



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

SOLUCIÓN A PROBLEMAS DE URBANISMO
UTILIZANDO TEORÍA DE GRÁFICAS

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

MATEMÁTICO

PRESENTA:

PABLO URIEL BENÍTEZ RAMÍREZ

TUTORA:

DRA. HORTENSIA GALEANA SÁNCHEZ

Ciudad de México, México, 2024





Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

*Por todos los que me acompañaron en éste viaje,
siempre los llevaré en mi corazón*

Agradecimientos

Inicio mis agradecimientos a mis papás, Susana y Uriel, por su infinito amor, su apoyo incondicional y por otorgarme la oportunidad de seguir mis sueños.

A mi asesora, Dra. Hortensia Galeana Sánchez, por creer en mí y apoyarme en la culminación de mi carrera.

A mis abuelas, Loreto y Catarina †, por siempre acompañarme en mi crecimiento y ayudarme a ser el hombre que soy.

A mi hermana Fátima, por ser siempre una luz en mi vida y un gran orgullo poder crecer junto a ella.

A mis amigos, Francisco, Erick y Obed, por su apoyo, amistad y excelente compañía desde los primeros años de la universidad.

A mis sinodales Dr. César Hernández Cruz, Dra. Isabel Alicia Hubard Escalera, Dra. María del Rocío Sánchez López y Dra. Martha Gabriela Araujo Pardo por su tiempo y entrega a la revisión de esta tesis.

Por último agradezco el apoyo recibido por parte del Consejo Nacional de Humanidades Ciencia y Tecnología, en particular al Sistema Nacional de Investigadores, por el apoyo con registro no. I1200/16/2020.

Introducción

En las últimas décadas, el crecimiento acelerado de las ciudades ha generado diversos problemas urbanos, tales como la congestión vehicular, la falta de espacios públicos y la desigualdad en la distribución de recursos, entre otros. Estos problemas tienen un impacto significativo en la calidad de vida de los habitantes y son una preocupación para los gobiernos locales y las comunidades.

La planificación urbana es un campo interdisciplinario que aborda los desafíos complejos que enfrentan las ciudades en el siglo XXI. Desde la congestión del tráfico hasta la falta de espacios verdes, los urbanistas se enfrentan a un conjunto cada vez más diverso de problemas urbanos que requieren soluciones innovadoras.

El urbanismo nace en la Revolución industrial [13], sin embargo tras la Segunda Guerra Mundial [2], la creciente urbanización en todo el mundo generó una necesidad cada vez mayor de planificación y gestión eficaz de las ciudades, las cuales se enfrentaban a problemas como la congestión del tráfico, la contaminación, la falta de espacios verdes y la desigualdad en la distribución de servicios urbanos. En los años 70 [12] [9] [10], varios académicos comenzaron a aplicar la teoría de gráficas en el análisis de problemas urbanos específicos.

Fue entonces que matemáticos, científicos sociales y urbanistas comenzaron a buscar nuevas herramientas para modelar y analizar los problemas urbanos. La teoría de gráficas se destacó como una herramienta especialmente valiosa debido a su capacidad para representar objetos y relaciones entre ellos de manera visual y matemática.

Actualmente, en México, la construcción desmedida de unidades habitacionales y zonas residenciales dentro y fuera de las principales ciudades del país, ha provocado a lo largo de los años graves problemas, principalmente ecológicos, tales como el desbordamiento de ríos, la pérdida de zonas agrícolas y forestales, emisiones desproporcionadas de gases de efecto invernadero, y escasez del agua. Además ese crecimiento desmedido

va de la mano con la creación de barrios pobres, con personas en condiciones de vida insalubres, en zonas de riesgo, hacinadas, sin seguridad de tenencia de sus viviendas y de la tierra, así como con mayores disparidades, desigualdades y discriminación [15].

En este contexto, la teoría de gráficas es una herramienta útil para el modelado y análisis de situaciones complejas, utilizando una serie de conceptos y técnicas para la modelación de problemas en forma de gráficas, donde los vértices representan los elementos del problema y las aristas representan las relaciones entre ellos.

La aplicación de la teoría de gráficas en el urbanismo se ha expandido y ha evolucionado para abordar una amplia gama de problemas urbanos, incluyendo la planificación de rutas de bicicletas, la optimización de redes de energía y agua y la identificación de zonas de alta densidad de tráfico.

El objetivo principal de esta tesis es mostrar cómo la teoría de gráficas puede ser utilizada para solucionar problemas de urbanismo. A través de casos de estudio concretos, se mostrará cómo esta herramienta puede ser aplicada para la optimización de rutas de transporte público, la planificación de redes de parques y espacios verdes, la identificación de zonas de alta densidad de tráfico y la distribución equitativa de recursos urbanos.

El presente trabajo se divide en 4 capítulos que abarcan desde los conceptos básicos de la teoría de gráficas hasta su aplicación en problemas específicos de urbanismo. Abordando temas y problemáticas actuales, se espera que esta tesis sea de utilidad para urbanistas, matemáticos y profesionales interesados en el desarrollo de ciudades más sostenibles, accesibles y equitativas.

Índice general

Agradecimientos	II
Introducción	III
1 Definiciones y resultados previos	1
1.1 Gráficas y gráficas simples	1
1.2 Caminos, árboles y conexidad	5
1.3 Gráficas dirigidas	8
1.4 Algoritmos	15
1.4.1 Algoritmo de Dijkstra	15
1.4.2 Algoritmo de Floyd-Warshall	22
2 Ruta de autobuses	35
2.1 Conceptos	35
2.2 Planteamiento del problema	37
2.3 Modelación matemática del problema	37
3 Ubicación de servicios de emergencia	43
3.1 Conceptos	43
3.2 Planteamiento del problema	46
3.3 Modelación matemática del problema	46
3.4 Planteamiento del problema	57
3.5 Modelación matemática del problema	57
3.6 Desarrollo del código (Python)	73
3.6.1 Construcción de gráficas, digráficas y multidigráficas con pesos .	73
3.6.2 Puntos centrales de la digráfica	79
3.6.3 Arborescencia: digráfica de caminos mínimos	83

<i>ÍNDICE GENERAL</i>	VI
4 Mejor ruta dada las condiciones del terreno	101
4.1 Planteamiento del problema	101
4.2 Modelación matemática del problema	101
Conclusiones	117
Figuras y mapas	118

Capítulo 1

Definiciones y resultados previos

1.1. Gráficas y gráficas simples

Esta sección está basada en la referencia [7].

Definición 1.1.1. El conjunto $\binom{V_G}{2}$ tal que sus elementos son de la forma $\{v_i, v_j\}$ con v_i y v_j vértices distintos de V_G (para simplificar la notación será escrito como $v_i v_j$). El conjunto $\binom{V_G}{1}$ tal que sus elementos son de la forma $\{v_i\}$ con v_i vértice de V_G (para simplificar la notación será escrito como $v_i v_i$).

Una *gráfica* G es una terna $G = (V_G, A_G, \psi_G)$, en la cual $V_G \neq \emptyset$ es el conjunto de vértices, A_G ajeno a V_G es el conjunto de aristas y $\psi_G : A_G \rightarrow \binom{V_G}{2} \cup \binom{V_G}{1}$ es la función de incidencia.

Definición 1.1.2. El *orden* de G es $n_G = |V_G|$ (cantidad de vértices) y el *tamaño* de G es $m_G = |A_G|$ (cantidad de aristas).

Ejemplo 1.1.1. Consideremos a la gráfica G tal que $A_G = \{a_1, a_2, a_3, a_4\}$ y $V_G = \{v_1, v_2, v_3, v_4\}$, la regla de correspondencia para ψ es $\psi(a_1) = v_1 v_2$, $\psi(a_2) = v_1 v_3$, $\psi(a_3) = v_2 v_4$, $\psi(a_4) = v_1 v_4$.

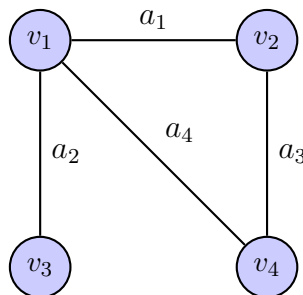


FIGURA 1.1: Gráfica G

El orden de G es $n_G = |V_G| = 4$ y el tamaño es $m_G = |A_G| = 4$.

Definición 1.1.3. Sean $a, b \in A_G$ y $v_i \in V_G$, si $\psi(a) = v_i v_i$ entonces decimos que a es un *lazo* (en v_i). Si $\psi(a) = \psi(b)$ entonces decimos que a y b son *multiaristas*.

Ejemplo 1.1.2. Consideremos una ligera modificación de la gráfica del Ejemplo 1.1.1, sea G la gráfica tal que $A_G = \{a_1, a_2, a_3, a_4, a_5, a_6\}$ y $V_G = \{v_1, v_2, v_3, v_4\}$, la regla de correspondencia es $\psi(a_1) = v_1 v_2$, $\psi(a_2) = v_1 v_3$, $\psi(a_3) = v_2 v_4$, $\psi(a_4) = v_1 v_4$, $\psi(a_5) = v_1 v_1$, $\psi(a_6) = v_1 v_2$.

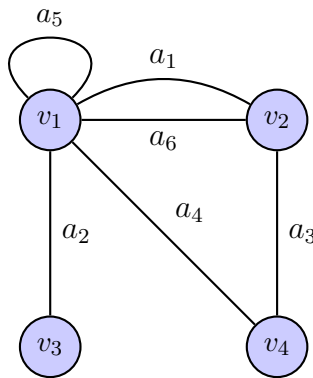


FIGURA 1.2: Gráfica G

La gráfica G tiene un lazo en el vértice v_1 y dos multiaristas entre los vértices v_1 y v_2 ; el orden de G es $n_G = |V_G| = 4$ y el tamaño es $m_G = |A_G| = 6$.

Definición 1.1.4. Una gráfica G es *simple* si no tiene lazos ni multiaristas.

Si G es simple, entonces podemos escribir a G como (V, A) donde $A \subseteq \binom{V}{2}$, el cual es un conjunto tal que sus elementos son de la forma $\{v_i, v_j\}$ con v_i y v_j vértices de V . La función de incidencia es inyectiva (no hay multiaristas) y cada arista se corresponde con un subconjunto de exactamente dos vértices bajo la función incidencia (no hay lazos). Un ejemplo de una gráfica simple es el Ejemplo 1.1.1.

Definición 1.1.5. Sea $G = (V_G, A_G, \psi_G)$ una gráfica, supongamos que enumeramos a los vértices de G como $V_G = \{v_1, \dots, v_n\}$, la *matriz de adyacencia* de G , $\mathbf{M}_G = (a_{ij})$, es la matriz de $n \times n$ entradas a_{ij} , donde a_{ij} es la cantidad de aristas que unen a v_i con v_j , es decir, $a_{ij} = |\{a \in A_G : \psi(a) = v_i v_j\}|$.

Ejemplo 1.1.3. A continuación se muestran las matrices de adyacencia de los ejemplos

1.1.1 y 1.1.2 respectivamente:

$$\begin{array}{c}
 v_1 \quad v_2 \quad v_3 \quad v_4 \\
 v_1 \quad \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \\
 v_2 \\
 v_3 \\
 v_4
 \end{array}$$

Matriz del ejemplo 1.1.1

$$\begin{array}{c}
 v_1 \quad v_2 \quad v_3 \quad v_4 \\
 v_1 \quad \begin{pmatrix} 1 & 2 & 1 & 1 \\ 2 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \\
 v_2 \\
 v_3 \\
 v_4
 \end{array}$$

Matriz del ejemplo 1.1.2

Definición 1.1.6. Sean $G = (V_G, A_G, \psi_G)$ y $H = (V_H, A_H, \psi_H)$ gráficas, decimos que H es *subgráfica* de G ($H \leq G$) si y sólo si $V_H \subseteq V_G$, $A_H \subseteq A_G$, y ψ_H es la restricción de ψ_G a A_H , es decir, $\psi_H(a) = \psi_G(a)$ para toda $a \in A_H$, en símbolos escribimos $\psi_H = \psi_G|_{A_H}$.

Definición 1.1.7. Sea $H \leq G$, si $V_H = V_G$, entonces decimos que H es *subgráfica generadora*.

Ejemplo 1.1.4. Sean G una gráfica y H una subgráfica generadora de G tal que $V_G = V_H = \{v_1, v_2, v_3, v_4, v_5, v_6\}$, $|A_G| = 11$ y $|A_H| = 5$,

$$\begin{array}{lll}
 \psi_G(a_1) = \psi_H(b_1) = v_1v_2 & \psi_G(a_2) = v_1v_3 & \psi_G(a_3) = v_1v_4 \\
 \psi_G(a_4) = v_1v_5 & \psi_G(a_5) = \psi_H(b_2) = v_1v_6 & \psi_G(a_6) = v_2v_3 \\
 \psi_G(a_7) = v_2v_6 & \psi_G(a_8) = \psi_H(b_3) = v_3v_4 & \psi_G(a_9) = \psi_H(b_4) = v_3v_5 \\
 \psi_G(a_{10}) = v_3v_6 & \psi_G(a_{11}) = \psi_H(b_5) = v_5v_6 &
 \end{array}$$

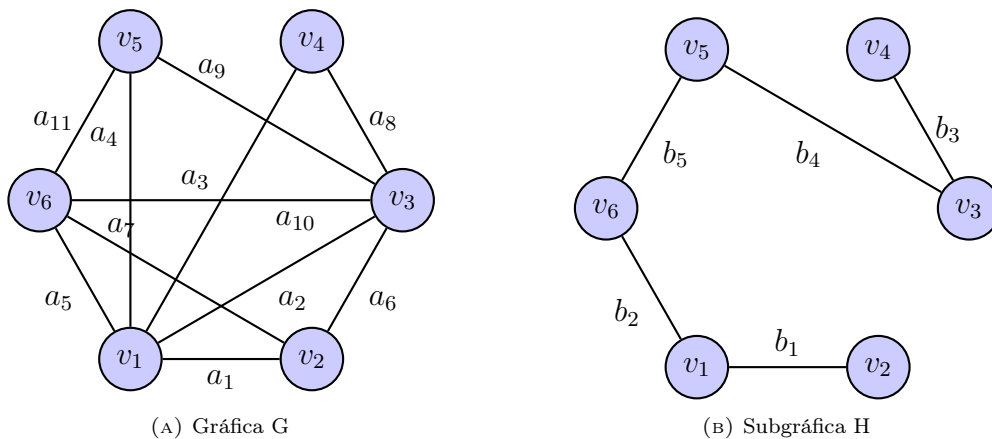


FIGURA 1.3

Definición 1.1.8. Sea $G = (V_G, A_G, \psi_G)$ una gráfica y $V' \subseteq V_G$ un subconjunto no vacío, definimos al *conjunto de aristas inducido* por V' , como $A[V'] = \{a \in A_G : \psi_G(a) \in \binom{V'}{2}\}$

y a la *subgráfica inducida* por V' , como $G[V'] = (V', A[V'], \psi_G|_{A[V']})$.

Definición 1.1.9. Sea $G = (V_G, A_G, \psi_G)$ una gráfica y $A' \subseteq A_G$, definimos al *conjunto de vértices inducido* por A' , como $V[A'] = \{v \in V_G : vw = \psi_G(a) \text{ para alguna } a \in A' \text{ y para algún } w \in V_G - \{v\}\}$ y a la *subgráfica inducida* por A' , como $G[A'] = (V[A'], A', \psi_G|_{A'})$.

Definición 1.1.10. La gráfica (simple) completa de orden n es la gráfica simple $K_n = (V_{K_n}, A_{K_n})$ con $n = |V_{K_n}|$ y $A_{K_n} = \{uv : u, v \in V_{K_n}\}$ tal que $u \neq v$. Esta gráfica tiene a todas las aristas posibles para una gráfica simple de orden n .

Ejemplo 1.1.5. Ejemplos de gráficas K_2 , K_3 y K_4 .

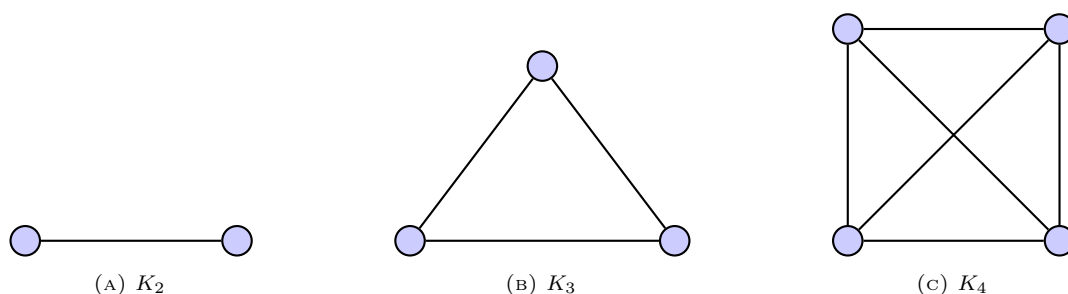


FIGURA 1.4: Ejemplos de gráficas completas

Definición 1.1.11. El *grado* de un vértice v en una gráfica G , se define como $deg(v) = |\{a \in A_G : \psi_G(a) = vw \text{ p.a. } w \in V_G - \{v\} \text{ y } a \text{ no es lazo}\}| + 2 |\{a \in A_G : a \text{ es un lazo de } v\}|$, es decir, $deg(v)$ es la cantidad de aristas de G en las que incide v contando a los lazos doblemente.

Ejemplo 1.1.6. En la gráfica G del Ejemplo 1.1.4 los grados de los vértices son:

$$deg(v_1) = 5 \quad deg(v_2) = 3 \quad deg(v_3) = 5 \quad deg(v_4) = 2 \quad deg(v_5) = 3 \quad deg(v_6) = 4$$

Definición 1.1.12. Dada una gráfica, definimos a $\delta_G = \min\{g_G(v) : v \in V_G\}$ y a $\Delta_G = \max\{g_G(v) : v \in V_G\}$.

Ejemplo 1.1.7. En la gráfica G del Ejemplo 1.1.4.

$$\delta_G = 2$$

$$\Delta_G = 5$$

1.2. Caminos, árboles y conexidad

Esta sección está basada en la referencia [7].

Definición 1.2.1. En una gráfica G , un *camino de longitud k* es una sucesión

$$C = (v_0, a_0, v_1, a_1, \dots, v_{k-1}, a_{k-1}, v_k)$$

en la que se alternan vértices y aristas de G , tal que $\psi(a_i) = v_{i-1}v_i$ para toda i con $1 \leq i \leq k$. En este caso decimos que C es un camino entre v_0 y v_k o un v_0v_k -camino de longitud k .

