 1 y salen del nodo $n = 7$ son meramente ilustrativas y muestran el flujo total (exógeno) en la k -ésima iteración $F^{(k)}$.

Iteración 0.

Como solamente se aclaran los límites de flujo máximo en cada arista, es posible asumir que $l_{ij} = 0, \forall (i, j) \in A$. Luego una solución factible inicial es tomar todos los flujos como nulos ($f_{ij} = 0$ en toda arista) y por ende $F = 0$. El grafo $G^{(0)}$ con esta primera solución factible se presenta en la figura 9.

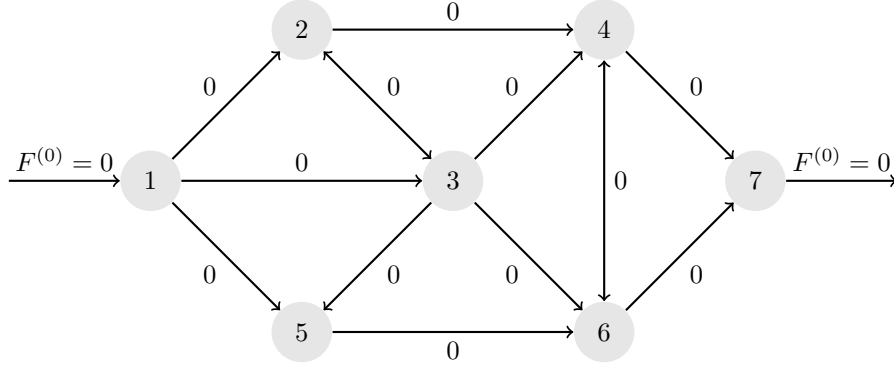


Figura 9: Grafo factible inicial $G^{(0)}$.

Ahora, el grafo auxiliar $G'^{(0)}$ se halla como es descrito en el primer paso del algoritmo. Como todos los flujos de la solución inicialmente son nulos, $\Delta_{ij} = u_{ij} - f_{ij} = u_{ij}$ para toda arista, obteniendo $G'^{(0)} = G$, $B_1 = A$ y $B_2 = \emptyset$. El grafo auxiliar de esta primera iteración se enseña en la figura 10.

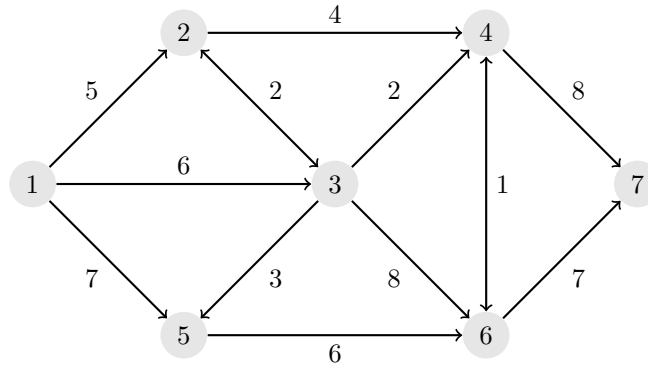


Figura 10: Grafo auxiliar inicial $G'^{(0)} = G$.

A continuación, se debe crear un camino $C'^{(0)}$ desde el nodo 1 hasta el nodo 7. En particular, si se toma $C'^{(0)} = 1 \rightarrow 5 \rightarrow 6 \rightarrow 7$ es posible calcular la mínima capacidad de transporte por ese camino para poder continuar a la siguiente iteración. Siguiendo el paso 2, $C'^{(0)} \cap B_1 = C'^{(0)}$ y $C'^{(0)} \cap B_2 = \emptyset$, por lo que $\varepsilon^{(0)} = \min\{7, 6, 7\} = 6$ que es el residual o la capacidad de incremento de flujo del grafo.

A continuación, se define:

$$F^{(1)} = F^{(0)} + \varepsilon^{(0)} = 6$$

$$f_{ij}^{(1)} = \varepsilon, \quad \text{si } (i, j) \in C'^{(0)}$$

y se procede con la siguiente iteración.

Iteración 1.

Al actualizar el grafo (ahora $G^{(1)}$) con el flujo actual, se obtiene el enseñado en la figura 11.

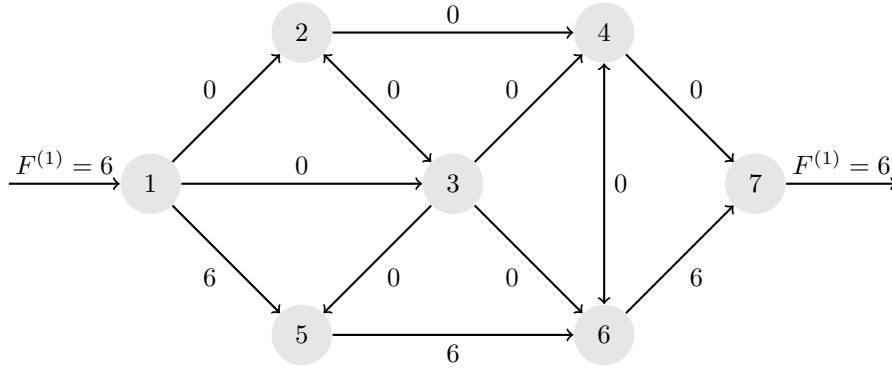


Figura 11: Grafo factible $G^{(1)}$.

Posteriormente se construye su grafo auxiliar $G'^{(1)}$ como se describió en el algoritmo FF, obteniendo el enseñado en la figura 12.