Ejemplo 1.2.1. Podemos considerar a la gráfica H del Ejemplo 1.1.4 como el camino C_1

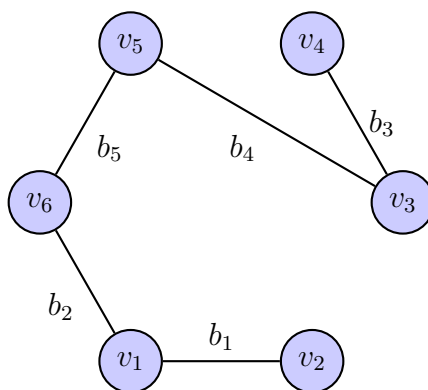


FIGURA 1.5: gráfica H

$$C_1 = (v_2, b_1, v_1, b_2, v_6, b_5, v_5, b_4, v_3, b_3, v_4)$$

Definición 1.2.2. Un *paseo* o *recorrido* es un camino que no repite aristas. Una *trayectoria* es un camino que no repite vértices (y por ende tampoco repite aristas).

Definición 1.2.3. Sean $v, u \in V_G$; decimos que u y v están *conectados* si existe un uv -camino en G .

Proposición 1.2.4. Sea \sim_c la relación en V_G tal que $u \sim_c v$ si y sólo si u y v están conectados, entonces \sim_c es de equivalencia.

Demostración. Para todo $v \in V_G$, el camino trivial $C = (v)$ es un vv -camino (de longitud cero), por lo que $v \sim_c v$. Si $u \sim_c v$, existe un uv -camino C , entonces u y v

están conectados, por lo que también existe un vu -camino C' , entonces $v \sim_c u$. Si $v \sim_c u$ y $u \sim_c w$, entonces existe un vu -camino C y un uw -camino C' , entonces u , v y w están conectados, entonces también existe un vw -camino, lo que implica que $v \sim_c w$. \square

Definición 1.2.5. Sea $\{V_1, V_2, \dots, V_\omega\}$ la partición de V_G inducida por \sim_c , es decir, u y v están en el mismo conjunto V_i si y sólo si $u \sim_c v$. Las subgráficas $G[V_1], \dots, G[V_\omega]$ son las *componentes conexas* de la gráfica G . Al número de componentes conexas de G lo denotamos por ω_G u $\omega(G)$.

Decimos que la gráfica G es conexa cuando sólo tiene una componente conexa ($\omega(G) = 1$), en otro caso decimos que es inconexa ($\omega(G) \geq 2$).

Definición 1.2.6. La *distancia* entre dos vértices $u, v \in V_G$ se define como

$$d_G(u, v) = \begin{cases} \text{mín}\{k \in \mathbb{N}: \text{existe un } uv\text{-camino de longitud } k\} & \text{si } G \text{ es conexa} \\ \infty & \text{si } G \text{ es inconexa (no hay } uv\text{-caminos en } G) \end{cases}$$

Un uv -camino de longitud $d_G(u, v)$ es llamado una geodésica, o uv -geodésica o un camino mínimo de G .

Definición 1.2.7. Un *camino cerrado* es un uv -camino en el que $u = v$. Un camino cerrado de longitud al menos 3, en el que la única repetición de vértices es la que ocurre en los extremos, es un *ciclo*. Si la longitud de dicho camino es k , lo llamamos un k -ciclo.

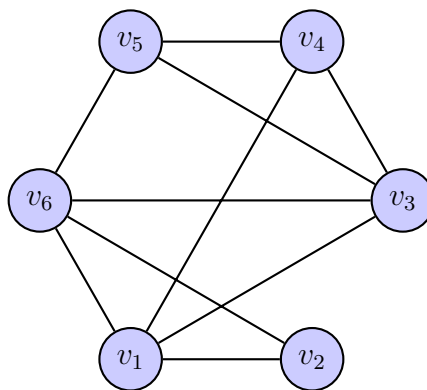


FIGURA 1.6

Ejemplo 1.2.2. El camino $C_1 = (v_1, v_3, v_4, v_5, v_6, v_3, v_1)$ de la Figura 1.6 es cerrado y tiene longitud 6, sin embargo no es un ciclo. El camino $C_2 = (v_1, v_6, v_2, v_1)$ es cerrado y es un 3-ciclo.

Definición 1.2.8. Una gráfica G es un árbol si es conexa y no tiene ciclos.

Ejemplo 1.2.3.



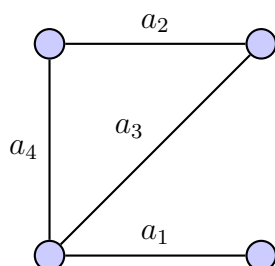
FIGURA 1.7

Teorema 1.2.9. Si G es un árbol, entonces $m_G = n_G - 1$.

Definición 1.2.10. Sea $G = (V_G, A_G, \phi_G)$ una gráfica, definimos a $G - a$ como la gráfica tal que para alguna $a \in A_G$ se tiene que $a \notin G - a$.

Decimos que $a \in A_G$ es *arista de corte* de G si $\omega(G - a) > \omega(G)$.

Ejemplo 1.2.4.



a_1 es *arista de corte* ya que $\omega(G - a_1) = 2 > 1 = \omega(G)$
 pero a_2 no es *arista de corte* puesto que $\omega(G - a_2) = 1 \not> 1 = \omega(G)$

FIGURA 1.8

Definición 1.2.11. Sea G una gráfica conexa, una subgráfica $H \leq G$ es llamada un árbol generador, si es una subgráfica generadora y además es un árbol.

Proposición 1.2.12. Toda gráfica conexa G tiene un árbol generador.

Demostración. Sean G una gráfica conexa y $H \leq G$ una subgráfica generadora de G tal que toda arista $a \in V_H$ sea de corte, entonces $H - a$ es inconexa para toda $a \in A(H)$, además H no tiene ciclos, ya que de tenerlo se tendría una arista $b \in A(H)$ tal que $H - b$ es conexa, lo que contradice la construcción de H , entonces como es subgráfica

generadora de una gráfica conexa, también es conexa y por ende es un árbol (Definición 1.2.8). Por lo tanto es un árbol generador. \square

1.3. Gráficas dirigidas

Esta sección está basada en las referencias [4] y [3].

Definición 1.3.1. Una *gráfica dirigida* (o simplemente *digráfica*) G es una terna $G = (V_G, F_G, \phi_G)$, en la cual $V_G \neq \emptyset$ es el conjunto de vértices, F_G ajeno a V_G es el conjunto de flechas y la *función de incidencia* ϕ_G que asocia a cada flecha de G un par ordenado de vértices de G (no necesariamente distintos).

Si a es una flecha y u y v son vértices tales que $\phi_G(f) = (u, v)$, entonces decimos que a une a u con v , o bien, u es la *cola* de a y v es la *cabeza*. También se dice que la flecha (u, v) *inicia* en u y *termina* en v .

Observación 1.3.2. Cuando se habla de una sola digráfica, para fines prácticos, podemos expresarla como $G = (V, F, \phi)$.

Ejemplo 1.3.1. Consideremos a la digráfica G tal que $F_G = \{f_1, f_2, f_3, f_4\}$ y $V_G = \{v_1, v_2, v_3, v_4\}$, la regla de correspondencia para ϕ es $\phi(f_1) = (v_2, v_1)$, $\phi(f_2) = (v_3, v_1)$, $\phi(f_3) = (v_2, v_4)$, $\phi(f_4) = (v_1, v_4)$.

Otro ejemplo es la digráfica H tal que $F_H = \{g_1, g_2, g_3, g_4, g_5\}$ y $V_H = \{u_1, u_2, u_3, u_4\}$, la regla de correspondencia para ϕ es $\phi(g_1) = (u_3, u_4)$, $\phi(g_2) = (u_1, u_2)$, $\phi(g_3) = (u_4, u_2)$, $\phi(g_4) = (u_1, u_3)$, $\phi(g_5) = (u_2, u_3)$.

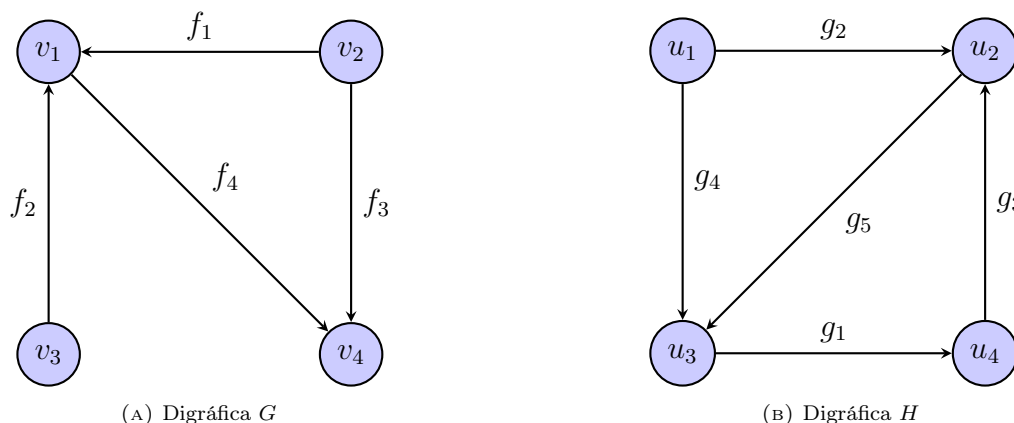


FIGURA 1.9

Definición 1.3.3. Sea $G = (V, F, \phi)$ una digráfica, decimos que una flecha *incide desde* $v \in V$ si ésta inicia en v . Del mismo modo decimos que una flecha *incide hacia* $v \in V$ si ésta termina en v .

Definición 1.3.4. Sea $G = (V, F, \phi)$ una digráfica, definimos lo siguiente:

$$N_G^+(v) = \{v \in V - u: vu \in F\}, \quad N_G^-(v) = \{w \in V - u: wv \in F\}$$

Tal que los conjuntos $N_G^+(v)$, $N_G^-(v)$ y $N_G(v) = N_G^+(v) \cup N_G^-(v)$, se les conoce como *ex-vecindario*, *in-vecindario* y *vecindario* de v .

Definición 1.3.5. Sea $G = (V, F, \phi)$ una digráfica, definimos lo siguiente

- a) El *in-grado* ($d_G^-(v)$) de un vértice es el número de flechas que inciden hacia dicho vértice.
- b) El *ex-grado* ($d_G^+(v)$) de un vértice es el número de flechas que inciden desde dicho vértice.

Definición 1.3.6. Sea $G = (V, F)$ una digráfica, el *grado* ($d_G(v)$) de un vértice es la suma del in-grado y el ex-grado, es decir, el número de flechas con un extremo en v

$$d_G(v) = d_G^-(v) + d_G^+(v)$$

Ejemplo 1.3.2. Sea G la digráfica siguiente:

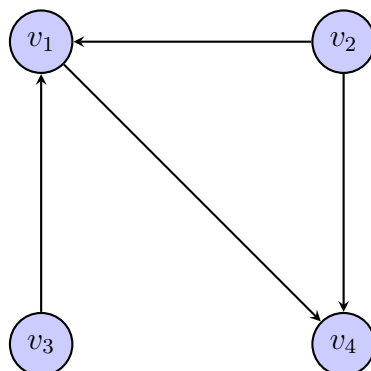


FIGURA 1.10

Los in-grados, ex-grados y grados de cada vértice son los siguientes:

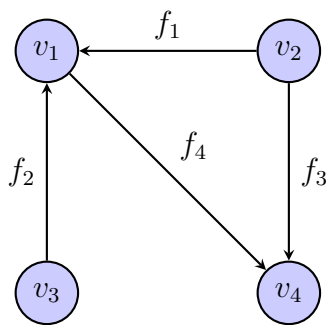
$$\begin{array}{lll}
 d_G^-(v_1) = 2 & d_G^+(v_1) = 1 & d_G(v_1) = 3 \\
 d_G^-(v_2) = 0 & d_G^+(v_2) = 2 & d_G(v_2) = 2 \\
 d_G^-(v_3) = 0 & d_G^+(v_3) = 1 & d_G(v_3) = 1 \\
 d_G^-(v_4) = 2 & d_G^+(v_4) = 0 & d_G(v_4) = 2
 \end{array}$$

Definición 1.3.7. Sean $G = (V, F, \phi)$ una digráfica, $\{f, g\} \subseteq F$ y $v_i \in V$. Si $\phi(f) = (v_i, v_i)$, decimos que f es un *lazo*. Si $\phi(f) = \phi(g)$, con $f \neq g$, decimos que f y g son *multiflechas*.

Definición 1.3.8. Una digráfica G que permite multiflechas pero no lazos, se le conoce como *multidigráfica*. Y si no permite multiflechas ni lazos, se le conoce como *digráfica simple*.

Definición 1.3.9. Sea $G = (V, F, \phi)$ una digráfica, supongamos que enumeramos a los vértices de G como $V = \{v_1, v_2, \dots, v_n\}$. La *matriz de adyacencia* de G , es la matriz de $n \times n$ entradas a_{ij} , donde a_{ij} es la cantidad de flechas que inician en v_i y terminan en v_j , con $v_i, v_j \in V$.

Ejemplo 1.3.3. A continuación veremos las matrices de adyacencia de cada una de las digráficas del ejemplo 1.3.1.

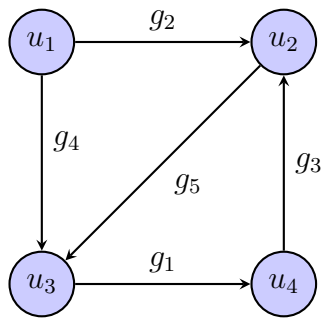


(A) Digráfica G

$$\begin{array}{l}
 v_1 \quad v_2 \quad v_3 \quad v_4 \\
 v_1 \quad \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\
 v_2 \\
 v_3 \\
 v_4
 \end{array}$$

(B) Matriz de la digráfica G

FIGURA 1.11



(A) Digráfica H

	u_1	u_2	u_3	u_4
u_1	0	1	1	0
u_2	0	0	1	0
u_3	0	0	0	1
u_4	0	1	0	0

(B) Matriz de la digráfica H

FIGURA 1.12

Definición 1.3.10. En una digráfica $G = (V, F, \phi)$ un *camino dirigido de longitud k* es una sucesión

$$C = (v_1, f_1, v_2, f_2, \dots, v_{k-1}, f_{k-1}, v_k, f_k, v_{k+1})$$

en la que se alternan vértices v_i y flechas f_j de G , de tal manera que la flecha f_j inicia en v_j y termina en v_{j+1} , para cada $j \in \{1, 2, \dots, k\}$.

Un camino dirigido que inicia en u y termina en v , con $u, v \in V$, es llamado un *uv -camino dirigido*.

Ejemplo 1.3.4. Sea G la siguiente digráfica:

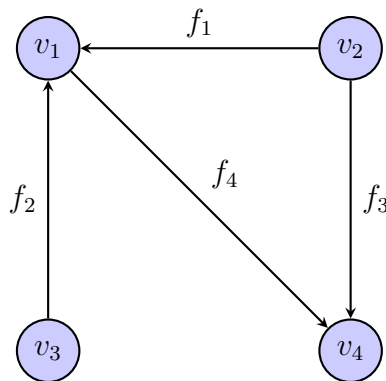


FIGURA 1.13

Algunos caminos dirigidos que se tienen son los siguientes:

$$C_1 = (v_3, f_2, v_1, f_4, v_4) \quad C_2 = (v_2, f_1, v_1) \quad C_3 = (v_2, f_3, v_4)$$

Definición 1.3.11. Una digráfica G es un *árbol dirigido* es una orientación de un árbol (Definición 1.2.8).

Definición 1.3.12. Una *arborescencia* $G = (V, F, \phi)$ es un árbol dirigido, es decir, una digráfica conexa que no posee algún uv -camino dirigido tal que $u = v$, en el cual todos los vértices tendrán in-grado igual a la unidad, a excepción de uno, la *raíz* de la arborescencia, la cual es un vértice $s \in V$ tal que para toda $x \in V$ y $x \neq s$ se tiene que $d_G^-(x) = 1$ y $d_G^-(s) = 0$.

Ejemplo 1.3.5. Sean $G = (V_G, F_G, \psi_G)$ y $H = (V_H, F_H, \psi_H)$ las digráficas siguientes:

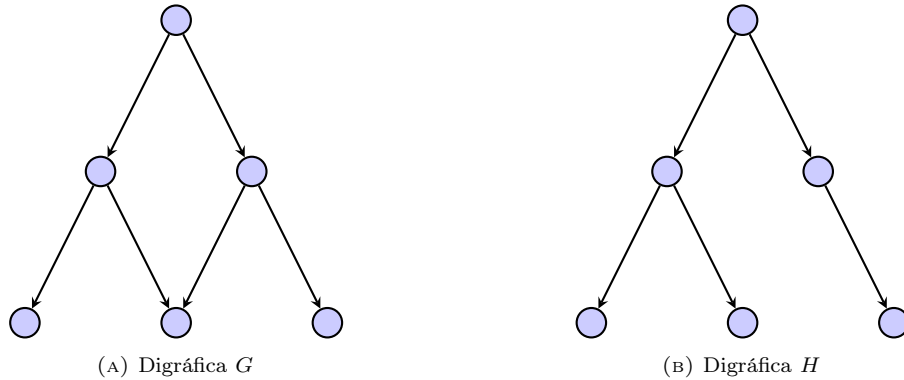


FIGURA 1.14

La digráfica H es una arborescencia, sin embargo G no lo es, ya que tiene un vértice con in-grado dos.

Definición 1.3.13. Una *digráfica ponderada* (o *digráfica con pesos*) es una terna $G = (V, F, W)$, donde $V = \{v_1, v_2, \dots, v_k\}$ es el conjunto de vértices, $F = \{f_1, f_2, \dots, f_n\}$ es el conjunto de flechas, ajeno a V , y $W = \{w_1, w_2, \dots, w_n\}$ con $w_i \in \mathbb{R}$, es el conjunto de pesos asociados a cada flecha. W se puede representar como una función de asignación de pesos $w : F \rightarrow \mathbb{R}$, de modo que para cualquier flecha $f \in F$ su peso es $w(f) \in \mathbb{R}$.

Observación 1.3.14. Se puede expresar a la digráfica con pesos $G = (V, F, W)$ como $G = (V, F, w)$, con w la función de asignación de pesos.

Observación 1.3.15. También podemos hablar de gráficas simples con pesos en las aristas, a éstas se les puede expresar como la terna $G = (V, A, w)$, con V el conjunto de los vértices, A el conjunto de aristas y w la función de asignación de pesos a cada una de las aristas de la gráfica.

Ejemplo 1.3.6. Sea $G = (V, F, \phi)$ una digráfica tal que $F = \{f_1, f_2, f_3, f_4, f_5\}$ y $V = \{u_1, u_2, u_3, u_4\}$, y $f_1 = (u_3, u_4)$, $f_2 = (u_1, u_2)$, $f_3 = (u_4, u_2)$, $f_4 = (u_1, u_3)$, $f_5 = (u_2, u_3)$.

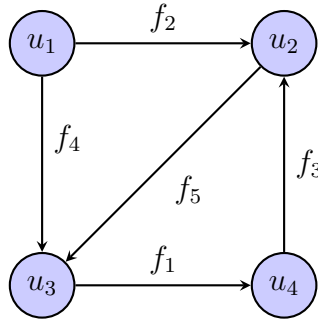


FIGURA 1.15

Ahora asignamos los siguientes pesos a cada flecha: $w(f_1) = 3$, $w(f_2) = 2$, $w(f_3) = 5$, $w(f_4) = 4$ y $w(f_5) = 1$. Obteniendo a la digráfica con pesos $H = (V, F, W)$.

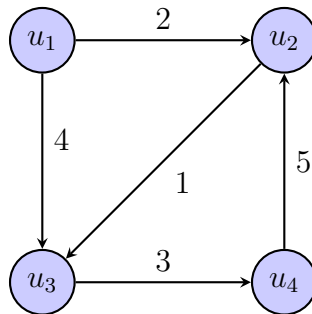


FIGURA 1.16

Observación 1.3.16. Siempre que se hable de longitud de un camino de una digráfica ponderada, nos referimos a la suma de los pesos asociados a las flechas que conforman ese camino.

Definición 1.3.17. Sea $G = (V, F, W)$ una digráfica con pesos, que no posee pesos negativos, de lo contrario esta definición carece de sentido. Definimos la *distancia* de u hacia v para $u, v \in V$, denotada por $d(u, v)$, como lo siguiente :

- $d(u, v) = k$, donde k es la longitud de un camino dirigido de peso mínimo de u a v , cuando existe al menos un uv -camino dirigido.
- Cuando no existe un uv -camino dirigido se tiene que $d(u, v) = \infty$.

Por lo tanto, la función cumple con la desigualdad del triángulo,

- $d(x, z) \leq d(x, y) + d(y, z)$ para toda triplete de vértices x, y, z .

Observación 1.3.18. Nótese que no siempre $d(u, v) = d(v, u)$.

Ejemplo 1.3.7. Sea G la digráfica siguiente:

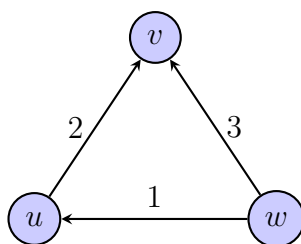


FIGURA 1.17

La distancia entre cada par de vértices es la siguiente:

$$\begin{aligned} d(u, v) &= 2 & d(v, u) &= \infty & d(w, u) &= 1 \\ d(u, w) &= \infty & d(v, w) &= \infty & d(w, v) &= 3 \end{aligned}$$

Definición 1.3.19. Sea $G = (V, F, W)$ una digráfica con pesos, supongámos que enumeramos a los vértices de G como $V = \{v_1, \dots, v_n\}$. La *matriz de distancias* de G es la matriz de $n \times n$ con entradas a_{ij} , donde a_{ij} es la distancia de v_i hacia v_j , es decir, $a_{ij} = \{d(v_i, v_j) \mid v_i, v_j \in V\}$. La matriz de distancias (D) están dada por:

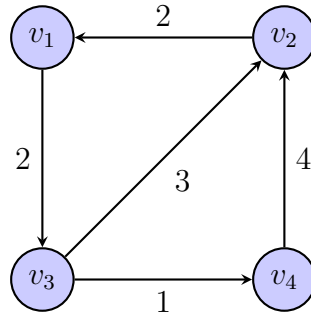
$$a_{ij} = \begin{cases} 0 & \text{cuando } i = j \\ \infty & \text{cuando } i \neq j \text{ y no existe un } v_i v_j \text{ - camino dirigido} \\ d(v_i, v_j) & \text{cuando } i \neq j \text{ y existe un } v_i v_j \text{ - camino dirigido} \end{cases}$$

Definición 1.3.20. Sea $G = (V, F, W)$ una digráfica con pesos tal que $V = \{v_1, \dots, v_n\}$, llamamos *matriz de pesos* de G a la matriz de orden $n \times n$, con entradas w_{ij} , donde

$$w_{ij} = \begin{cases} 0 & \text{cuando } i = j \\ w(v_i, v_j) & \text{cuando } i \neq j \text{ y } (v_i, v_j) \in F \\ \infty & \text{cuando } i \neq j \text{ y } (v_i, v_j) \notin F \end{cases}$$

Observación 1.3.21. En los casos en los que la matriz tenga demasiados elementos, para no causar confusión y que no sea tan cansado de ver se cambiará la notación de ∞ por $-$.

Ejemplo 1.3.8. Sea $G = (V, F, W)$ la digráfica con pesos siguiente:



Las matrices de distancia y pesos, D y W respectivamente, son las siguientes:

$$D = \begin{matrix} & \begin{matrix} v_1 & v_2 & v_3 & v_4 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} & \begin{pmatrix} 0 & 5 & 2 & 3 \\ 2 & 0 & 4 & 5 \\ 5 & 3 & 0 & 1 \\ 6 & 4 & 8 & 0 \end{pmatrix} \end{matrix}$$

Matriz de distancias de G

$$W = \begin{matrix} & \begin{matrix} v_1 & v_2 & v_3 & v_4 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} & \begin{pmatrix} 0 & \infty & 2 & \infty \\ 2 & 0 & \infty & \infty \\ \infty & 3 & 0 & 1 \\ \infty & 4 & \infty & 0 \end{pmatrix} \end{matrix}$$

Matriz de pesos de G

1.4. Algoritmos

1.4.1. Algoritmo de Dijkstra

El algoritmo de Dijkstra [3], concebido en 1956, y publicado por Edsger Dijkstra en 1959, encuentra la distancia desde un vértice s en una digráfica con pesos, al resto de los vértices.

Algoritmo 1.4.1. Sea $G = (V, F, w)$ una digráfica con pesos ($G = (V, A, w)$ una gráfica con pesos) de n vértices, el *algoritmo de Dijkstra* tiene como finalidad encontrar la distancia desde un vértice s a cualquier otro vértice.

Para el desarrollo del algoritmo, el conjunto de vértices V será partido en los conjuntos P y Q . Además un parámetro δ_v es asignado a cada vértice $v \in V$. Inicialmente todos los vértices están en Q . En el proceso los vértices alcanzables desde s se moverán de Q a P . Cuando un vértice v está en Q el parámetro δ_v es cota superior de $d(s, v)$, es decir, $\delta_v \geq d(s, v)$; finalmente cuando v se mueve a P $\delta_v = d(s, v)$. Una descripción formal del algoritmo es la siguiente:

Entrada. Una digráfica con pesos $G = (V, F, w)$, tal que $w(f) \geq 0$ ($w(a) \geq 0$) para toda $f \in F$.

Salida. El parámetro δ_v tal que para todo $v \in V$, $\delta_v = d(s, v)$.

Los pasos a seguir son los siguientes:

1. Establecer $P := \emptyset$, $Q := V$, $\delta_s = 0$ y $\delta_v := \infty$ para todo $v \in V - s$.
2. Si Q es no vacío hágase lo siguiente:

Encuentra un vértice $v \in Q$ tal que $\delta_v = \min\{\delta_u : u \in Q\}$.

Establecer $Q := Q - v$, $P := P \cup v$.

$\delta_u := \min\{\delta_u, \delta_v + w(v, u)\}$ para cada $u \in Q \cap N^+(v)$ (Definición 1.3.4).

3. FIN si $Q = \emptyset$, de lo contrario regresar al paso 2.

Para probar la efectividad del algoritmo de Dijkstra es suficiente probar la siguiente proposición.

Proposición 1.4.2. En cualquier iteración durante la ejecución del algoritmo, tenemos que

1. Para todo $v \in P$, $\delta_v = d(s, v)$.
2. Para todo $u \in Q$, δ_u es la distancia de s a u de la subdigráfica de G inducida por $P \cup u$.

Demostración. Bastará probar que para todo $v \in P$ $\delta_v = d(s, v)$, ya que **2** es la consecuencia de **1**. Se probará por contradicción,

Nótese que cuando añadimos u a P , δ_u no ha cambiado y deberá ser $d(s, u)$.

P.D. $\delta_u = d(s, u)$.

Sean $G = (V, F, w)$ la digráfica con pesos, $P = \emptyset$ y $Q = V$. Supongamos que **1** no funciona para $P' = P \cup v$, es decir, v es el último vértice añadido a P' , lo que significa que

$$\delta_v > d(s, v) \tag{1.1}$$

Sea C el camino de peso mínimo de s a v en G . Como $\delta_v > d(s, v)$, C debe contener al menos un vértice de $Q' = Q - v$. Sea $u \in C$ el primer vértice que no está en P y como u lo tomamos de Q' $u \neq v$. Por la desigualdad del triángulo

$$d(s, u) \leq d(s, v) + d(v, u) \implies d(s, u) \leq d(s, v) \tag{1.2}$$

Además, como es válido para P y Q que $\delta_x > d(s, x)$ para $x \in P$ y $u \in Q - v$ tenemos que

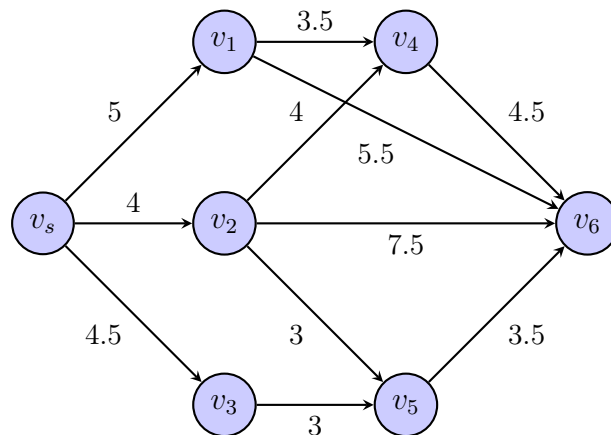
$$\delta_u = d(s, u) \tag{1.3}$$

De (1,1),(1,2) y (1,3) tenemos que

$$\delta_u = d(s, u) \leq d(s, v) < \delta_v \implies \delta_u < \delta_v \quad \textbf{Contradicción}$$

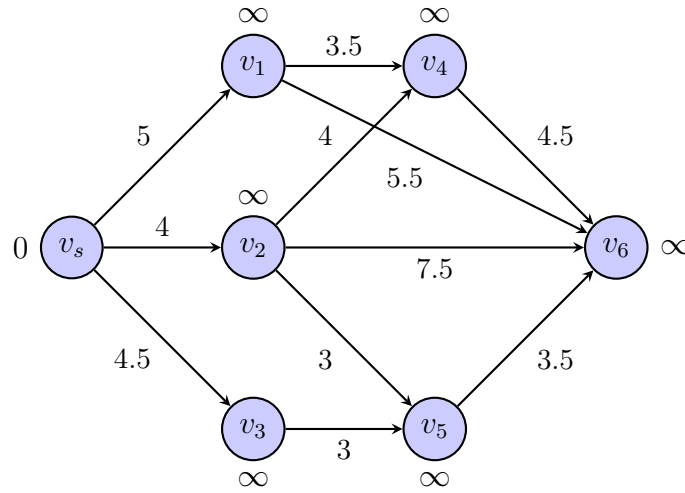
Se llega a una contradicción por la elección de v . Por lo tanto $\delta_v = d(s, v)$ para todo $v \in P$. □

Ejemplo 1.4.1. Sea G la siguiente digráfica con pesos:



Se desea encontrar la arborescencia de caminos mínimos desde el vértice v_s . Para ello se utilizará el algoritmo de Dijkstra.

Paso 1. Sea $P = \emptyset$ y $Q = V$ y $\delta_v = \infty$ para todo $v \in V - v_s$.

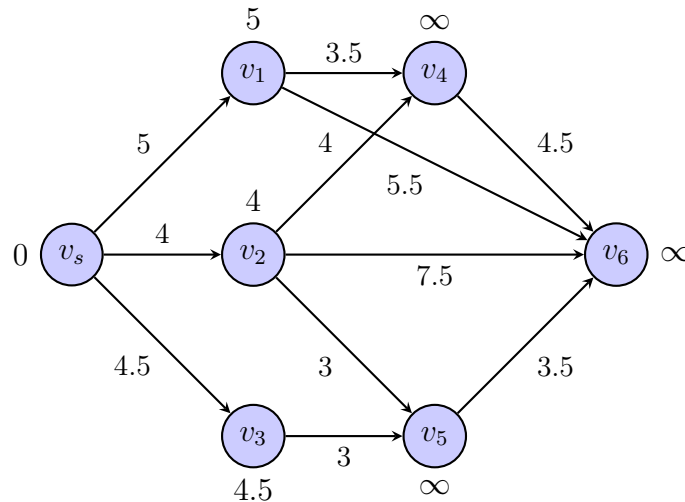


Paso 2. Como Q no es vacío,

Encontramos un vértice v_s tal que $\delta_{v_s} = 0 = \min\{\delta_u : u \in Q\}$.

Sea $Q_0 = Q - v_s$ y $P_0 = P \cup v_s$, y como $Q_0 \cap N^+(v_s) = \{v_1, v_2, v_3\}$

$$\delta_{v_1} = \min\{\infty, 0 + 5\} = 5, \quad \delta_{v_2} = \min\{\infty, 0 + 4\} = 4, \quad \delta_{v_3} = \min\{\infty, 0 + 4.5\} = 4.5$$



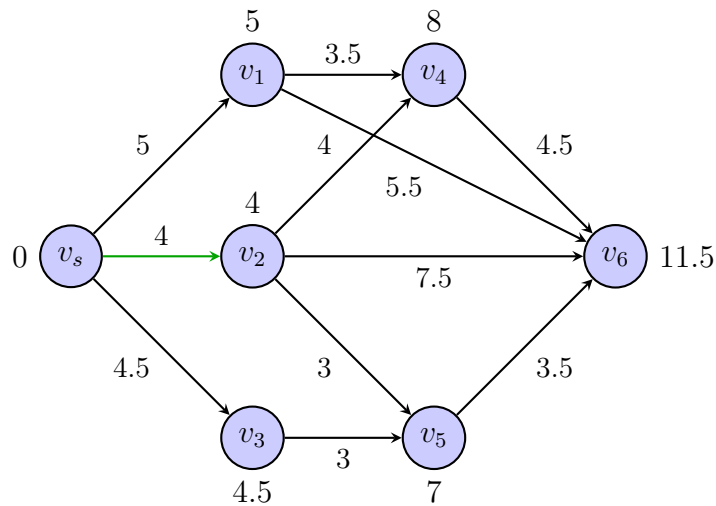
Paso 3. $Q_0 \neq \emptyset$, se va al paso 2.

Paso 2. Como Q_0 no es vacío,

Encontramos un vértice v_2 tal que $\delta_{v_2} = 4 = \min\{\delta_u : u \in Q_0\}$.

Sea $Q_1 = Q_0 - v_2$ y $P_1 = P_0 \cup v_2$, y como $Q_1 \cap N^+(v_2) = \{v_4, v_5, v_6\}$

$$\delta_{v_4} = \min\{\infty, 4 + 4\} = 8, \quad \delta_{v_5} = \min\{\infty, 4 + 3\} = 7, \quad \delta_{v_6} = \min\{\infty, 4 + 7.5\} = 11.5$$



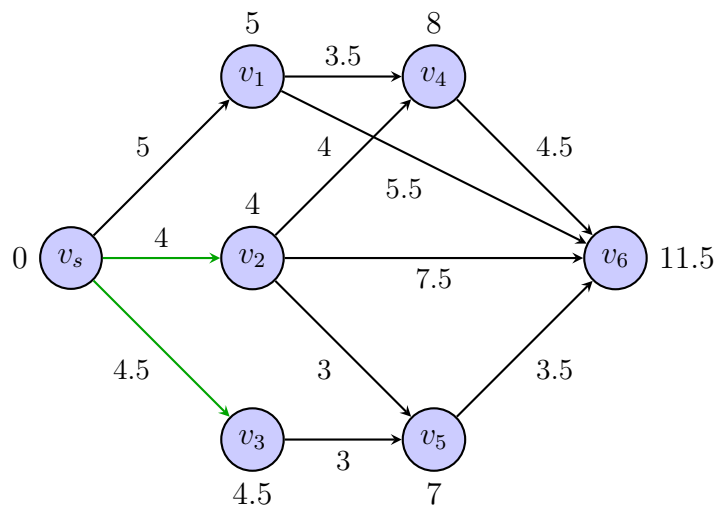
Paso 3. $Q_1 \neq \emptyset$, se va al paso 2.

Paso 2. Como Q_1 no es vacío,

Encontramos un vértice v_3 tal que $\delta_{v_3} = 4.5 = \min\{\delta_u : u \in Q_1\}$.

Sea $Q_2 = Q_1 - v_3$ y $P_2 = P_1 \cup v_3$, y como $Q_2 \cap N^+(v_3) = \{v_5\}$

$$\delta_{v_5} = \min\{7, 4.5 + 3\} = 7$$



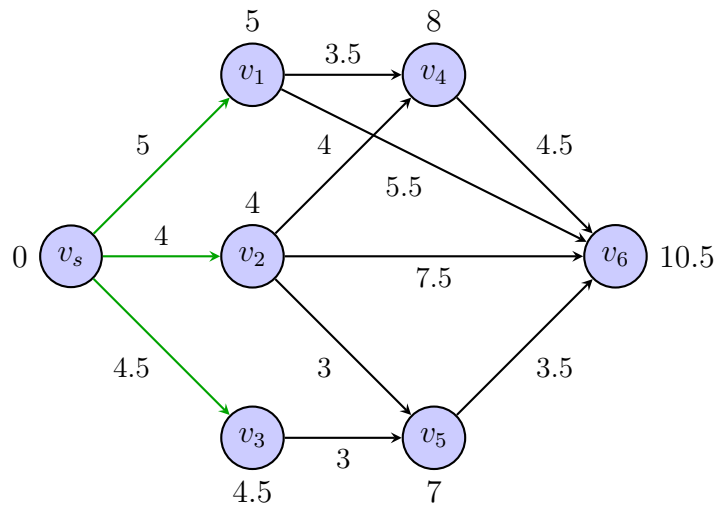
Paso 3. $Q_2 \neq \emptyset$, se va al paso 2.

Paso 2. Como Q_2 no es vacío,

Encontramos un vértice v_1 tal que $\delta_{v_1} = 5 = \min\{\delta_u : u \in Q_2\}$.