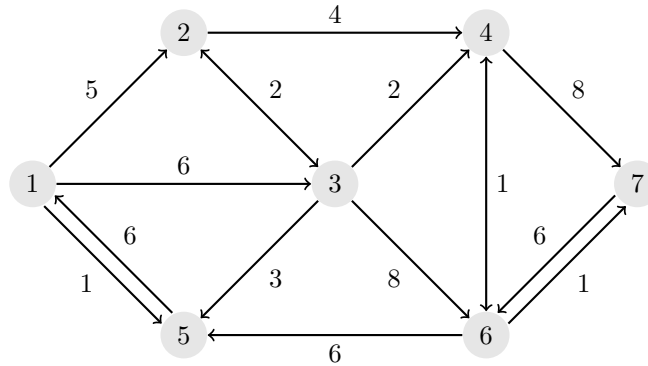


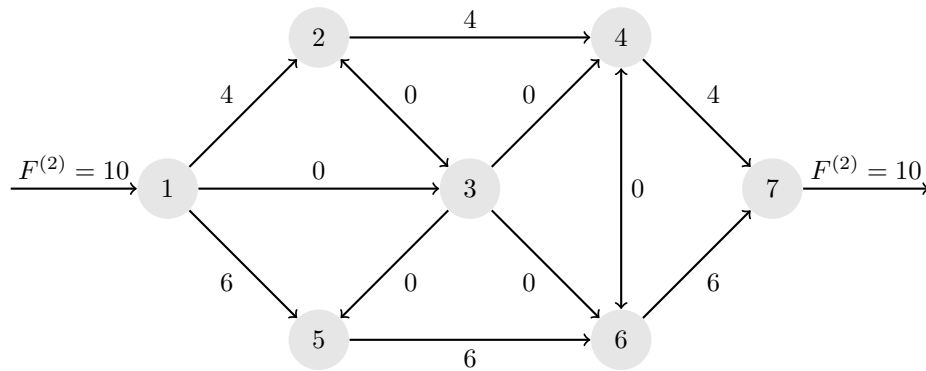
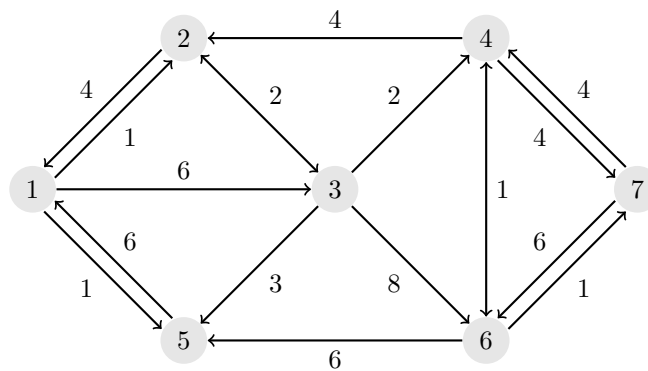
Figura 12: Grafo auxiliar $G'^{(1)}$.

Ahora, se empleará el camino $C'^{(1)} = 1 \rightarrow 2 \rightarrow 4 \rightarrow 7$, obteniendo $\varepsilon^{(1)} = 4$. Así, se actualizan los flujos de las aristas involucradas, y se obtiene para la siguiente iteración $F^{(2)} = 10$.

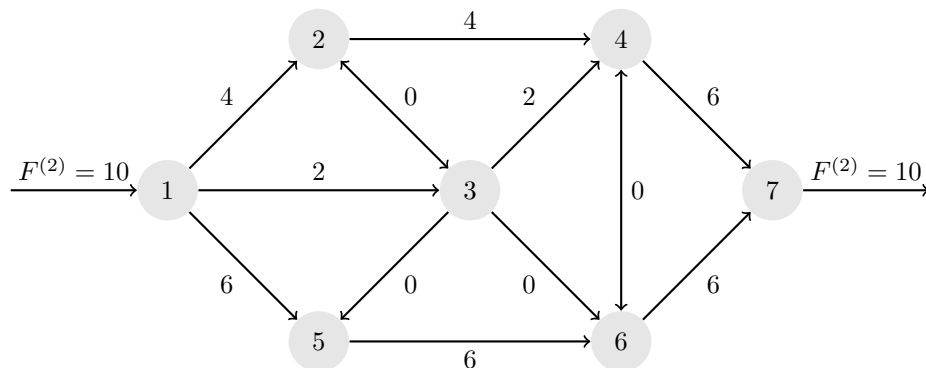
Iteración 2.

El proceso es repetitivo e intuitivo, al actualizar el grafo $G^{(2)}$ con el flujo actual, se obtiene el enseñado en la figura 13, y de forma análoga a la descripción anterior, se construye el grafo auxiliar $G'^{(2)}$, obteniendo el enseñado en 14.

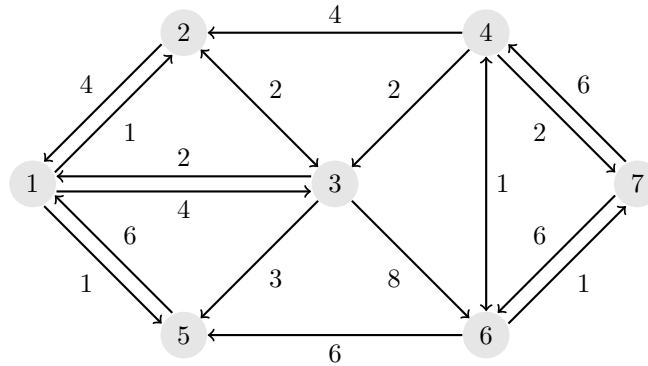
En esta iteración, se define el camino $C'^{(1)} = 1 \rightarrow 3 \rightarrow 4 \rightarrow 7$, obteniendo $\varepsilon^{(1)} = 2$. Así, se actualizan los flujos de las aristas involucradas, y se obtiene para la siguiente iteración $F^{(3)} = 12$.