Sea $Q_3 = Q_2 - v_1$ y $P_3 = P_2 \cup v_1$, y como $Q_3 \cap N^+(v_1) = \{v_4, v_6\}$

$$\delta_{v_4} = \min\{8, 5 + 3.5\} = 8, \quad \delta_{v_6} = \min\{11.5, 5 + 5.5\} = 10.5$$



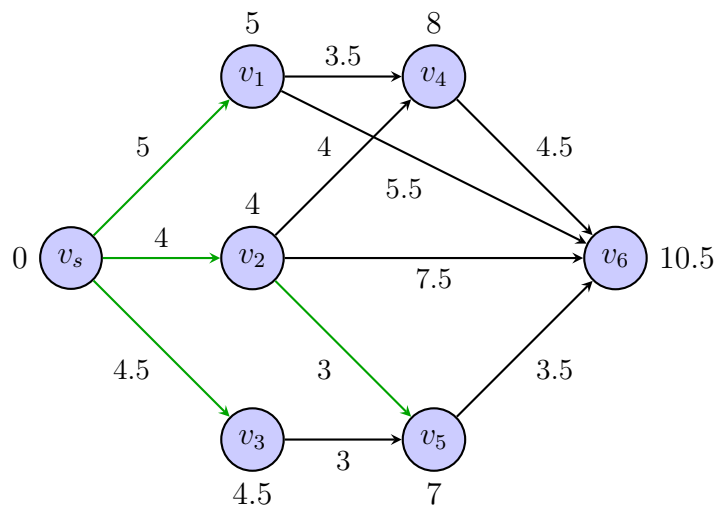
Paso 3. $Q_3 \neq \emptyset$, se va al paso 2.

Paso 2. Como Q_3 no es vacío,

Encontramos un vértice v_5 tal que $\delta_{v_5} = 7 = \min\{\delta_u : u \in Q_3\}$.

Sea $Q_4 = Q_3 - v_5$ y $P_4 = P_3 \cup v_5$, y como $Q_4 \cap N^+(v_5) = \{v_6\}$

$$\delta_{v_6} = \min\{10.5, 7 + 3.5\} = 10.5$$



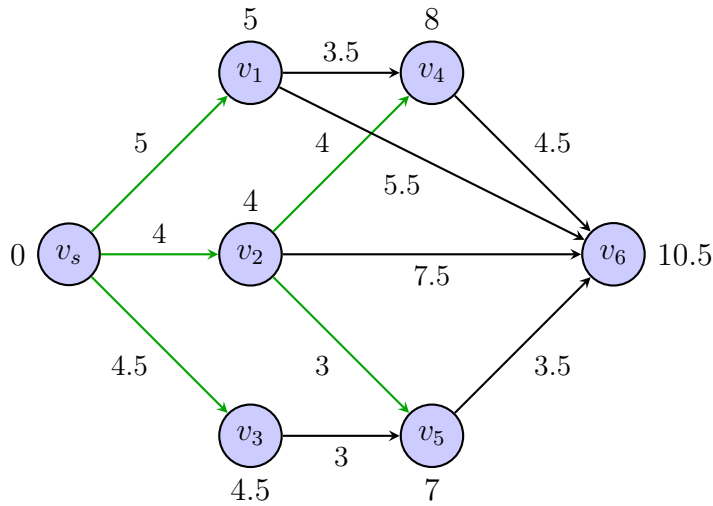
Paso 3. $Q_4 \neq \emptyset$, se va al paso 2.

Paso 2. Como Q_4 no es vacío,

Encontramos un vértice v_4 tal que $\delta_{v_4} = 8 = \min\{\delta_u : u \in Q_4\}$.

Sea $Q_5 = Q_4 - v_4$ y $P_5 = P_4 \cup v_4$, y como $Q_5 \cap N^+(v_4) = \{v_6\}$

$$\delta_{v_6} = \min\{10.5, 8 + 4.5\} = 10.5$$

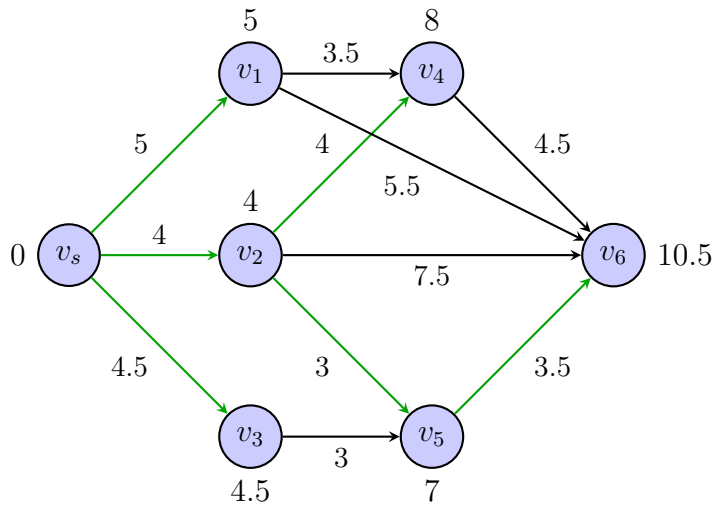


Paso 3. $Q_5 \neq \emptyset$, se va al paso 2.

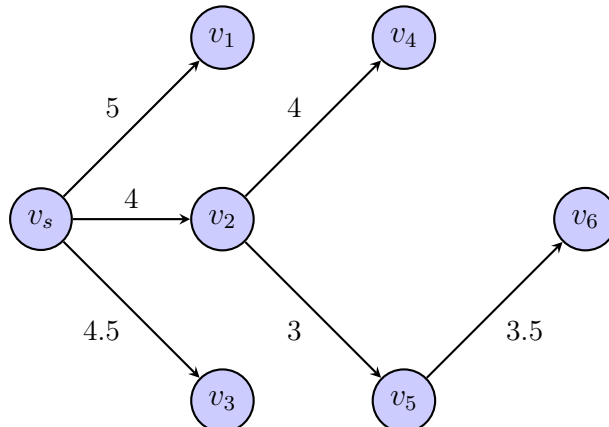
Paso 2. Como Q_5 no es vacío,

Encontramos un vértice v_6 tal que $\delta_{v_6} = 10.5 = \min\{\delta_u : u \in Q_5\}$.

Sea $Q_6 = Q_5 - v_6$ y $P_6 = P_5 \cup v_6$, y como $Q_6 \cap N^+(v_6) = \emptyset$



Paso 3. $P_6 = V$ y $Q_6 = \emptyset$, FIN. La arborescencia de caminos mínimos desde el vértice v_s es la siguiente:



1.4.2. Algoritmo de Floyd-Warshall

El algoritmo de Floyd-Warshall, desarrollado por Robert Floyd y Stephen Warshall en 1962, encuentra la distancia entre cada par de vértices de la digráfica.

Definición 1.4.3. [16] La *matriz* S de *sucesores* nos ofrece un historial de los vértices por los cuales tendremos el camino más corto. Está dada inicialmente por:

$$S = s_{ij} = \begin{matrix} & 1 & 2 & \dots & n \\ \begin{matrix} 1 \\ 2 \\ \vdots \\ n \end{matrix} & \begin{pmatrix} 1 & 2 & \dots & n \\ 1 & 2 & \dots & n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 2 & \dots & n \end{pmatrix} \end{matrix}$$

Definición 1.4.4. [8] Sea $G = (V, F, W)$ una digráfica con pesos de n vértices con $V = \{v_1, \dots, v_n\}$, la cual no contiene ciclos negativos, es decir, no contiene caminos cuyo peso sea negativo.

Utilizaremos una matriz W_0 , la cual es la matriz de pesos (definición 1.3.20), y una matriz S_0 , la cual es la matriz de sucesores (definición 1.4.3). Los pasos son los siguientes:

1. Tomar $k = 1$
2. Resaltar la fila y columna de $v_k \in V$. Estos valores no serán afectados
3. Para cualesquiera vértices v_i y v_j , tales que $v_i \neq v_k$ y $v_j \neq v_k$:

si $w_{ik} + w_{kj} < w_{ij}$ entonces:

- reemplazar la fila v_i y columna v_j de la matriz W_0 , es decir, w_{ij} , con $w_{ik} + w_{kj}$
- reemplazar la fila v_i y la columna v_j de la matriz S_0 , por el valor de la fila v_i y la columna v_k , es decir s_{ij} por s_{ik} .

Ahora se tienen nuevas matrices de peso y de sucesores, W_k y S_k respectivamente.

4. Parar si $k = n$, de lo contrario sea $k = k + 1$ y repita desde el paso 2.

Para probar la efectividad del algoritmo de Floyd-Warshall es suficiente probar que $W_k = D$, con D la matriz de distancias. Se demostrará por inducción sobre k .

Demostración. Sea $G = (V, F, W)$ una digráfica con pesos de k vértices, la cual no contiene ciclos negativos.

Nótese que si $w'_{ij} \in W_k$ y $w_{ij}, w_{ik}, w_{kj} \in W_{k-1}$ entonces

$$w'_{ij} = \text{mín}\{w_{ij}, w_{ik-1} + w_{k-1j}\} \quad (1)$$

Caso base: notemos que al inicio del algoritmo, es decir, cuando $k = 0$ tenemos que la matriz W_0 contiene los pesos entre cada par de vértices. Entonces w_{ij} es el peso de la flecha (i, j) .

Hipótesis de inducción: supongamos que W_l es efectiva para toda $l < k$.

Paso inductivo: denotemos a C_{ij} como un camino mínimo de v_i a v_j con $i, j \in \{1, 2, \dots, k-1\}$. C_{ij} no contiene a k por lo que el peso de C_{ij} es el peso del camino mínimo, es decir la distancia de v_i a v_j , es decir, $d(v_i, v_j) = w_{ij}$ con $w_{ij} \in W_{k-1}$.

Si C_{ij} no contiene a k , entonces C_{ij} es la unión de dos caminos C_{ik} y C_{kj} , de v_i a v_k , y de v_k a v_j respectivamente.

Entonces C_{ik} es un camino mínimo de v_i a v_k con $i, k \in \{1, 2, \dots, k-1\}$, y C_{kj} es un camino mínimo de v_k a v_j con $k, j \in \{1, 2, \dots, k-1\}$. Por lo que por hipótesis de inducción tenemos que $d(v_i, v_k) = w_{ik-1}$ con $w_{ik-1} \in W_{k-1}$, y que $d(v_k, v_j) = w_{k-1j}$ con $w_{k-1j} \in W_{k-1}$. Por lo tanto

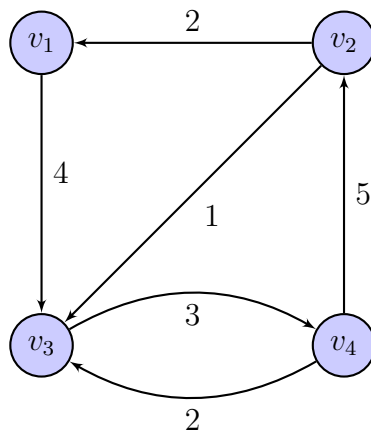
$$w_{ij} = d(v_i, v_j) \leq d(v_i, v_k) + d(v_k, v_j) = w_{ik-1} + w_{k-1j}$$

$$w_{ij} \leq w_{ik-1} + w_{k-1j}$$

Entonces como el algoritmo es recursivo como se ve en (1), al considerar a ambos y tomar el mínimo de estos se tiene que W_k es efectiva, y a su vez $W_k = D$, donde D es la matriz de distancias.

□

Ejemplo 1.4.2. Sea G la siguiente multidigráfica con pesos en las flechas:



Recordando la Definición 1.3.20 hacemos la matriz de pesos de G , la cual será W_0 , la matriz inicial:

$$W_0 = \begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & 0 & \infty & 4 & \infty \\ v_2 & 2 & 0 & 1 & \infty \\ v_3 & \infty & \infty & 0 & 3 \\ v_4 & \infty & 5 & 2 & 0 \end{array}$$

Para encontrar el camino mínimo entre todo par de vértices, aplicamos el algoritmo de Floyd-Warshall.

Las matrices W_0 y S_0 :

	v_1	v_2	v_3	v_4
v_1	0	∞	4	∞
v_2	2	0	1	∞
v_3	∞	∞	0	3
v_4	∞	5	2	0

W_0

	v_1	v_2	v_3	v_4
v_1	1	2	3	4
v_2	1	2	3	4
v_3	1	2	3	4
v_4	1	2	3	4

S_0

Paso 1: tomamos $k = 1$.

Paso 2: Resaltamos la fila y columna de v_1 .

	v_1	v_2	v_3	v_4
v_1	0	∞	4	∞
v_2	2	0	1	∞
v_3	∞	∞	0	3
v_4	∞	5	2	0

D_1

	v_1	v_2	v_3	v_4
v_1	1	2	3	4
v_2	1	2	3	4
v_3	1	2	3	4
v_4	1	2	3	4

S_1

Paso 3: Para todos los vértices $v_i \neq v_1$ y $v_j \neq v_1$ vemos si:

$$w_{i1} + w_{1j} < w_{ij} \quad i \neq 1, \quad j \neq 1$$

$$w_{21} + w_{12} = 2 + \infty \not< 0 = w_{22}$$

$$w_{21} + w_{13} = 2 + 4 \not< 1 = w_{23}$$

$$w_{21} + w_{14} = 2 + \infty \not< \infty = w_{24}$$

$$w_{31} + w_{12} = \infty + \infty \not< \infty = w_{32}$$

$$w_{31} + w_{13} = \infty + 4 \not< 0 = w_{33}$$

$$w_{31} + w_{14} = \infty + \infty \not< 3 = w_{34}$$

$$w_{41} + w_{12} = \infty + \infty \not< 5 = w_{42}$$

$$w_{41} + w_{13} = \infty + 4 \not< 2 = w_{43}$$

$$w_{41} + w_{14} = \infty + \infty \not< 0 = w_{44}$$

Como no hubo cambios, se tienen que $W_1 = W_2$ y $S_1 = S_2$.

Paso 4: $k = 1 \neq n = 4$, entonces tomamos $k = k + 1 = 2$ y repetir desde el paso 2.

Paso 2: Resaltamos la fila y columna de v_2 .

	v_1	v_2	v_3	v_4
v_1	0	∞	4	∞
v_2	2	0	1	∞
v_3	∞	∞	0	3
v_4	∞	5	2	0

W_2

	v_1	v_2	v_3	v_4
v_1	1	2	3	4
v_2	1	2	3	4
v_3	1	2	3	4
v_4	1	2	3	4

S_2

Paso 3: Para todos los vértices $v_i \neq v_2$ y $v_j \neq v_2$ vemos si:

$$w_{i2} + w_{2j} < w_{ij} \quad i \neq 2, \quad j \neq 2$$

$$w_{12} + w_{21} = \infty + 2 \not< 0 = w_{11}$$

$$w_{12} + w_{23} = \infty + 1 \not< 4 = w_{13}$$

$$w_{12} + w_{24} = \infty + \infty \not< \infty = w_{14}$$

$$w_{32} + w_{21} = \infty + 2 \not< \infty = w_{31}$$

$$w_{32} + w_{23} = \infty + 1 \not< 0 = w_{33}$$

$$w_{32} + w_{24} = \infty + \infty \not< 3 = w_{34}$$

$$\bullet \quad w_{42} + w_{21} = 2 + 5 < \infty = w_{41}$$

$$w_{42} + w_{23} = 5 + 1 \not< 2 = w_{43}$$

$$w_{42} + w_{24} = 5 + \infty \not< 0 = w_{44}$$

Reemplazamos en donde sí se cumple la desigualdad (\bullet):

- w_{41} de la matriz W_2 con $w_{42} + w_{21} = 2 + 5 = 7$; s_{41} de la matriz S_2 , por s_{42} .

Las matrices resultantes serán W_3 y S_3 .

	v_1	v_2	v_3	v_4
v_1	0	∞	4	∞
v_2	2	0	1	∞
v_3	∞	∞	0	3
v_4	7	5	2	0

W_3

	v_1	v_2	v_3	v_4
v_1	1	2	3	4
v_2	1	2	3	4
v_3	1	2	3	4
v_4	2	2	3	4

S_3

Paso 4: $k = 2 \neq n = 4$, entonces tomamos $k = k + 1 = 3$ y repetir desde el paso 2.

Paso 2: Resaltamos la fila y columna de v_3 .

	v_1	v_2	v_3	v_4
v_1	0	∞	4	∞
v_2	2	0	1	∞
v_3	∞	∞	0	3
v_4	7	5	2	0

W_3

	v_1	v_2	v_3	v_4
v_1	1	2	3	4
v_2	1	2	3	4
v_3	1	2	3	4
v_4	2	2	3	4

S_3

Paso 3: Para todos los vértices $v_i \neq v_3$ y $v_j \neq v_3$ vemos si:

$$w_{i3} + w_{3j} < w_{ij} \quad i \neq 3, \quad j \neq 3$$

$$w_{13} + w_{31} = 4 + \infty \not< 0 = w_{11}$$

$$w_{13} + w_{32} = 4 + \infty \not< \infty = w_{12}$$

- $w_{13} + w_{34} = 4 + 3 < \infty = w_{14}$

$$w_{23} + w_{31} = \infty + 1 \not< 2 = w_{21}$$

$$w_{23} + w_{32} = \infty + 1 \not< 0 = w_{22}$$

- $w_{23} + w_{34} = 1 + 3 < \infty = w_{24}$

$$w_{43} + w_{31} = 2 + \infty \not< 7 = w_{41}$$

$$w_{43} + w_{32} = 2 + \infty \not< 5 = w_{42}$$

$$w_{43} + w_{34} = 3 + 2 \not< 0 = w_{44}$$

Reemplazamos en donde sí se cumple la desigualdad (●):

- w_{14} de la matriz W_3 con $w_{13} + w_{34} = 4 + 3 = 7$; s_{14} de la matriz S_3 , por el valor de s_{13} .
- w_{24} de la matriz W_3 con $w_{23} + w_{34} = 1 + 3 = 4$; s_{24} de la matriz S_3 , por el valor de s_{23} .

Las matrices resultantes serán W_4 y S_4 .

	v_1	v_2	v_3	v_4
v_1	0	∞	4	7
v_2	2	0	1	4
v_3	∞	∞	0	3
v_4	7	5	2	0

W_4

	v_1	v_2	v_3	v_4
v_1	1	2	3	3
v_2	1	2	3	3
v_3	1	2	3	4
v_4	2	2	3	4

S_4

Paso 4: $k = 3 \neq n = 4$, entonces tomamos $k = k + 1 = 4$ y repetir desde el paso 2.

Paso 2: Resaltamos la fila y la columna de v_4 .

	v_1	v_2	v_3	v_4
v_1	0	∞	4	7
v_2	2	0	1	4
v_3	∞	∞	0	3
v_4	7	5	2	0

W_4

	v_1	v_2	v_3	v_4
v_1	1	2	3	3
v_2	1	2	3	3
v_3	1	2	3	4
v_4	2	2	3	4

S_4

Paso 3: Para todos los vértices $v_i \neq v_4$ y $v_j \neq v_4$ vemos si:

$$w_{i4} + w_{4j} < w_{ij} \quad i \neq 4, \quad j \neq 4$$

$$w_{14} + w_{41} = 7 + 7 \not< 0 = w_{11}$$

● $w_{14} + w_{42} = 7 + 5 < \infty = w_{12}$

$$w_{14} + w_{43} = 7 + 2 \not< 4 = w_{13}$$

$$w_{24} + w_{41} = 4 + 7 \not\leq 2 = w_{21}$$

$$w_{24} + w_{42} = 4 + 5 \not\leq 0 = w_{22}$$

$$w_{24} + w_{43} = 4 + 2 \not\leq 1 = w_{23}$$

- $w_{34} + w_{41} = 3 + 7 < \infty = w_{31}$

- $w_{34} + w_{42} = 3 + 5 < \infty = w_{32}$

$$w_{34} + w_{43} = 3 + 2 \not\leq 0 = w_{33}$$

Reemplazamos en donde sí se cumple la desigualdad (●):

- w_{12} de la matriz W_4 con $w_{14} + w_{42} = 7 + 5 = 12$; s_{12} de la matriz S_4 , por el valor de s_{14} .
- w_{31} de la matriz W_4 con $w_{34} + w_{41} = 7 + 3 = 10$; s_{31} de la matriz S_4 , por el valor de s_{34} .
- w_{32} de la matriz W_4 con $w_{34} + w_{42} = 5 + 3 = 8$; s_{32} de la matriz S_4 , por el valor de s_{34} .

Las matrices resultantes serán W_f y S_f .

	v_1	v_2	v_3	v_4
v_1	0	12	4	7
v_2	2	0	1	4
v_3	10	8	0	3
v_4	7	5	2	0

W_f

	v_1	v_2	v_3	v_4
v_1	1	3	3	3
v_2	1	2	3	3
v_3	4	4	3	4
v_4	2	2	3	4

S_f

Paso 4: $k = 4 = n$, fin.

Tenemos como resultado a las matrices finales W_f , la cual es la matriz de distancias D (definición 1.3.19), y S_f , la matriz final de sucesores (definición 1.4.3).

	v_1	v_2	v_3	v_4
v_1	0	12	4	7
v_2	2	0	1	4
v_3	10	8	0	3
v_4	7	5	2	0

 D

	v_1	v_2	v_3	v_4
v_1	1	3	3	3
v_2	1	2	3	3
v_3	4	4	3	4
v_4	2	2	3	4

 S_f

Si deseamos conocer:

- el camino mínimo entre los vértices v_2 y v_4 , por D la distancia mínima a recorrer es de 4 unidades, y por S_f el camino a seguir sería:
 - a) nos fijamos en la fila del vértice origen, es decir v_2 , y la columna del vértice destino, es decir v_4 , la cual nos indica que el sucesor de v_2 es v_3 .
 - b) ahora nos fijamos en la fila v_3 , el cual es el vértice sucesor de v_2 , y la columna del vértice destino, es decir v_4 , la cual nos indica que el sucesor de v_3 es v_4 .

Entonces como ya llegamos a v_4 , el cual es el vértice destino, el camino mínimo es:

$$C = (v_2, (v_2, v_3), v_3, (v_3, v_4), v_4)$$

- el camino mínimo entre los vértices v_3 y v_1 , por D_f la distancia mínima a recorrer es de 10 unidades, y por S_f el camino a seguir sería:
 - a) nos fijamos en la fila del vértice origen, es decir v_3 , y la columna del vértice destino, es decir v_1 , la cual nos indica que el sucesor de v_3 es v_4 .
 - b) ahora nos fijamos en la fila v_4 , el cual es el sucesor de v_3 , y la columna del vértice destino, es decir v_1 , la cual nos indica que el sucesor de v_4 es v_2 .
 - c) ahora nos fijamos en la fila v_2 , el cual es el vértice sucesor de v_4 , y la columna del vértice destino, es decir v_1 , la cual nos indica que el sucesor de v_2 es v_1 .

Entonces como ya llegamos a v_1 , el cual es el vértice destino, el camino es:

$$C = (v_3, (v_3, v_4), v_4, (v_4, v_2), v_2, (v_2, v_1), v_1)$$

Observación 1.4.5. Una instancia distinta del algoritmo de Floyd-Warshall la cual no considera a la matriz de sucesores, es la siguiente:

Utilizar una matriz $W_k[i][j]$. Inicialmente $W_0[i][j] = W$, la cual es la matriz de pesos (definición 1.5.9.).

Los pasos a seguir son los siguientes:

1. Tomar $k = 1$
2. Se calcula para cada $i \neq k$ y $j \neq k$, aplicando:

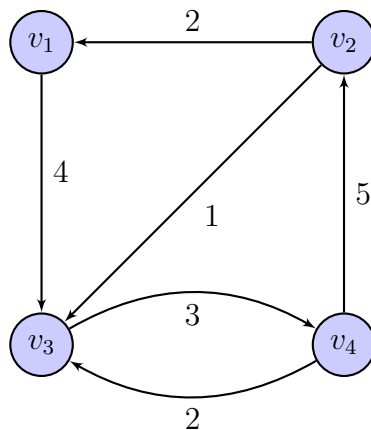
$$W_k[i][j] = \min\{W_{k-1}[i][j], W_{k-1}[i][k] + W_{k-1}[k][j]\} \quad \forall i \neq j$$

3. Parar si $k = n$, de lo contrario sea $k = k + 1$ y repita desde el paso 2.

Esta variante da pie, de una forma más sencilla, al desarrollo de un programa en el ambiente Python, el cual hará que el tiempo invertido en el calculo de cada una de las iteraciones sera menor que el necesario para la primera variante del algoritmo de Floyd-Warshall.

Ejemplo 1.4.3. Retomaremos el problema del Ejemplo 1.4.2, para visualizar que ésta concluye en un número menor de pasos.

Sea G la multidigráfica siguiente con pesos en las flechas:



Utilizaremos una matriz $W_k[i][j]$, inicialmente $W_0[i][j] = W$, la cual es, recordando la definición 1.2.6, la matriz de pesos de la digráfica G . Entonces tenemos que:

$$W_0[i][j] = \begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & 0 & \infty & 4 & \infty \\ v_2 & 2 & 0 & 1 & \infty \\ v_3 & \infty & \infty & 0 & 3 \\ v_4 & \infty & 5 & 2 & 0 \end{array}$$

Para encontrar el camino mínimo entre todo par de vértices, aplicamos ésta variante del algoritmo de Floyd-Warshall.

Paso 1: tomamos $k = 1$.

Paso 2: Se calcula para cada $i \neq 1$ y $j \neq 1$, aplicando:

$$W_1[i][j] = \min\{W_0[i][j], W_0[i][k] + W_0[k][j]\} \quad \forall i \neq j$$

$$W_1[2][3] = \min\{W_0[2][3], W_0[2][1] + W_0[1][3]\} = \min\{1, 2 + 4\} = 1$$

$$W_1[2][4] = \min\{W_0[2][4], W_0[2][1] + W_0[1][4]\} = \min\{\infty, 2 + \infty\} = \infty$$

$$W_1[3][2] = \min\{W_0[3][2], W_0[3][1] + W_0[1][2]\} = \min\{\infty, \infty + \infty\} = \infty$$

$$W_1[3][4] = \min\{W_0[3][4], W_0[3][1] + W_0[1][4]\} = \min\{3, \infty + \infty\} = 3$$

$$W_1[4][2] = \min\{W_0[4][2], W_0[4][1] + W_0[1][2]\} = \min\{5, \infty + \infty\} = 5$$

$$W_1[4][3] = \min\{W_0[4][3], W_0[4][1] + W_0[1][3]\} = \min\{2, \infty + 4\} = 2$$

La matriz $W_1[i][j]$ es la siguiente (resaltados en color naranja si fueron modificados de la matriz anterior):

$$W_1[i][j] = \begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & 0 & \infty & 4 & \infty \\ v_2 & 2 & 0 & 1 & \infty \\ v_3 & \infty & \infty & 0 & 3 \\ v_4 & \infty & 5 & 2 & 0 \end{array}$$

Paso 3: $k = 1 \neq 4 = n$, entonces sea $k = 1 + 1 = 2$ y se repite desde el paso 2.

Paso 2: Se calcula para cada $i \neq 2$ y $j \neq 2$, aplicando:

$$W_2[i][j] = \min\{W_1[i][j], W_1[i][k] + W_1[k][j]\} \quad \forall i \neq j$$

$$W_2[1][3] = \min\{W_1[1][3], W_1[1][2] + W_1[2][3]\} = \min\{4, \infty + 1\} = 4$$

$$W_2[1][4] = \min\{W_1[1][4], W_1[1][2] + W_1[2][4]\} = \min\{\infty, \infty + \infty\} = \infty$$

$$W_2[3][1] = \min\{W_1[3][1], W_1[3][2] + W_1[2][1]\} = \min\{\infty, \infty + 2\} = \infty$$

$$W_2[3][4] = \min\{W_1[3][4], W_1[3][2] + W_1[2][4]\} = \min\{3, \infty + \infty\} = 3$$

$$W_2[4][1] = \min\{W_1[4][1], W_1[4][2] + W_1[2][1]\} = \min\{\infty, 5 + 2\} = 7$$

$$W_2[4][3] = \min\{W_1[4][3], W_1[4][2] + W_1[2][3]\} = \min\{2, 5 + 1\} = 2$$

La matriz $W_2[i][j]$ es la siguiente (resaltados en color naranja si fueron modificados de la matriz anterior):

$$W_2[i][j] = \begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & 0 & \infty & 4 & \infty \\ v_2 & 2 & 0 & 1 & \infty \\ v_3 & \infty & \infty & 0 & 3 \\ v_4 & 7 & 5 & 2 & 0 \end{array}$$

Paso 3: $k = 2 \neq 4 = n$, entonces sea $k = 2 + 1 = 3$ y se repite desde el paso 2.

Paso 2: Se calcula para cada $i \neq 3$ y $j \neq 3$, aplicando:

$$W_3[i][j] = \min\{W_2[i][j], W_2[i][k] + W_2[k][j]\} \quad \forall i \neq j$$

$$W_3[1][2] = \min\{W_2[1][2], W_2[1][3] + W_2[3][2]\} = \min\{\infty, 4 + \infty\} = \infty$$

$$W_3[1][4] = \min\{W_2[1][4], W_2[1][3] + W_2[3][4]\} = \min\{\infty, 4 + 3\} = 7$$

$$W_3[2][1] = \min\{W_2[2][1], W_2[2][3] + W_2[3][1]\} = \min\{2, 1 + \infty\} = 2$$

$$W_3[2][4] = \min\{W_2[2][4], W_2[2][3] + W_2[3][4]\} = \min\{\infty, 1 + 3\} = 4$$

$$W_3[4][1] = \min\{W_2[4][1], W_2[4][3] + W_2[3][1]\} = \min\{7, 2 + \infty\} = 7$$

$$W_3[4][2] = \min\{W_2[4][2], W_2[4][3] + W_2[3][2]\} = \min\{5, 1 + \infty\} = 5$$

La matriz $W_3[i][j]$ es la siguiente (resaltados en color naranja si fueron modificados de la matriz anterior):

$$W_3[i][j] = \begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & 0 & \infty & 4 & 7 \\ v_2 & 2 & 0 & 1 & 4 \\ v_3 & \infty & \infty & 0 & 3 \\ v_4 & 7 & 5 & 2 & 0 \end{array}$$

Paso 3: $k = 3 \neq 4 = n$, entonces sea $k = 3 + 1 = 4$ y se repite desde el paso 2.

Paso 2: Se calcula para cada $i \neq 4$ y $j \neq 4$, aplicando:

$$W_4[i][j] = \min\{W_3[i][j], W_3[i][k] + W_3[k][j]\} \quad \forall i \neq j$$

$$W_4[1][2] = \min\{W_3[1][2], W_3[1][4] + W_3[4][2]\} = \min\{\infty, 7 + 5\} = 12$$

$$W_4[1][3] = \min\{W_3[1][3], W_3[1][4] + W_3[4][3]\} = \min\{4, 7 + 2\} = 4$$

$$W_4[2][1] = \min\{W_3[2][1], W_3[2][4] + W_3[4][1]\} = \min\{2, 4 + 7\} = 2$$

$$W_4[2][3] = \min\{W_3[2][3], W_3[2][4] + W_3[4][3]\} = \min\{1, 4 + 2\} = 1$$

$$W_4[3][1] = \min\{W_3[3][1], W_3[3][4] + W_3[4][1]\} = \min\{\infty, 3 + 7\} = 10$$

$$W_4[3][2] = \min\{W_3[3][2], W_3[3][4] + W_3[4][2]\} = \min\{\infty, 3 + 5\} = 8$$

La matriz $W_4[i][j]$ es la siguiente (resaltados en color naranja si fueron modificados de la matriz anterior):

$$W_4[i][j] = \begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & 0 & 12 & 4 & 7 \\ v_2 & 2 & 0 & 1 & 4 \\ v_3 & 10 & 8 & 0 & 3 \\ v_4 & 7 & 5 & 2 & 0 \end{array}$$

Paso 3: $k = 4 = n$, fin.

Tenemos como resultado la matriz $W_4[i][j]$, la cual es la matriz de distancias D (definición 1.3.19), además de que coincide con el resultado obtenido anteriormente.

Capítulo 2

Ruta de autobuses

2.1. Conceptos

Definición 2.1.1. [14] Sea G una gráfica conexa, denotamos a el máximo número de árboles de expansión ajenos por aristas como $\sigma(G)$, también conocido por el número de empaquetamiento de árboles de expansión (*EAE*, en inglés *STP*, *spanning tree packing number*).

Definición 2.1.2. Una *partición* de un conjunto A es una colección de subconjuntos de A , digamos A_1, A_2, \dots, A_n , que satisfacen lo siguiente:

- La unión de todos los subconjuntos es igual al conjunto dado.

$$A_1 \cup A_2 \cup \dots \cup A_n = A$$

- Todos los subconjuntos son ajenos entre sí.

$$A_i \cap A_j = \emptyset \quad i \neq j \quad \forall i, j \in \{1, 2, \dots, n\}$$

- Ningún subconjunto es vacío.

$$A_i \neq \emptyset \quad \forall i \in \{1, 2, \dots, n\}$$

Definición 2.1.3. Dada una partición P de $V(G)$, denotamos por $A_P(G)$ al conjunto de aquellas aristas de G que unen vértices pertenecientes a diferentes elementos de P .

Teorema 2.1.4 (Nash-Williams, Tutte). [14] Una gráfica conexa G , tiene al menos k árboles de expansión ajenos por aristas si y sólo si, para cada partición P de $V(G)$ en $r = |P|$ partes, hay al menos $k(r - 1)$ aristas entre las partes, es decir,

$$|A_P(G)| \geq k(r - 1) \quad \text{para toda partición } P \text{ de } V(G)$$

Ejemplo 2.1.1. Tomaría demasiado tiempo considerar todas las particiones de $V(G)$, para poder obtener una cota superior para k , así que tomamos dos particiones particulares que son muy significativas, puesto que sirven para recalcar que entre mayor sea nuestra pátición nos otorga más información de como es k :

Sea $G = (V(G), A(G))$ la gráfica de la Figura 2.1, sea P una partición de $V(G) = \{v_1, v_2, v_3, v_4, v_5, v_6\}$, tal que se generen los subconjuntos $V_1(G) = \{v_1, v_2, v_3\}$, $V_2(G) = \{v_4\}$ y $V_3(G) = \{v_5, v_6\}$. La gráfica G es la siguiente:

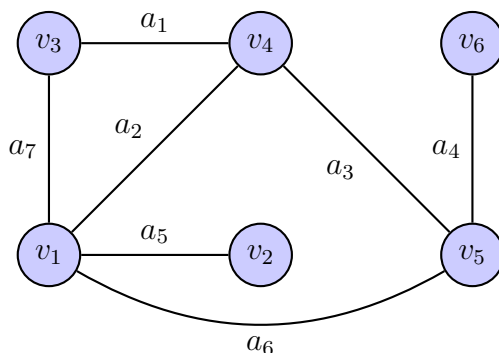


FIGURA 2.1

Como $A_P(G) = \{a_1, a_2, a_6, a_3\}$, entonces $|A_P(G)| = 4$. Por lo tanto; dado que

$$|A_P(G)| \geq k(r - 1) \implies 4 \geq k(3 - 1)$$

$$k \leq \frac{4}{(3 - 1)} = 2 \implies k \leq 2$$

Así, el máximo número de árboles de expansión es a lo más 2.

Ahora tomando la partición $P = \{\{v_1\}, \{v_2\}, \{v_3\}, \{v_4\}, \{v_5\}, \{v_6\}\}$, retomando el ejemplo anterior:

$$r = |P| = 6 \quad |A_P(G)| = 7$$

$$k \leq \frac{|A_P(G)|}{(r - 1)} = \frac{7}{(6 - 1)} = \frac{7}{5}$$

Entonces como k es entero, $\sigma(G) = 1$, es decir, el máximo número de árboles de expansión ajenos por aristas de la gráfica es 1.

Observación 2.1.5. Nótemos que la aproximación mas cercana al valor de k , se obtiene cuando se toma $|P| = |V(G)|$, ya que con esta condición obtenemos la cota superior para k .

2.2. Planteamiento del problema

En una ciudad existe un número de compañías de autobuses que quieren cubrir el servicio de transporte, a través de diferentes estaciones, de tal manera que en cada ruta los pasajeros puedan llegar desde cada una de las estaciones hasta cualquier otra aunque no sea de manera directa. Sin embargo, la administración vial no permitirá que dos compañías operen sobre un mismo tramo de camino. ¿Cuál es el máximo número de compañías que puede admitir la administración vial?

2.3. Modelación matemática del problema

Problema 2.3.1. La Secretaría de Comunicaciones y Transportes (SCT) realiza un proyecto integral de mejoramiento de vías, cambio de durmientes y de balasto de la línea 14 del metro de la Ciudad de México, comprende un total de 7 estaciones, las cuales durante un periodo no mayor a 9 meses estarán fuera de servicio.

Dado que la movilidad no se detiene, la SCT ofrecerá rutas de transporte colectivo.