Figura 13: Grafo factible $G^{(2)}$.Figura 14: Grafo auxiliar $G'^{(2)}$.**Iteración 3.**

Iterando, al actualizar el grafo $G^{(2)}$ con el flujo actual, se obtiene $G^{(3)}$, enseñado en la figura 15.

Figura 15: Grafo factible $G^{(3)}$.

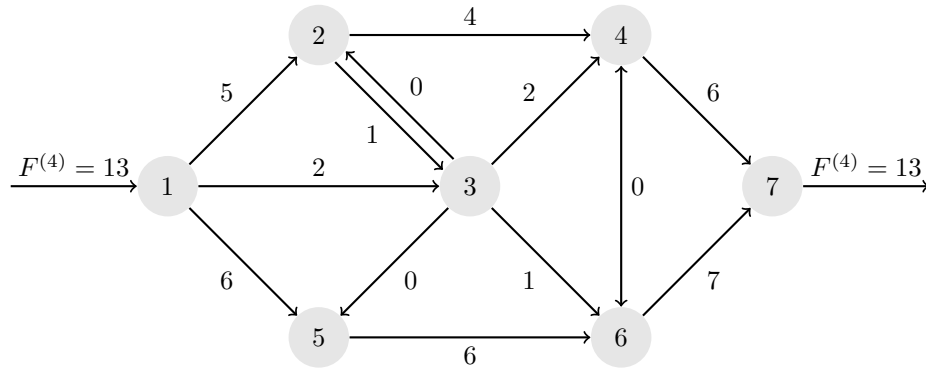
E idénticamente, se construye su grafo auxiliar $G'^{(3)}$ como se describió anteriormente, obteniendo el enseñado en 16.

Figura 16: Grafo factible $G^{(3)}$.

En esta iteración, se define el camino $C'^{(3)} = 1 \rightarrow 2 \rightarrow 3 \rightarrow 6 \rightarrow 7$, obteniendo $\varepsilon^{(3)} = 1$. Así, se actualizan los flujos de las aristas involucradas, y se obtiene para la siguiente iteración $F^{(4)} = 13$.

Iteración 4.

Iterando, al actualizar el grafo $G^{(4)}$ con el flujo actual, se obtiene el enseñado en la figura 17.

Figura 17: Grafo factible $G^{(4)}$.

Posteriormente se construye su grafo auxiliar $G'^{(4)}$ como se describió en el algoritmo FF, obteniendo el enseñado en la figura 18.

En esta iteración, se define el camino $C'^{(4)} = 1 \rightarrow 3 \rightarrow 6 \rightarrow 4 \rightarrow 7$, obteniendo $\varepsilon^{(4)} = 1$. Así, se actualizan los flujos de las aristas involucradas, y se obtiene para la siguiente iteración $F^{(5)} = 14$.

Iteración 5.

Iterando, al actualizar el grafo $G^{(4)}$ con el flujo actual, para obtener $G^{(5)}$, se obtiene el enseñado en la figura 19.

Siguendo el proceso idénticamente a como se hizo en pasos anteriores, se construye su grafo auxiliar $G'^{(5)}$, generando el enseñado en 20. Este grafo auxiliar tiene una peculiaridad, observando detenida-

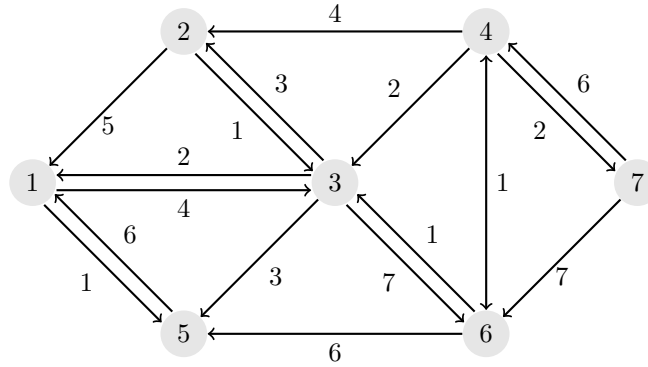


Figura 18: Grafo auxiliar $G'^{(4)}$.

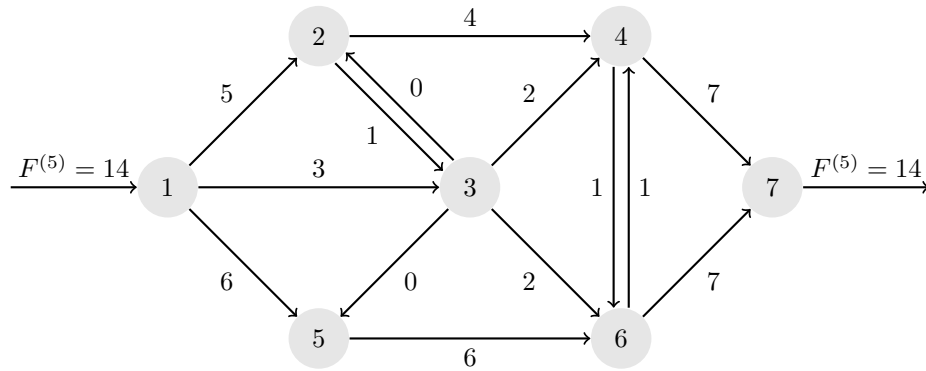


Figura 19: Grafo de flujo exógeno final $G^{(5)}$.

mente, es posible identificar que no existe ningún camino desde el nodo inicial 1 hasta el nodo final 7.

Particularmente, este es el criterio de parada del algoritmo FF, y lo que indica es que la solución actual no puede ser mejorada, es decir, no puede incrementarse ningún flujo f_{ij} sin violar las restricciones de flujo máximo y sin perjudicar el resultado anterior $F^{(5)} = 14$.