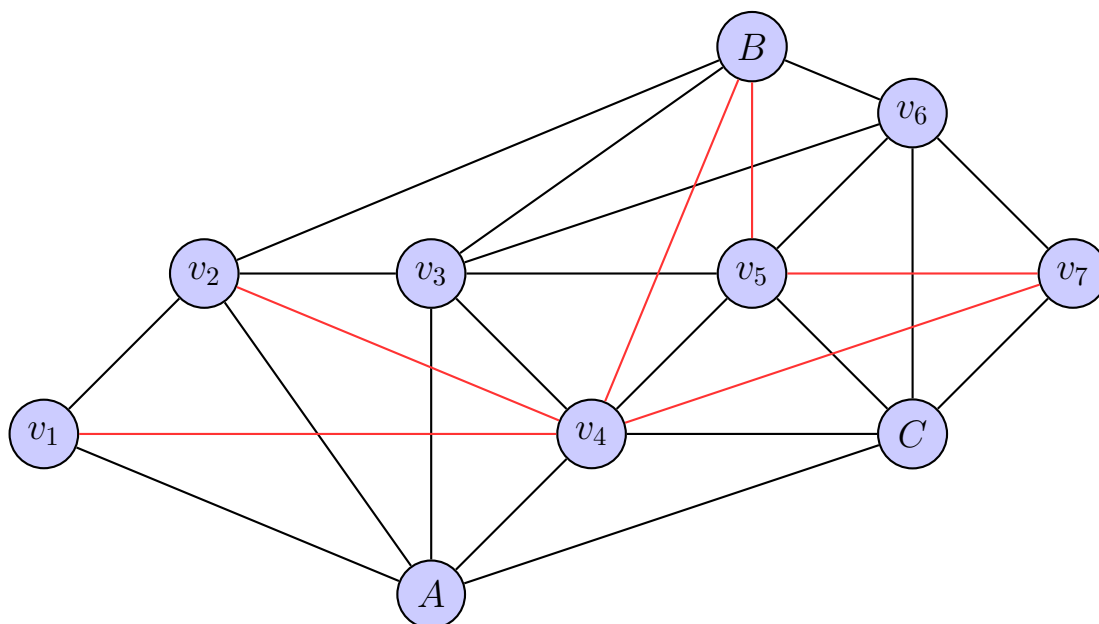
Se necesitan establecer una o varias rutas de transporte, que comuniquen las 7 estaciones de la línea 14, de tal manera que dos rutas distintas no usen una misma calle.

Cada ruta de autobuses produce una subgráfica, la cual deberá incluir a todos los vértices y debe ser conexa, de tal manera que cada pasajero pueda viajar desde un punto a otro a través de dicha ruta. Entonces el problema se reduce a descomponer la gráfica en en el máximo número de árboles de expansión ajenos por aristas.

Este problema se modelará utilizando teoría de gráficas de la siguiente manera:

Sea G la gráfica, tal que dibujemos un vértice por cada una de las 7 estaciones de la

línea 14 ($\{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$), y tres vértices más por los 3 puntos ($\{A, B, C\}$), que fueron clasificados como de mayor concurrencia. Dibujemos aristas de color **negro** en representación de conexiones directas por caminos terráneos, y aristas de color **rojo**, en representación de conexiones directas por caminos subterráneos. Esta convención de colores se usa porque al tratarse de un problema real, la subgráfica de los caminos terráneos (sin considerar variaciones de elevación como puentes) y subterráneos es plana, aunque un camino terráneo con un subterráneo se pueden intersectar en un punto que no es vértice de la gráfica. La gráfica G es la siguiente:



Solución 2.3.2. Tenemos 7 estaciones, tres puntos de alta concurrencia, 26 caminos, un total de 10 vértices y 26 aristas. Cada compañía de autobuses necesita una ruta que la conecte con cada estación, considerando que cada compañía debe tomar una ruta distinta. Necesitamos dividir a la gráfica G en el máximo número de árboles de expansión ajenos por aristas.

Por la observación 2.1.5 tomamos $|P| = |V(G)| = 10$, es decir, $P = \{\{v_1\}, \{v_2\}, \{v_3\}, \{v_4\}, \{v_5\}, \{v_6\}, \{v_7\}, \{A\}, \{B\}, \{C\}\}$, entonces $A_P(G) = A(G) \implies |A_P(G)| = 26$, por lo tanto:

$$r = |P| = 10 \quad |A_P(G)| = 26$$

$$k \leq \frac{|A_P(G)|}{(r-1)} = \frac{26}{(10-1)} = \frac{26}{9}$$

Por lo tanto el entero que se aproxima mejor al valor de k es 2, así $\sigma(G) = 2$. Entonces cada subgráfica de cada ruta de autobuses debe de tener (por el Teorema 1.2.9) 9 aristas, por lo que a lo más se tendrán 2 rutas, tales que no compartan caminos. La siguiente es una asignación correcta para cada una de las dos posibles rutas de autobuses, de las cuales pueden ser recorridas parcial o totalmente por una o más unidades:

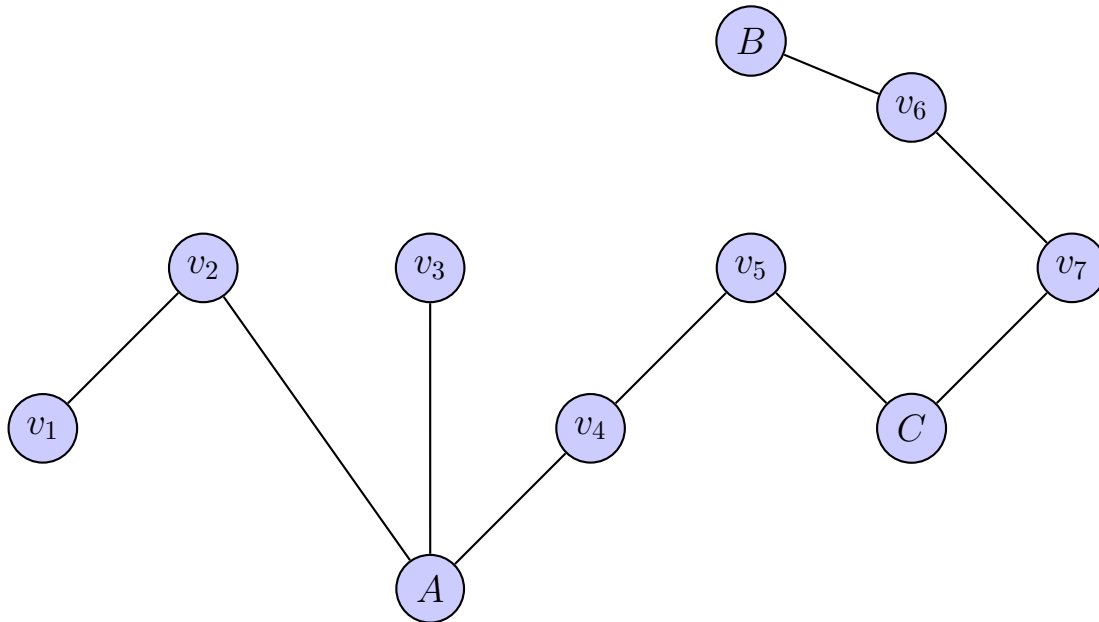


FIGURA 2.2: Ruta 1

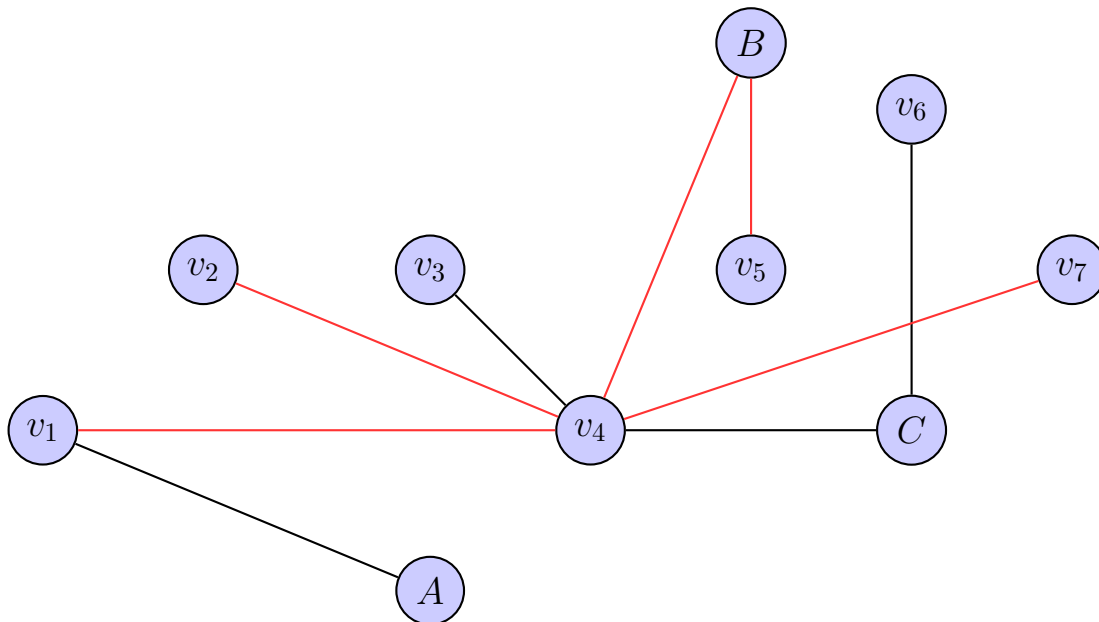


FIGURA 2.3: Ruta 2

Problema 2.3.3. La Secretaría de Turismo y Cultura del estado de Morelos desea establecer rutas turísticas para promocionar la belleza del estado, cada ruta turística tendrá que recorrer un total de 9 municipios, y se planea que para que se tenga una vista distinta durante el recorrido de un pueblo a otro, y una apreciación desde distintas perspectivas se planea que las rutas sean distintas. Por lo tanto, los encargados se dan a la tarea de encontrar el máximo número de rutas turísticas que van a circular por el estado.

Los municipios son los siguientes:

- | | | |
|-------------------|---------------|----------------------|
| 1. Totolapan | 2. Tlayacapan | 3. Atlatlahucan |
| 4. Tepoztlán | 5. Yautepec | 6. Cuautla |
| 7. Villa de Ayala | 8. Yecapixtla | 9. Tetela del Volcán |

Para construir la gráfica G , la cual es una representación fiel a la distribución geográfica de los municipios, ponemos los vértices $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, en representación de cada uno de los municipios, aristas en color **negro** son caminos de asfalto y las aristas en color **azul** representan conexiones directas en globo aerostático, teniendo un total de 24 aristas. La gráfica G es la siguiente:

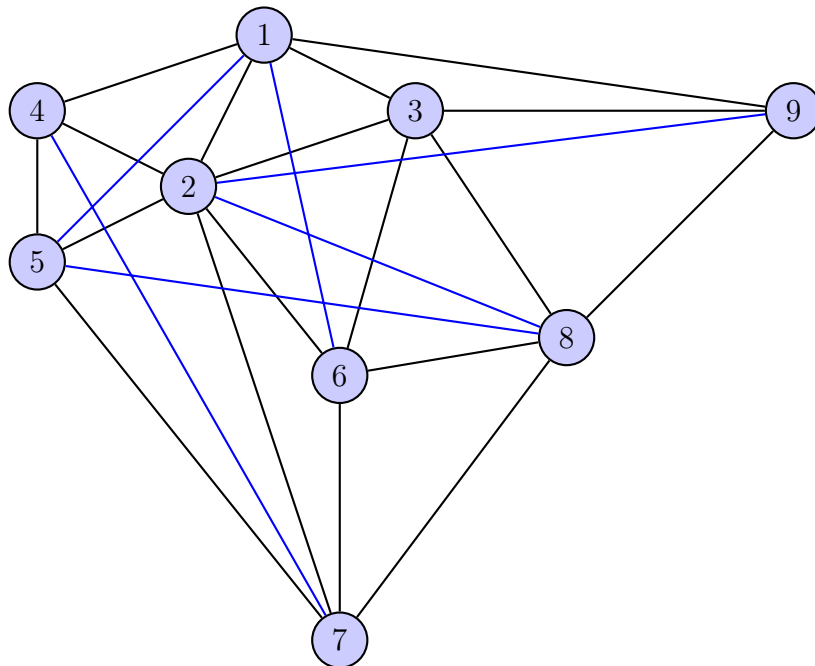


FIGURA 2.4

Solución 2.3.4. Se tienen 9 municipios y 24 caminos, es decir, tenemos 9 vértices y 24

aristas. Cada ruta turística debe de tener una ruta que la conecte con cada municipio, considerando que cada ruta turística debe tomar un camino distinto. Necesitamos dividir a la gráfica G en el máximo número de árboles de expansión ajenos por aristas ($\sigma(G)$).

Por la observación 2.1.5 tomamos $|P| = |V(G)| = 9$, es decir, $P = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}\}$, entonces $A_P(G) = A(G) \implies |A_P(G)| = 24$. Por lo tanto, como

$$r = |P| = 9 \quad |A_P(G)| = 24$$

$$k \leq \frac{|A_P(G)|}{(r-1)} = \frac{24}{(9-1)} = \frac{24}{8} = 3$$

Entonces el entero que se aproxima mejor al valor de k es 3, lo que implica que $\sigma(G) = 3$. Por lo que a lo más se tendrán 3 rutas, tales que cualesquiera dos de ellas no comparten caminos.

La siguiente es una asignación correcta para cada una de las tres posibles rutas turísticas, las cuales pueden ser recorridas parcial o totalmente por una o más unidades:

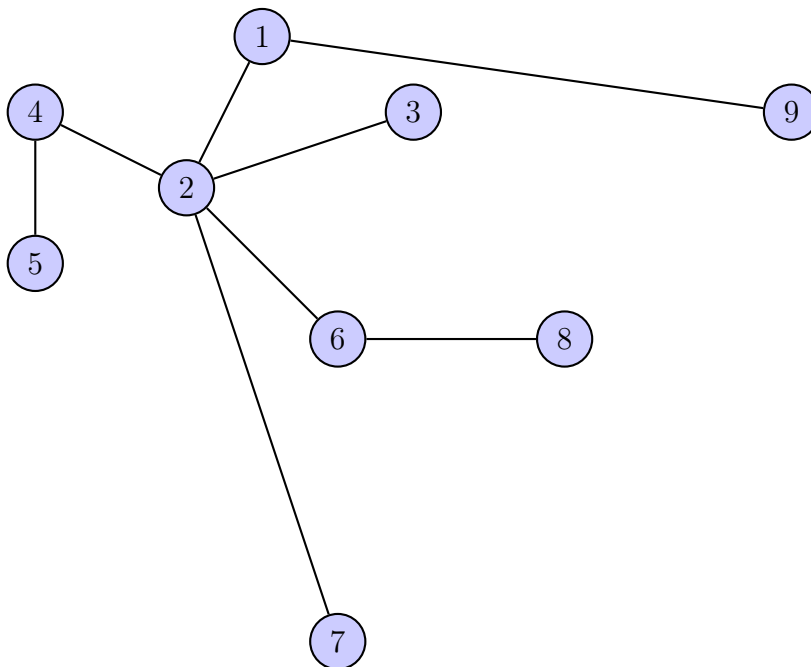


FIGURA 2.5: Ruta 1

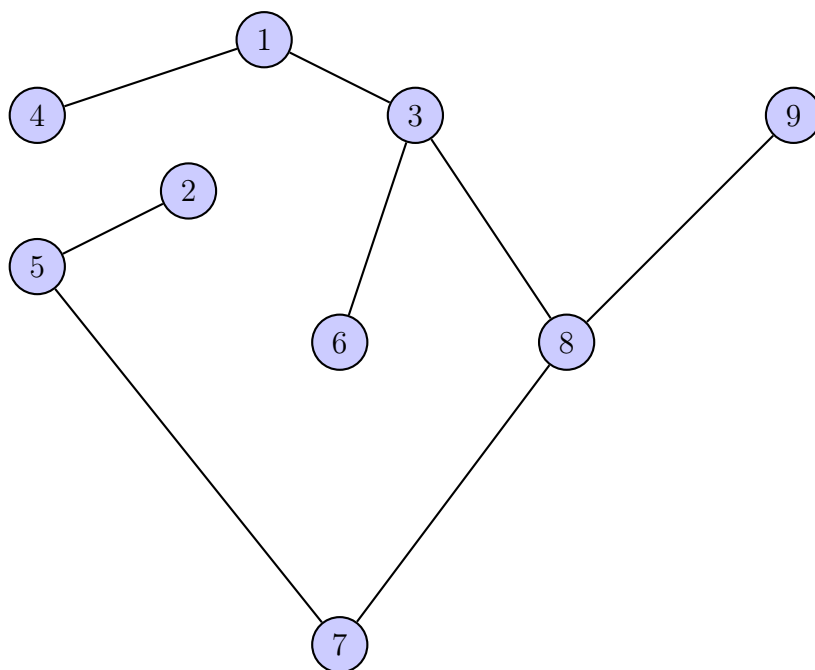


FIGURA 2.6: Ruta 2

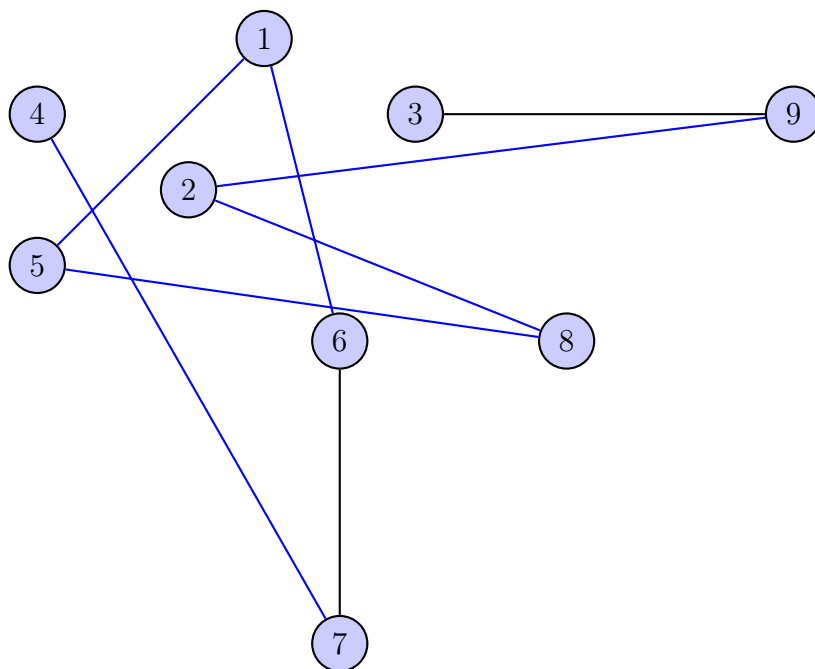


FIGURA 2.7: Ruta 3

Teniendo así tres rutas con la característica de que en la **Ruta 3** el recorrido se hace mayoritariamente en globo aerostático.

Capítulo 3

Ubicación de servicios de emergencia

3.1. Conceptos

Definición 3.1.1. [6] Sean G una gráfica y v un vértice de G , entonces *la excentricidad* del vértice v es la distancia máxima desde v hasta cualquier vértice de la gráfica. Se denota como $e(v) = \max\{d(v, u) : u \in V_G\}$.

Ejemplo 3.1.1. Sea G la gráfica siguiente:

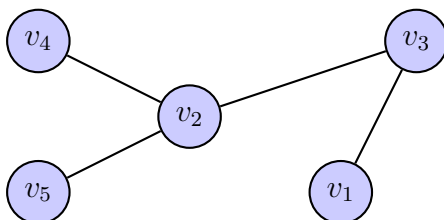


FIGURA 3.1

La *excentricidad* del vértice v_1 ,

$$\begin{aligned} e(v_1) &= \max\{d(v_1, v_2), d(v_1, v_3), d(v_1, v_4), d(v_1, v_5)\} \\ &= \max\{2, 1, 3, 3\} = 3 \end{aligned}$$

Definición 3.1.2. [6] El *diámetro* de G es la excentricidad máxima entre los vértices de G , por lo tanto $\text{diámetro}(G) = \max\{e(v) : v \in V_G\}$ y lo denotaremos como $\text{diam}(G)$.

Ejemplo 3.1.2. Sea G la gráfica siguiente:

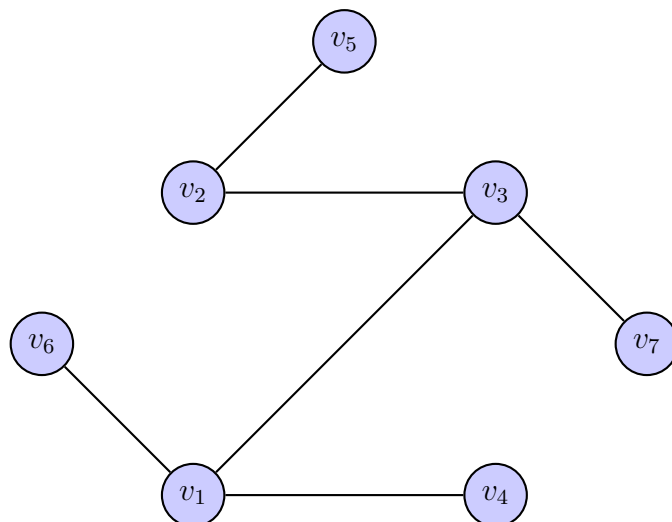


FIGURA 3.2

El *diámetro* de G ,

$$\begin{aligned} \text{diam}(G) &= \text{máx}\{e(v_1), e(v_2), e(v_3), e(v_4), e(v_5), e(v_6), e(v_7)\} \\ &= \text{máx}\{3, 3, 2, 3, 4, 4, 3\} = 4 \end{aligned}$$

Definición 3.1.3. [6] El *radio* de G es la excentricidad mínima entre los vértices de G , por lo tanto $\text{radio}(G) = \min\{e(v) : v \in V_G\}$ y lo denotaremos como $\text{rad}(G)$.

Ejemplo 3.1.3. Retomando el ejemplo 3.1.2, el *radio* de G ,

$$\begin{aligned} \text{rad}(G) &= \text{mín}\{e(v_1), e(v_2), e(v_3), e(v_4), e(v_5), e(v_6), e(v_7)\} \\ &= \text{mín}\{3, 3, 2, 3, 4, 4, 3\} = 2 \end{aligned}$$

Definición 3.1.4. [6] Sea G una gráfica y $v \in V_G$, si $e(v) = \text{rad}(G)$, entonces v es un *punto central* de G . Con esto podemos definir al *centro* como el conjunto de vértices de excentricidad igual al radio (puntos centrales), y lo denotamos como $\text{cen}(G) = \{v \in V_G : e(v) = \text{rad}(G)\}$.

Ejemplo 3.1.4. Del ejemplo 3.1.2,

$$\begin{aligned} \text{rad}(G) &= 2, \quad e(v_1) = 3, \quad e(v_2) = 3, \quad e(v_3) = 2, \\ e(v_4) &= 3, \quad e(v_5) = 4, \quad e(v_6) = 4, \quad e(v_7) = 3 \\ \implies \text{cen}(G) &= \{v \in V_G : e(v) = \text{rad}(G)\} = \{v_3\} \end{aligned}$$

Ejemplo 3.1.5. Sea G la gráfica siguiente:

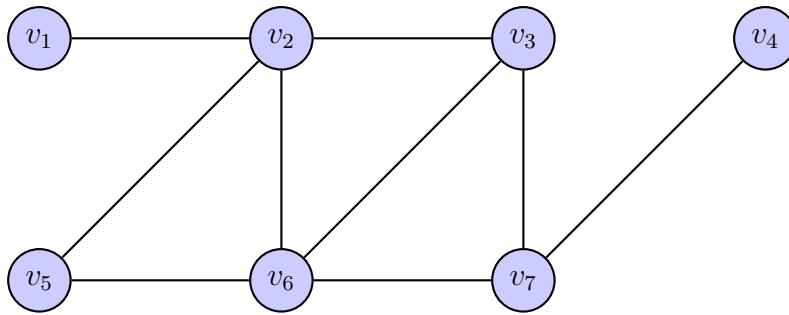


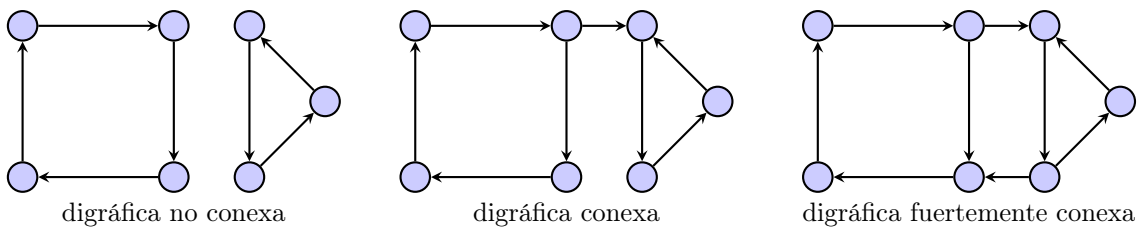
FIGURA 3.3

$$\begin{aligned}
 rad(G) &= \min\{e(v_1), e(v_2), e(v_3), e(v_4), e(v_5), e(v_6), e(v_7)\} \\
 &= \min\{4, 3, 2, 4, 3, 2, 3\} = 2 \\
 &\quad e(v_2) = 2 \quad \text{y} \quad e(v_6) = 2 \\
 \implies \quad cen(G) &= \{v_2, v_6\}
 \end{aligned}$$

Definición 3.1.5. Sea $G = (V_G, F_G)$ una digráfica, una *gráfica subyacente* H de G , es una gráfica tal que $V_H = V_D$ y entre cualquier par de vértices $u, v \in V_G$, si la digráfica D tiene una flecha (u, v) o una flecha (v, u) , la gráfica H no dirigida subyacente incluye una arista $\{v, w\}$.

Definición 3.1.6. [1] Una digráfica G es *conexa* si su gráfica subyacente lo es, de lo contrario es *no conexa*. Una digráfica $D = (V, F)$ es *fuertemente conexa* si para cada $u, v \in V$ existe un uv -camino dirigido y un vu -camino dirigido.

Ejemplo 3.1.6.



3.2. Planteamiento del problema

A medida que la población de una ciudad crece surgen las primeras necesidades, salud y la seguridad. La necesidad de un hospital o centro de salud es imperativa, así como de estaciones de policía o bomberos, dados los costos de construcción y mantenimiento, se prevé que se encuentren en el lugar más accesible desde cualquier punto.

Encontrar el centro, mediante el uso de la teoría de gráficas, es beneficioso para los problemas de ubicación de centros de salud o estaciones de policía o bomberos, donde el objetivo es eliminar la distancia en el peor de los casos.

3.3. Modelación matemática del problema

Problema 3.3.1. En el estado de Jalisco, uno de los principales estados en producción pecuaria, dada la creciente preocupación de los ganaderos locales por las posibles enfermedades, principalmente en el ganado bovino, además de las posibles repercusiones que se generarían a partir de la pérdida del ganado, no solo de los ganaderos, sino del gobierno estatal.

La Secretaría de Agricultura y Desarrollo Rural (SADER) y el gobierno del estado de Jalisco, planean la construcción de un hospital veterinario y un centro de detección y control de enfermedades pecuarias. Considerando que la ubicación del hospital debe ser tal que el personal pueda llegar de manera oportuna a cada centro de producción pecuaria.

Se busca la ubicación del hospital veterinario, de manera que la distancia desde y hacia cualquiera de los centros de producción pecuario de la región sea la menor. Entonces el problema se reduce a encontrar los puntos centrales de la gráfica asociada al problema.

Este problema se modelará utilizando la teoría de gráficas de la siguiente manera:

Sea G la gráfica, tal que dibujemos un vértice por cada una de los 12 centros de producción pecuaria y aristas en representación de conexiones directas. La gráfica es la siguiente:

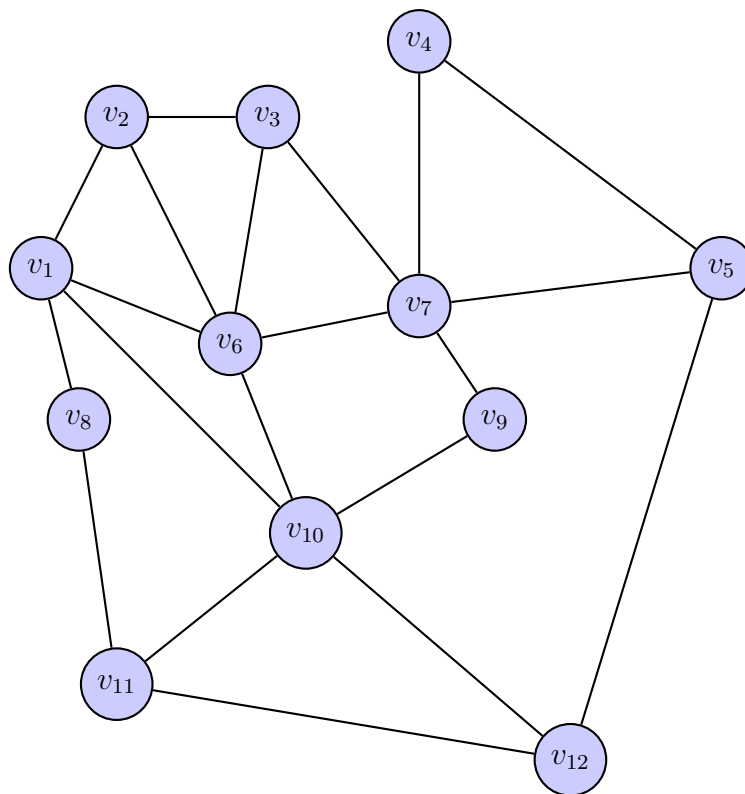


FIGURA 3.4

Solución 3.3.2. Se tienen 12 centros de producción pecuaria y 25 caminos, es decir, 12 vértices y 25 aristas. Considerando que solo se construirá un hospital veterinario, éste deberá estar en un vértice central. Necesitamos encontrar el centro de la gráfica G . Calculamos las excentricidades de cada vértice de G (figura 3.4)

- $$e(v_1) = \max\{d(v_1, v_2), d(v_1, v_3), d(v_1, v_4), d(v_1, v_5), d(v_1, v_6), d(v_1, v_7), d(v_1, v_8),$$

$$d(v_1, v_9), d(v_1, v_{10}), d(v_1, v_{11}), d(v_1, v_{12})\}$$

$$= \max\{1, 2, 3, 3, 1, 2, 1, 2, 1, 2, 2\} = 3$$

$\implies e(v_1) = 3$

- $e(v_2) = \text{máx}\{d(v_2, v_1), d(v_2, v_3), d(v_2, v_4), d(v_2, v_5), d(v_2, v_6), d(v_2, v_7), d(v_2, v_8),$
 $d(v_2, v_9), d(v_2, v_{10}), d(v_2, v_{11}), d(v_2, v_{12})\}$
 $= \text{máx}\{1, 1, 3, 3, 1, 2, 2, 3, 2, 3, 3\} = 3$

$$\implies e(v_2) = 3$$

- $e(v_3) = \text{máx}\{d(v_3, v_1), d(v_3, v_2), d(v_3, v_4), d(v_3, v_5), d(v_3, v_6), d(v_3, v_7), d(v_3, v_8),$
 $d(v_3, v_9), d(v_3, v_{10}), d(v_3, v_{11}), d(v_3, v_{12})\}$
 $= \text{máx}\{2, 1, 2, 2, 1, 1, 3, 2, 2, 3, 3\} = 3$

$$\implies e(v_3) = 3$$

- $e(v_4) = \text{máx}\{d(v_4, v_1), d(v_4, v_2), d(v_4, v_3), d(v_4, v_5), d(v_4, v_6), d(v_4, v_7), d(v_4, v_8),$
 $d(v_4, v_9), d(v_4, v_{10}), d(v_4, v_{11}), d(v_4, v_{12})\}$
 $= \text{máx}\{3, 3, 2, 1, 2, 1, 4, 2, 3, 3, 2\} = 4$

$$\implies e(v_4) = 4$$

- $e(v_5) = \text{máx}\{d(v_5, v_1), d(v_5, v_2), d(v_5, v_3), d(v_5, v_4), d(v_5, v_6), d(v_5, v_7), d(v_5, v_8),$
 $d(v_5, v_9), d(v_5, v_{10}), d(v_5, v_{11}), d(v_5, v_{12})\}$
 $= \text{máx}\{3, 3, 2, 1, 2, 1, 3, 2, 2, 2, 2\} = 3$

$$\implies e(v_5) = 3$$

- $e(v_6) = \text{máx}\{d(v_6, v_1), d(v_6, v_2), d(v_6, v_3), d(v_6, v_4), d(v_6, v_5), d(v_6, v_7), d(v_6, v_8),$
 $d(v_6, v_9), d(v_6, v_{10}), d(v_6, v_{11}), d(v_6, v_{12})\}$
 $= \text{máx}\{1, 1, 1, 2, 2, 1, 2, 2, 1, 2, 2\} = 2$

$$\implies e(v_6) = 2$$

- $e(v_7) = \text{máx}\{d(v_7, v_1), d(v_7, v_2), d(v_7, v_3), d(v_7, v_4), d(v_7, v_5), d(v_7, v_6), d(v_7, v_8),$
 $d(v_7, v_9), d(v_7, v_{10}), d(v_7, v_{11}), d(v_7, v_{12})\}$
 $= \text{máx}\{2, 2, 1, 1, 1, 1, 3, 1, 2, 3, 2\} = 3$

$$\implies e(v_7) = 3$$

- $e(v_8) = \text{máx}\{d(v_8, v_1), d(v_8, v_2), d(v_8, v_3), d(v_8, v_4), d(v_8, v_5), d(v_8, v_6), d(v_8, v_7),$
 $d(v_8, v_9), d(v_8, v_{10}), d(v_8, v_{11}), d(v_8, v_{12})\}$
 $= \text{máx}\{1, 2, 3, 4, 3, 2, 3, 3, 2, 1, 2\} = 4$

$$\implies e(v_8) = 3$$

$$\begin{aligned}
\bullet \quad e(v_9) &= \text{máx}\{d(v_9, v_1), d(v_9, v_2), d(v_9, v_3), d(v_9, v_4), d(v_9, v_5), d(v_9, v_6), d(v_9, v_7), \\
&\quad d(v_9, v_8), d(v_9, v_{10}), d(v_9, v_{11}), d(v_9, v_{12})\} \\
&= \text{máx}\{3, 3, 2, 2, 2, 2, 1, 3, 1, 2, 2\} = 3
\end{aligned}$$

$$\implies e(v_9) = 3$$

$$\begin{aligned}
\bullet \quad e(v_{10}) &= \text{máx}\{d(v_{10}, v_1), d(v_{10}, v_2), d(v_{10}, v_3), d(v_{10}, v_4), d(v_{10}, v_5), d(v_{10}, v_6), d(v_{10}, v_7), \\
&\quad d(v_{10}, v_8), d(v_{10}, v_9), d(v_{10}, v_{11}), d(v_{10}, v_{12})\} \\
&= \text{máx}\{1, 2, 2, 3, 2, 1, 2, 2, 1, 1, 1\} = 3
\end{aligned}$$

$$\implies e(v_{10}) = 3$$

$$\begin{aligned}
\bullet \quad e(v_{11}) &= \text{máx}\{d(v_{11}, v_1), d(v_{11}, v_2), d(v_{11}, v_3), d(v_{11}, v_4), d(v_{11}, v_5), d(v_{11}, v_6), d(v_{11}, v_7), \\
&\quad d(v_{11}, v_8), d(v_{11}, v_9), d(v_{11}, v_{10}), d(v_{11}, v_{12})\} \\
&= \text{máx}\{2, 3, 3, 3, 2, 2, 3, 1, 2, 1, 1\} = 3
\end{aligned}$$

$$\implies e(v_{11}) = 3$$

$$\begin{aligned}
\bullet \quad e(v_{12}) &= \text{máx}\{d(v_{12}, v_1), d(v_{12}, v_2), d(v_{12}, v_3), d(v_{12}, v_4), d(v_{12}, v_5), d(v_{12}, v_6), d(v_{12}, v_7), \\
&\quad d(v_{12}, v_8), d(v_{12}, v_9), d(v_{12}, v_{10}), d(v_{12}, v_{11})\} \\
&= \text{máx}\{2, 3, 3, 2, 1, 2, 2, 2, 2, 1, 1\} = 3
\end{aligned}$$

$$\implies e(v_{12}) = 3$$

El radio de G ($rad(G)$):

$$\begin{aligned}
rad(G) &= \text{mín}\{e(v_1), e(v_2), e(v_3), e(v_4), e(v_5), e(v_6), e(v_7), e(v_8), e(v_9), e(v_{10}), e(v_{11}), e(v_{12})\} \\
&= \text{mín}\{3, 3, 3, 4, 3, 2, 3, 4, 3, 3, 3, 3\} = 2
\end{aligned}$$

Ahora el centro de G , ($cen(G)$):

$$\begin{aligned}
e(v_6) &= 2 \quad \text{y} \quad rad(G) = 2 \\
\implies cen(G) &= \{v \in V_G : e(v) = rad(G)\} = \{v_6\}
\end{aligned}$$

Por lo tanto, el centro de la gráfica G es $\{v_6\}$. Por lo que el lugar óptimo para establecer el hospital veterinario, así como el centro de detección y control de enfermedades pecuarias, es el vértice v_6 .