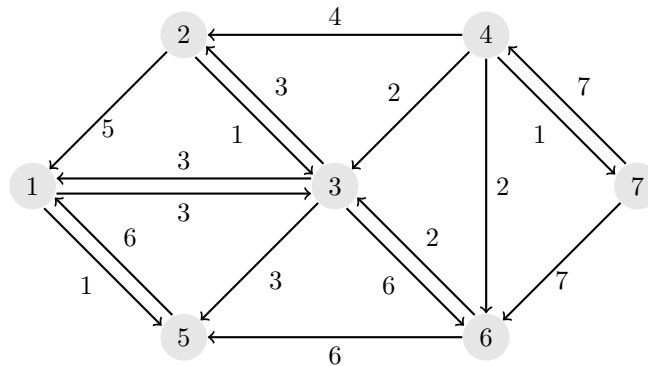


Figura 20: Grafo auxiliar $G''^{(5)}$.

Así, el flujo óptimo es el indicado en el último grafo de flujo factible (ver figura 19). En particular, el flujo máximo de personas entre el nodo 1 y 7 es de $F^* = F^{(5)} = 14$ mil vehículos. Además, para ejemplificar la interpretación del flujo sobre las aristas, para alcanzar este flujo óptimo, entre los nodos 3 y 5 no debe haber flujo alguno, es decir $f_{35}^* = f_{35}^{(5)} = 0$. A partir de esto, se resuelven a continuación las preguntas propuestas:

- (a) ¿Puede usted garantizar que esta vía permite el flujo de 12 mil carros por hora?

En efecto, es posible garantizarlo. Teniendo en cuenta que el flujo máximo permitido en la red son 14 mil carros (por unidad de tiempo), todo flujo inferior no negativo será factible. Quizá los flujos de las aristas no sean todos enteros, sin embargo al interpretarse como un problema temporal, es coherente hablar de vías con flujo de 7,5 carros por hora. Luego, sin imponer la entereza de las variables de decisión, se garantiza el flujo de 12 mil carros por hora.

- (b) ¿Cuál es el flujo máximo de carros que pueden ir del nodo 1 al nodo 7 por hora?

Esta pregunta la responde el desarrollo presentado anteriormente. El flujo máximo de carros que pueden ir del nodo 1 al nodo 7 son 14 mil vehículos por hora.

- (c) ¿Cuál es la ruta que permite más cantidad de vehículos por hora?

Para responder esta pregunta, debemos analizar los caminos posibles del grafo final obtenido. Hay varias formas de llegar del nodo 1 al nodo 7, pero en particular, el camino $C = 1 \rightarrow 5 \rightarrow 6 \rightarrow 7$ es aquel que maximiza el número de vehículos pasantes por hora. Ahora, la cantidad de carros que pueden pasar por esta ruta directamente son $\min\{6, 6, 7\} = 6$, por lo que pueden pasar 6 mil carros por hora.

Se seleccionó este camino, porque cualquier otro camino implica pasar por lo menos una arista con capacidad menor a 6, afectando los miles de carros que pasan por hora. Este argumento es simple solución de un problema $\max\{\min\{\cdot\}\}$ que tiene como argumento, las aristas de los caminos del nodo 1 al 7.

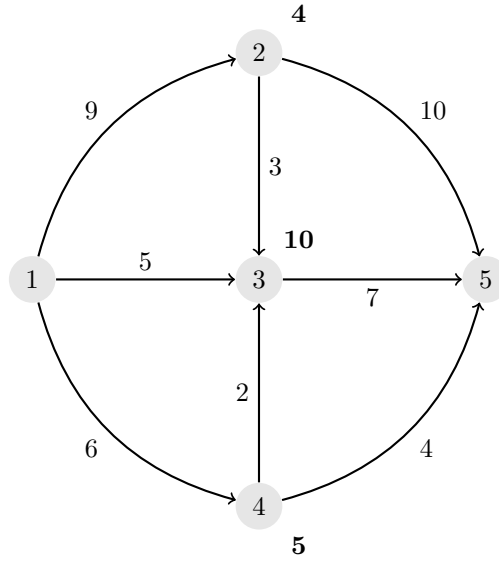
3. Problema de flujo máximo

Problema 3. Considere la siguiente red en la que además de capacidades máximas de flujo en cada arco, existen capacidades máximas de flujo que puede pasar por cada nodo. Se desea encontrar el flujo máximo que puede ser enviado desde el nodo 1 hasta el nodo 5 (ver figura 21). Modifique la red para poder encontrar el flujo máximo de 1 a 5 aplicando el algoritmo de Ford y Fulkerson. Resuelva y determine el flujo máximo (de forma analítica o mediante implementación). Describa cada etapa del algoritmo.

A continuación, se presenta una descripción analítica del problema en cuestión y de nuevo se resuelve mediante el algoritmo FF presentado en el problema anterior.

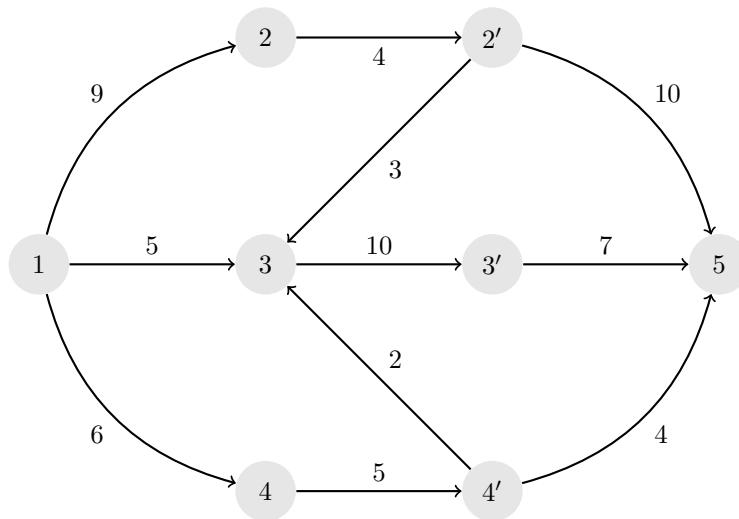
En general, una variante interesante del problema de flujo máximo es aquella en la que, además de tener una capacidad máxima $u(i, j) = u_{ij}$ para cada arco, también tenemos una capacidad máxima $u'(i)$ para uno o más vértices, y un flujo f_{ij} es factible solo si, además de satisfacer las restricciones de conservación y las restricciones de capacidad de arco, también cumple con las restricciones de capacidad de vértice:

$$\sum_{i:(i,j) \in N} f_{ij} \leq u'(i) \quad \text{para todo } j \in N$$

Figura 21: Grafo \mathcal{G} que describe el problema 3.

Es fácil ver que el problema se puede reducir al problema estándar de flujo máximo, dividiendo cada vértice con capacidad máxima i en dos vértices, i_{in} y i_{out} , agregando un arco (i_{in}, i_{out}) de capacidad $u'(i)$, y luego convirtiendo cada arco (k, i) en un arco (k, i_{in}) y cada arco (i, j) en un arco (i_{out}, j) . Es fácil demostrar que resolver el problema del flujo máximo (estándar, presentado en el problema 2) en la nueva red es equivalente a resolver el problema de flujo máximo con restricciones de capacidad de vértice en la red original.

Entonces, para resolver el problema 3, debemos considerar un grafo equivalente, donde los vértices no tengan asignados un flujo máximo. Para esto introducimos nuevos vértices artificiales (primados, de forma que $i = i_{in}$ y $i_{out} = i'$) como se expuso anteriormente:

Figura 22: Grafo $G = (N, A)$ equivalente al grafo inicial $\mathcal{G} = (\mathcal{N}, \mathcal{A})$.

Así, es posible aplicar el algoritmo FF. Claramente, la solución arrojada por el algoritmo para el flujo máximo en G , será también el flujo óptimo para la red \mathcal{G} .

Inicialmente consideremos la solución factible $f_{ij} = 0, \forall (i, j) \in A$, y por ende $F = 0$, dado que las cotas inferiores del grafo G son nulas. Luego, obtenemos el grafo inicial de flujo nulo presentado en la figura 23. Claramente el grafo auxiliar $G'^{(0)}$ es equivalente al grafo inicial G , por lo que el algoritmo continua buscando el camino $C'^{(0)}$, desde 1 hasta 5, cadena de $G'^{(0)}$.

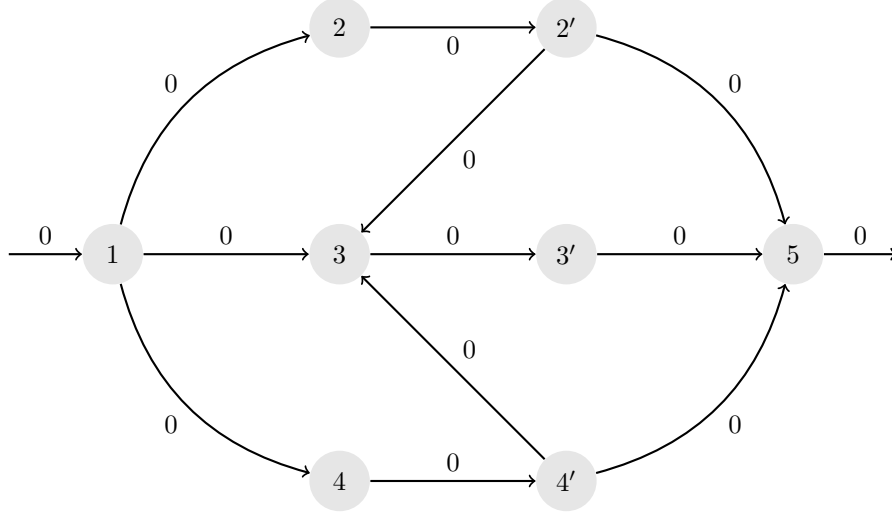


Figura 23: Grafo factible inicial $G^{(0)}$.

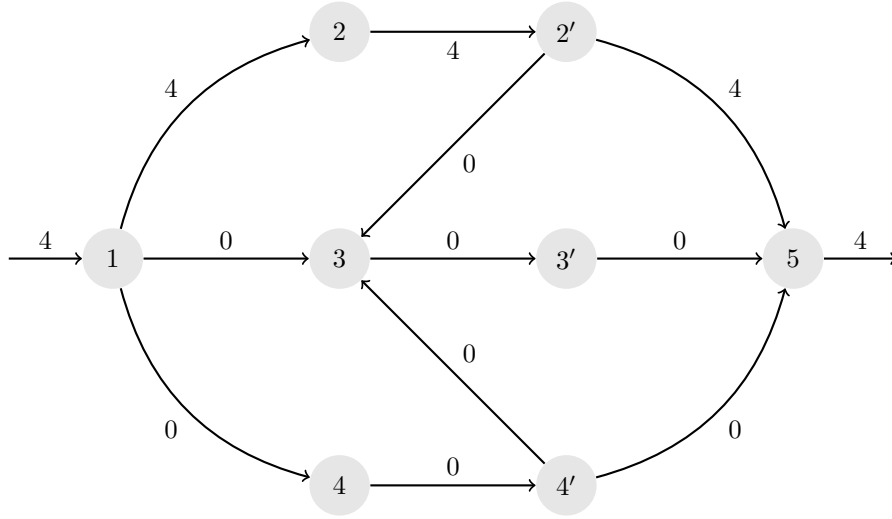
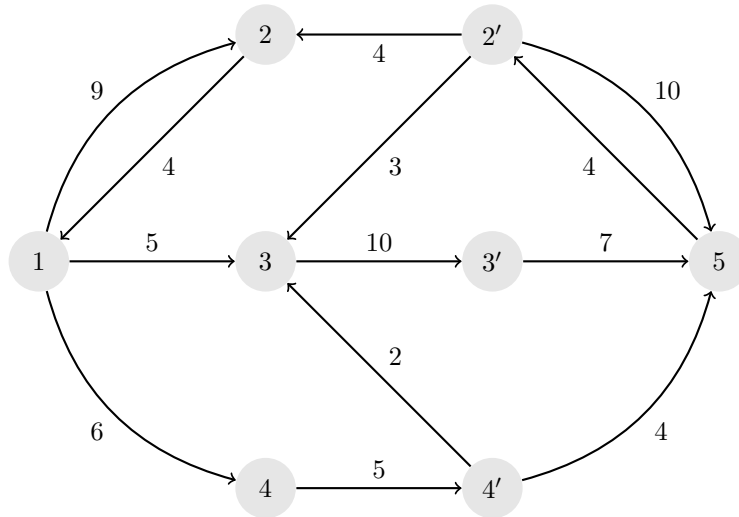
Si se considera el camino $C'^{(0)} = 1 \rightarrow 2 \rightarrow 2' \rightarrow 5$, siguiendo las instrucciones del algoritmo se obtendrá:

$$\begin{aligned} B_1^{(0)} &= A \\ B_2^{(0)} &= \emptyset \\ \min_{(i,j) \in C \cap B_1} (u_{ij} - f_{ij}) &= 4 \\ \varepsilon &= 4 \end{aligned}$$

Luego, haciendo $F^{(1)} = F^{(0)} + \varepsilon = 4$, y $f_{ij}^{(1)} = f_{ij}^{(0)} + \varepsilon$ para todo $(i, j) \in C^{(0)} \cap B_1 = C^{(0)}$, obtenemos el nuevo grafo factible $G^{(1)}$ enseñado en la figura 24.

Ahora, al igual que en el problema anterior, sigue iterar el algoritmo FF hasta que no exista camino C' en G' que conecte al nodo 1 con el nodo 5. Se narrará en cada iteración entonces el camino seleccionado, el incremento ε de cada iteración y se actualizarán el grafo factible $G^{(k)}$ y su asociado grafo auxiliar $G'^{(k)}$.

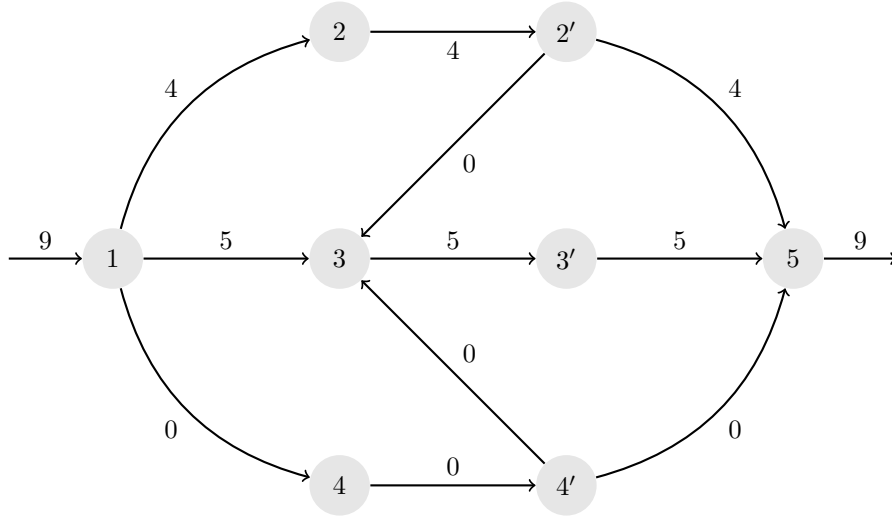
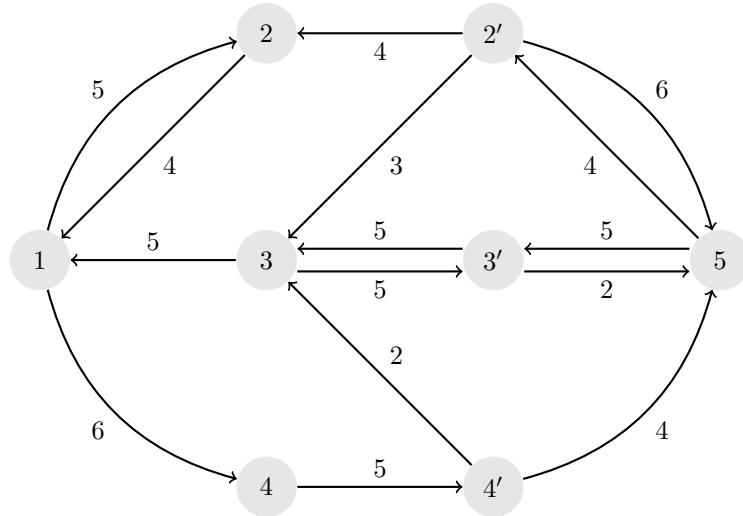
Para la iteración siguiente, se presenta el grafo auxiliar $G'^{(1)}$ en la figura 25.

Figura 24: Grafo factible luego de la primera iteración $G^{(1)}$.Figura 25: Grafo auxiliar del resultante en la primera iteración $G'^{(1)}$.

Consecuentemente, se contempla entonces el camino $C'^{(1)} = 1 \rightarrow 3 \rightarrow 3' \rightarrow 5$. Luego, $\varepsilon^{(1)} = \min\{5, 7, 10\} = 5$, y se actualiza el flujo exógeno máximo a $F^{(2)} = 9$, además de las aristas f_{ij} implicadas.

El siguiente grafo factible $G^{(2)}$ se enseña en la figura 26 con su respectivo flujo exógeno a la izquierda del nodo 1 y a la derecha del nodo 5.

Ahora es necesaria la construcción de otro grafo auxiliar $G'^{(2)}$ que tome en cuenta aquellas aristas que hayan superado sus capacidades máximas. El grafo auxiliar resultante, siguiendo los lineamientos del algoritmo FF se presenta en la figura 27.

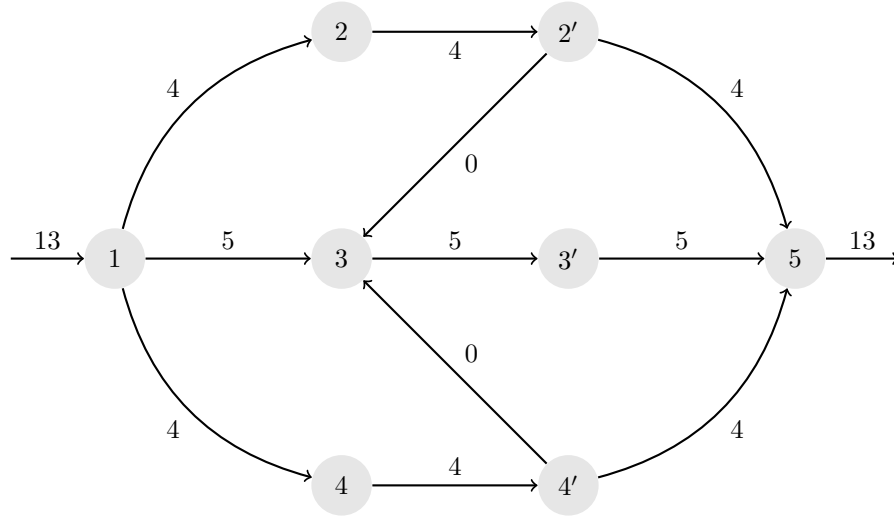
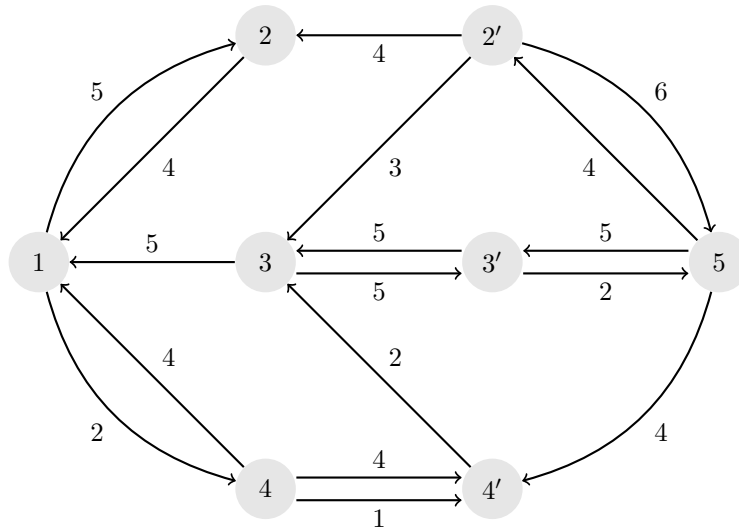
Figura 26: Grafo factible $G^{(2)}$.Figura 27: Grafo auxiliar $G''^{(2)}$.

En este punto se debe seleccionar una cadena en $G''^{(2)}$ que lleve del nodo 1 al 5. Para ello, consideramos $C''^{(2)} = 1 \rightarrow 4 \rightarrow 4' \rightarrow 5$.

Entonces, siguiendo la definición del incremento ε , se obtendrá $\varepsilon^{(2)} = \min\{6, 5, 4\} = 4$, obteniendo el nuevo flujo exógeno $F^{(3)} = 9 + 4 = 13$. Así, se actualiza entonces el grafo factible a $G^{(3)}$, que se puede observar en la figura 28.

De nuevo, es necesario crear el nuevo grafo auxiliar con las condiciones anteriormente dadas. Se puede notar que, los arcos se mantienen iguales menos los arcos del camino que se escogió. El grafo auxiliar correspondiente a $G^{(3)}$, se exhibe en la figura 29.

Y similarmente, en esta iteración debe buscarse un camino C' que lleve del nodo 1 al 5. Al observar el

Figura 28: Grafo factible $G^{(3)}$.Figura 29: Grafo auxiliar $G'^{(3)}$.

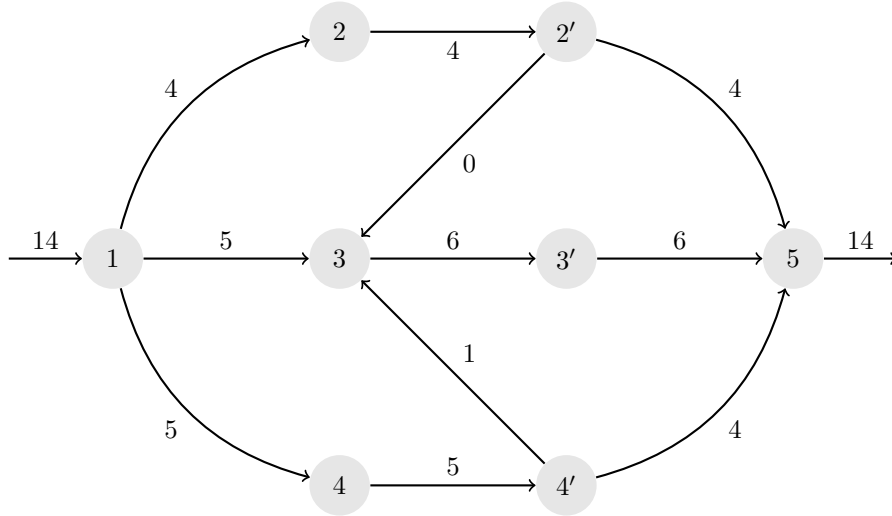
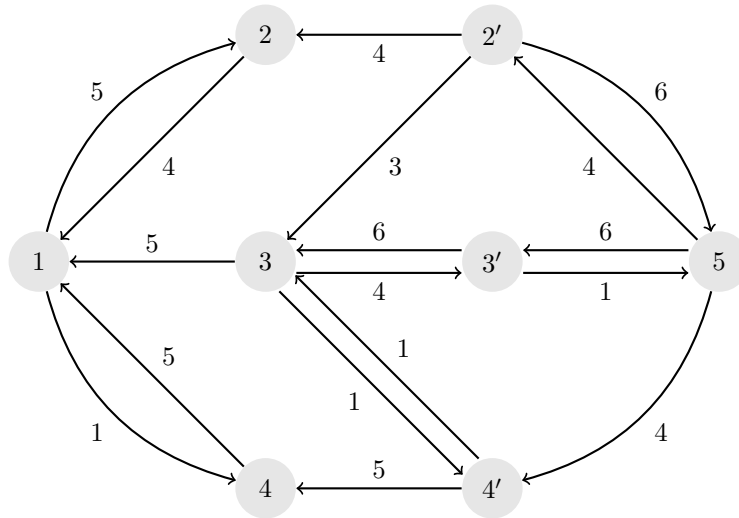
grafo auxiliar, se ve que un camino posible es $C^{(3)} = 1 \rightarrow 4 \rightarrow 4' \rightarrow 3 \rightarrow 3' \rightarrow 5$.

Entonces, el incremento ε queda definido como $\varepsilon^{(3)} = \min\{2, 1, 2, 5, 2\} = 1$, y por ende, el incremento en el flujo exógeno es esta cantidad, quedando $F^{(4)} = 14$.

El grafo factible actualizado $G^{(4)}$, (que veremos que es el último) se presenta en la figura 30.

Finalmente, ejecutando el algoritmo FF, debemos construir un grafo auxiliar que permita analizar la existencia de un camino C' del nodo inicial al final en él. El grafo auxiliar de $G^{(4)}$, $G'^{(4)}$ se observa en la figura 31 a continuación.

Observando detenidamente este grafo auxiliar, es posible identificar que no existe ningún camino desde

Figura 30: Grafo final (óptimo) $G^{(4)}$.Figura 31: Grafo auxiliar $G'^{(4)}$.

el nodo inicial 1 hasta el nodo final 5. Es decir, siguiendo la dirección de las flechas, es imposible llegar al nodo final.

En efecto, este es el criterio de parada del algoritmo FF, y lo que indica es que la solución actual no puede ser mejorada, es decir, no puede incrementarse ningún flujo f_{ij} sin violar las restricciones de flujo máximo y sin reducir el flujo exógeno anterior $F^{(4)} = 14$.

Entonces, el grafo anterior $G^{(4)}$ es óptimo, en cuanto a que este indica los flujos que deben pasar por cada arista para maximizar el flujo (los f_{ij}) y el flujo exógeno asociado, o valor máximo de la función objetivo $F^* = F^{(4)} = 14$.

Claramente, la consideración del grafo equivalente de \mathcal{G} , G , es una gran herramienta, para hallar el flujo máximo permitido por la red inicial expuesta en la figura 21, pero el grafo que expone este flujo

óptimo pierde interpretación en cuanto al flujo que le corresponde a cada arista, ¿qué significa un flujo de 4 unidades del nodo 3 al nodo 3'?

Para resolver esta carencia de interpretación, se puede intentar convertir el grafo final 30 a un grafo equivalente con flujos acumulados en los nodos 2, 3 y 4, eliminando los nodos primados. Un posible intento de llevar a cabo esto se expone en la figura 32, que corresponde a una “transformación inversa” de la propuesta inicialmente.

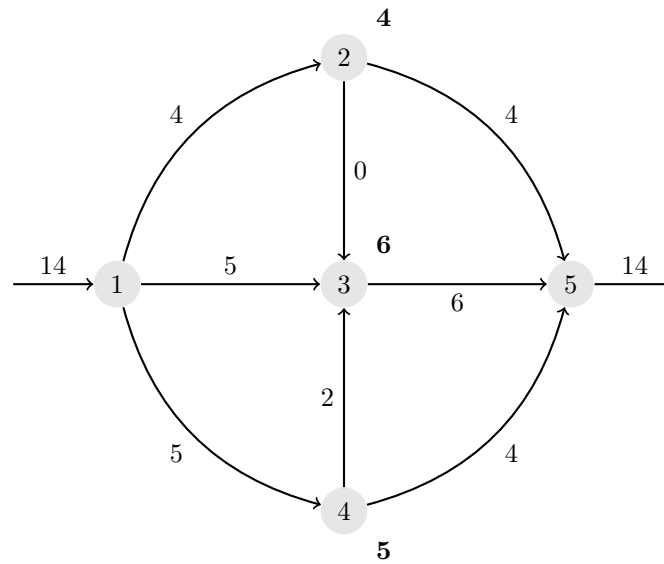


Figura 32: Caption

No obstante, no debe confundirse la conclusión a este problema, el flujo máximo permitido por la red \mathcal{G} es de 14 unidades por unidad de tiempo.

Referencias

- [1] T. Martinod, “optimizacion-1”, GitHub, 31-Oct-2023. [Online]. Available: <https://github.com/thomas-martinod/optimizacion-1>. [Accessed: 31-Oct-2023].
- [2] G. B. Dantzig, Linear Programming and Extensions. Princeton, NJ, USA: Princeton University Press, 1963.