La siguiente gráfica es la representación de la ubicación deseada, v_6 de color rojo:

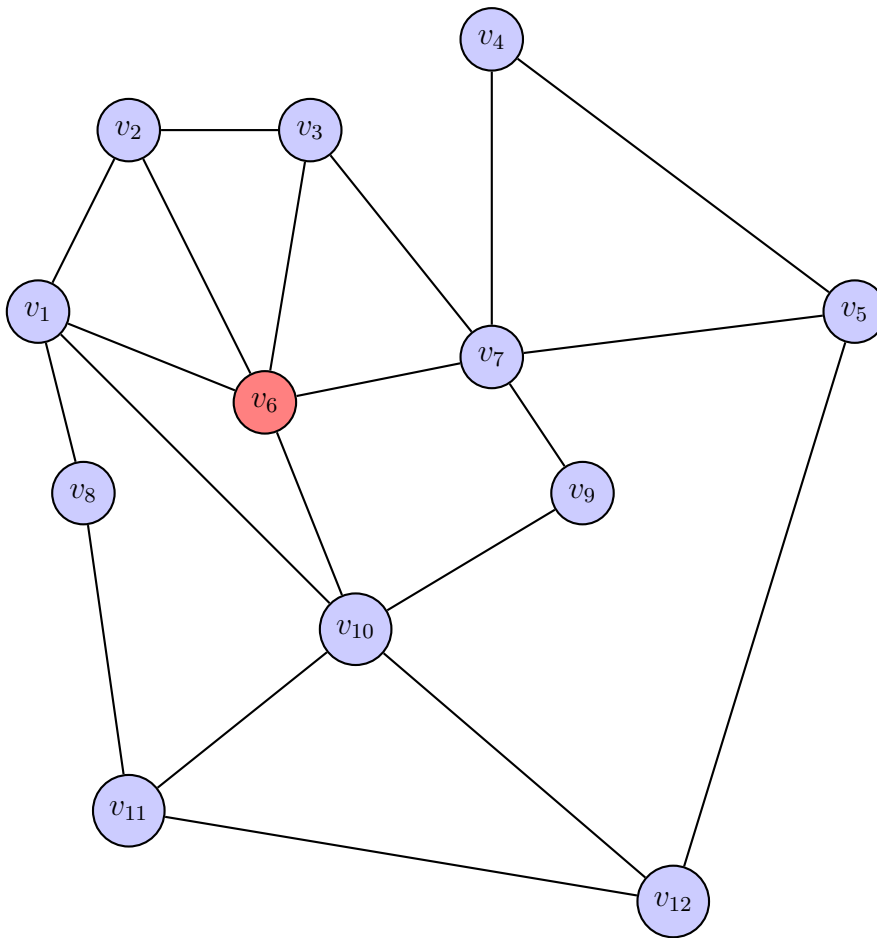


FIGURA 3.5

sólo los que estén en la carretera federal.

Solución 3.3.4. Se tienen un total de 18 intersecciones y 23 caminos, es decir, 18 vértices y 23 aristas. La ubicación del centro de salud deberá estar en un vértice central. Necesitamos encontrar el centro de la gráfica G ($cen(G)$).

Calculamos las excentricidades de cada vértice:

- $e(v_1) = \text{máx}\{d(v_1, v_2), d(v_1, v_3), d(v_1, v_4), d(v_1, v_5), d(v_1, v_6), d(v_1, v_7), d(v_1, v_8),$
 $d(v_1, v_9), d(v_1, v_{10}), d(v_1, v_{11}), d(v_1, v_{12}), d(v_1, v_{13}), d(v_1, v_{14}),$
 $d(v_1, v_{15}), d(v_1, v_{16}), d(v_1, v_{17}), d(v_1, v_{18})\}$
 $= \text{máx}\{2, 7, 1, 2, 6, 3, 2, 3, 4, 5, 3, 4, 5, 4, 6, 6\} = 7$

$$\implies e(v_1) = 7$$

- $e(v_2) = \text{máx}\{d(v_2, v_1), d(v_2, v_3), d(v_2, v_4), d(v_2, v_5), d(v_2, v_6), d(v_2, v_7), d(v_2, v_8),$
 $d(v_2, v_9), d(v_2, v_{10}), d(v_2, v_{11}), d(v_2, v_{12}), d(v_2, v_{13}), d(v_2, v_{14}),$
 $d(v_2, v_{15}), d(v_2, v_{16}), d(v_2, v_{17}), d(v_2, v_{18})\}$
 $= \text{máx}\{2, 7, 1, 2, 6, 3, 2, 3, 4, 5, 3, 4, 5, 4, 6, 6\} = 7$

$$\implies e(v_2) = 7$$

- $e(v_3) = \text{máx}\{d(v_3, v_1), d(v_3, v_2), d(v_3, v_4), d(v_3, v_5), d(v_3, v_6), d(v_3, v_7), d(v_3, v_8),$
 $d(v_3, v_9), d(v_3, v_{10}), d(v_3, v_{11}), d(v_3, v_{12}), d(v_3, v_{13}), d(v_3, v_{14}),$
 $d(v_3, v_{15}), d(v_3, v_{16}), d(v_3, v_{17}), d(v_3, v_{18})\}$
 $= \text{máx}\{7, 7, 6, 5, 1, 4, 5, 2, 4, 3, 3, 5, 4, 4, 6, 5, 5\} = 7$

$$\implies e(v_3) = 7$$

- $e(v_4) = \text{máx}\{d(v_4, v_1), d(v_4, v_2), d(v_4, v_3), d(v_4, v_5), d(v_4, v_6), d(v_4, v_7), d(v_4, v_8),$
 $d(v_4, v_9), d(v_4, v_{10}), d(v_4, v_{11}), d(v_4, v_{12}), d(v_4, v_{13}), d(v_4, v_{14}),$
 $d(v_4, v_{15}), d(v_4, v_{16}), d(v_4, v_{17}), d(v_4, v_{18})\}$
 $= \text{máx}\{1, 1, 6, 1, 5, 2, 1, 4, 2, 3, 4, 2, 3, 4, 3, 5, 5\} = 6$

$$\implies e(v_4) = 6$$

- $e(v_5) = \text{máx}\{d(v_5, v_1), d(v_5, v_2), d(v_5, v_3), d(v_5, v_4), d(v_5, v_6), d(v_5, v_7), d(v_5, v_8),$
 $d(v_5, v_9), d(v_5, v_{10}), d(v_5, v_{11}), d(v_5, v_{12}), d(v_5, v_{13}), d(v_5, v_{14}),$
 $d(v_5, v_{15}), d(v_5, v_{16}), d(v_5, v_{17}), d(v_5, v_{18})\}$

$$= \text{máx}\{2, 2, 5, 1, 4, 1, 2, 3, 1, 2, 3, 2, 3, 3, 4, 4, 4\} = 5$$

$$\implies e(v_5) = 5$$

- $e(v_6) = \text{máx}\{d(v_6, v_1), d(v_6, v_2), d(v_6, v_3), d(v_6, v_4), d(v_6, v_5), d(v_6, v_7), d(v_6, v_8),$
 $d(v_6, v_9), d(v_6, v_{10}), d(v_6, v_{11}), d(v_6, v_{12}), d(v_6, v_{13}), d(v_6, v_{14}),$
 $d(v_6, v_{15}), d(v_6, v_{16}), d(v_6, v_{17}), d(v_6, v_{18})\}$
 $= \text{máx}\{6, 6, 1, 5, 4, 3, 4, 1, 3, 2, 2, 4, 3, 3, 5, 4, 4\} = 6$

$$\implies e(v_6) = 6$$

- $e(v_7) = \text{máx}\{d(v_7, v_1), d(v_7, v_2), d(v_7, v_3), d(v_7, v_4), d(v_7, v_5), d(v_7, v_6), d(v_7, v_8),$
 $d(v_7, v_9), d(v_7, v_{10}), d(v_7, v_{11}), d(v_7, v_{12}), d(v_7, v_{13}), d(v_7, v_{14}),$
 $d(v_7, v_{15}), d(v_7, v_{16}), d(v_7, v_{17}), d(v_7, v_{18})\}$
 $= \text{máx}\{3, 3, 4, 2, 1, 3, 2, 2, 1, 1, 2, 2, 2, 2, 3, 3, 3\} = 4$

$$\implies e(v_7) = 4$$

- $e(v_8) = \text{máx}\{d(v_8, v_1), d(v_8, v_2), d(v_8, v_3), d(v_8, v_4), d(v_8, v_5), d(v_8, v_6), d(v_8, v_7),$
 $d(v_8, v_9), d(v_8, v_{10}), d(v_8, v_{11}), d(v_8, v_{12}), d(v_8, v_{13}), d(v_8, v_{14}),$
 $d(v_8, v_{15}), d(v_8, v_{16}), d(v_8, v_{17}), d(v_8, v_{18})\}$
 $= \text{máx}\{2, 2, 5, 1, 2, 4, 2, 3, 1, 2, 3, 1, 2, 3, 2, 4, 4\} = 5$

$$\implies e(v_8) = 5$$

- $e(v_9) = \text{máx}\{d(v_9, v_1), d(v_9, v_2), d(v_9, v_3), d(v_9, v_4), d(v_9, v_5), d(v_9, v_6), d(v_9, v_7),$
 $d(v_9, v_8), d(v_9, v_{10}), d(v_9, v_{11}), d(v_9, v_{12}), d(v_9, v_{13}), d(v_9, v_{14}),$
 $d(v_9, v_{15}), d(v_9, v_{16}), d(v_9, v_{17}), d(v_9, v_{18})\}$
 $= \text{máx}\{5, 5, 2, 4, 3, 1, 2, 3, 2, 1, 1, 3, 2, 2, 4, 3, 3\} = 5$

$$\implies e(v_9) = 5$$

- $e(v_{10}) = \text{máx}\{d(v_{10}, v_1), d(v_{10}, v_2), d(v_{10}, v_3), d(v_{10}, v_4), d(v_{10}, v_5), d(v_{10}, v_6), d(v_{10}, v_7),$
 $d(v_{10}, v_8), d(v_{10}, v_9), d(v_{10}, v_{11}), d(v_{10}, v_{12}), d(v_{10}, v_{13}), d(v_{10}, v_{14}),$
 $d(v_{10}, v_{15}), d(v_{10}, v_{16}), d(v_{10}, v_{17}), d(v_{10}, v_{18})\}$
 $= \text{máx}\{3, 3, 4, 2, 1, 3, 1, 1, 2, 1, 2, 1, 2, 2, 2, 3, 3\} = 4$

$$\implies e(v_{10}) = 4$$

- $e(v_{11}) = \text{máx}\{d(v_{11}, v_1), d(v_{11}, v_2), d(v_{11}, v_3), d(v_{11}, v_4), d(v_{11}, v_5), d(v_{11}, v_6), d(v_{11}, v_7),$
 $d(v_{11}, v_8), d(v_{11}, v_9), d(v_{11}, v_{10}), d(v_{11}, v_{12}), d(v_{11}, v_{13}), d(v_{11}, v_{14}),$

$$\begin{aligned} & d(v_{11}, v_{15}), d(v_{11}, v_{16}), d(v_{11}, v_{17}), d(v_{11}, v_{18})\} \\ & = \text{máx}\{4, 4, 3, 3, 2, 2, 1, 2, 1, 1, 1, 2, 1, 1, 3, 2, 2\} = 4 \end{aligned}$$

$$\implies e(v_{11}) = 4$$

- $e(v_{12}) = \text{máx}\{d(v_{12}, v_1), d(v_{12}, v_2), d(v_{12}, v_3), d(v_{12}, v_4), d(v_{12}, v_5), d(v_{12}, v_6), d(v_{12}, v_7),$
 $d(v_{12}, v_8), d(v_{12}, v_9), d(v_{12}, v_{10}), d(v_{12}, v_{11}), d(v_{12}, v_{13}), d(v_{12}, v_{14}),$
 $d(v_{12}, v_{15}), d(v_{12}, v_{16}), d(v_{12}, v_{17}), d(v_{12}, v_{18})\}$
 $= \text{máx}\{5, 5, 3, 4, 3, 2, 2, 3, 1, 2, 1, 3, 2, 2, 4, 3, 3\} = 5$

$$\implies e(v_{12}) = 5$$

- $e(v_{13}) = \text{máx}\{d(v_{13}, v_1), d(v_{13}, v_2), d(v_{13}, v_3), d(v_{13}, v_4), d(v_{13}, v_5), d(v_{13}, v_6), d(v_{13}, v_7),$
 $d(v_{13}, v_8), d(v_{13}, v_9), d(v_{13}, v_{10}), d(v_{13}, v_{11}), d(v_{13}, v_{12}), d(v_{13}, v_{14}),$
 $d(v_{13}, v_{15}), d(v_{13}, v_{16}), d(v_{13}, v_{17}), d(v_{13}, v_{18})\}$
 $= \text{máx}\{3, 3, 5, 2, 2, 4, 2, 1, 3, 1, 2, 3, 1, 3, 1, 4, 4\} = 5$

$$\implies e(v_{13}) = 5$$

- $e(v_{14}) = \text{máx}\{d(v_{14}, v_1), d(v_{14}, v_2), d(v_{14}, v_3), d(v_{14}, v_4), d(v_{14}, v_5), d(v_{14}, v_6), d(v_{14}, v_7),$
 $d(v_{14}, v_8), d(v_{14}, v_9), d(v_{14}, v_{10}), d(v_{14}, v_{11}), d(v_{14}, v_{12}), d(v_{14}, v_{13}),$
 $d(v_{14}, v_{15}), d(v_{14}, v_{16}), d(v_{14}, v_{17}), d(v_{14}, v_{18})\}$
 $= \text{máx}\{4, 4, 4, 3, 3, 3, 2, 2, 2, 2, 1, 2, 1, 2, 2, 3, 3\} = 4$

$$\implies e(v_{14}) = 4$$

- $e(v_{15}) = \text{máx}\{d(v_{15}, v_1), d(v_{15}, v_2), d(v_{15}, v_3), d(v_{15}, v_4), d(v_{15}, v_5), d(v_{15}, v_6), d(v_{15}, v_7),$
 $d(v_{15}, v_8), d(v_{15}, v_9), d(v_{15}, v_{10}), d(v_{15}, v_{11}), d(v_{15}, v_{12}), d(v_{15}, v_{13}),$
 $d(v_{15}, v_{14}), d(v_{15}, v_{16}), d(v_{15}, v_{17}), d(v_{15}, v_{18})\}$
 $= \text{máx}\{5, 5, 4, 4, 3, 3, 2, 3, 2, 2, 1, 2, 3, 2, 4, 1, 1\} = 5$

$$\implies e(v_{15}) = 5$$

- $e(v_{16}) = \text{máx}\{d(v_{16}, v_1), d(v_{16}, v_2), d(v_{16}, v_3), d(v_{16}, v_4), d(v_{16}, v_5), d(v_{16}, v_6), d(v_{16}, v_7),$
 $d(v_{16}, v_8), d(v_{16}, v_9), d(v_{16}, v_{10}), d(v_{16}, v_{11}), d(v_{16}, v_{12}), d(v_{16}, v_{13}),$
 $d(v_{16}, v_{14}), d(v_{16}, v_{15}), d(v_{16}, v_{17}), d(v_{16}, v_{18})\}$
 $= \text{máx}\{4, 4, 6, 3, 3, 5, 3, 2, 4, 2, 3, 4, 1, 2, 5, 5\} = 6$

$$\implies e(v_{16}) = 6$$

- $e(v_{17}) = \text{máx}\{d(v_{17}, v_1), d(v_{17}, v_2), d(v_{17}, v_3), d(v_{17}, v_4), d(v_{17}, v_5), d(v_{17}, v_6), d(v_{17}, v_7),$

$$\begin{aligned}
& d(v_{17}, v_8), d(v_{17}, v_9), d(v_{17}, v_{10}), d(v_{17}, v_{11}), d(v_{17}, v_{12}), d(v_{17}, v_{13}), \\
& d(v_{17}, v_{14}), d(v_{17}, v_{15}), d(v_{17}, v_{16}), d(v_{17}, v_{18})\} \\
& = \text{máx}\{6, 6, 5, 5, 4, 4, 3, 4, 3, 3, 2, 3, 4, 3, 4, 2\} = 6
\end{aligned}$$

$$\implies e(v_{17}) = 6$$

- $e(v_{18}) = \text{máx}\{d(v_{18}, v_1), d(v_{18}, v_2), d(v_{18}, v_3), d(v_{18}, v_4), d(v_{18}, v_5), d(v_{18}, v_6), d(v_{18}, v_7),$
 $d(v_{18}, v_8), d(v_{18}, v_9), d(v_{18}, v_{10}), d(v_{18}, v_{11}), d(v_{18}, v_{12}), d(v_{18}, v_{13}),$
 $d(v_{18}, v_{14}), d(v_{18}, v_{15}), d(v_{18}, v_{16}), d(v_{18}, v_{17})\}$
 $= \text{máx}\{6, 6, 5, 5, 4, 4, 3, 4, 3, 3, 2, 3, 4, 3, 4, 2\} = 6$

$$\implies e(v_{18}) = 6$$

El radio de G ($rad(G)$):

$$\begin{aligned}
rad(G) &= \text{mín}\{e(v_1), e(v_2), e(v_3), e(v_4), e(v_5), e(v_6), e(v_7), e(v_8), e(v_9), e(v_{10}), e(v_{11}), e(v_{12}), \\
& e(v_{13}), e(v_{14}), e(v_{15}), e(v_{16}), e(v_{17}), e(v_{18})\} \\
&= \text{mín}\{7, 7, 7, 6, 5, 6, 4, 5, 5, 4, 4, 5, 5, 4, 5, 6, 6, 6\} = 4
\end{aligned}$$

Ahora el centro de G , ($cen(G)$):

$$\begin{aligned}
& e(v_7) = e(v_{10}) = e(v_{11}) = e(v_{14}) = 4 \text{ y } rad(G) = 4 \\
& \implies cen(G) = \{v \in V_G : e(v) = rad(G)\} = \{v_7, v_{10}, v_{11}, v_{14}\}
\end{aligned}$$

Por lo tanto, el centro de la gráfica G es $\{v_7, v_{10}, v_{11}, v_{14}\}$. Por lo que el lugar óptimo para la construcción del centro de salud, considerando que esté cerca de la carretera federal, es en el vértice v_{10} o v_{11} .

Las siguientes gráficas son la representación de las dos opciones de la ubicación del centro de salud, v_{10} de color rojo y v_{11} de color verde:

Opción 1:

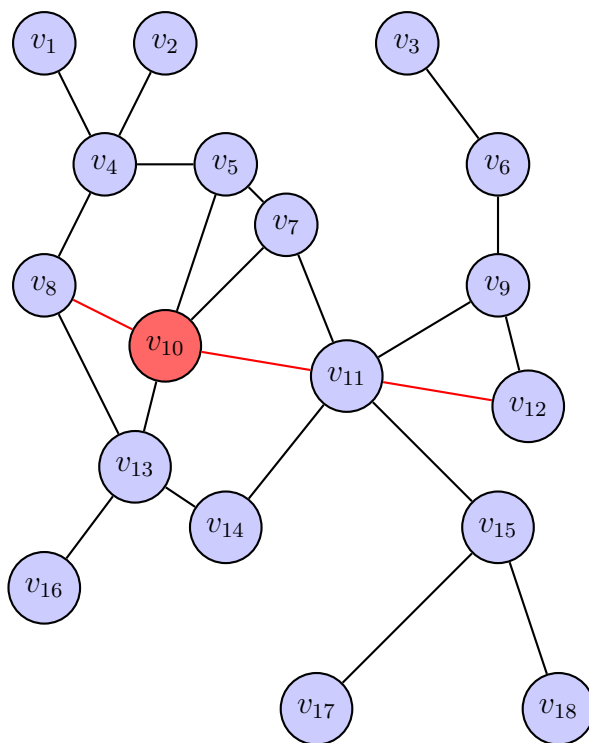


FIGURA 3.6

Opción 2:

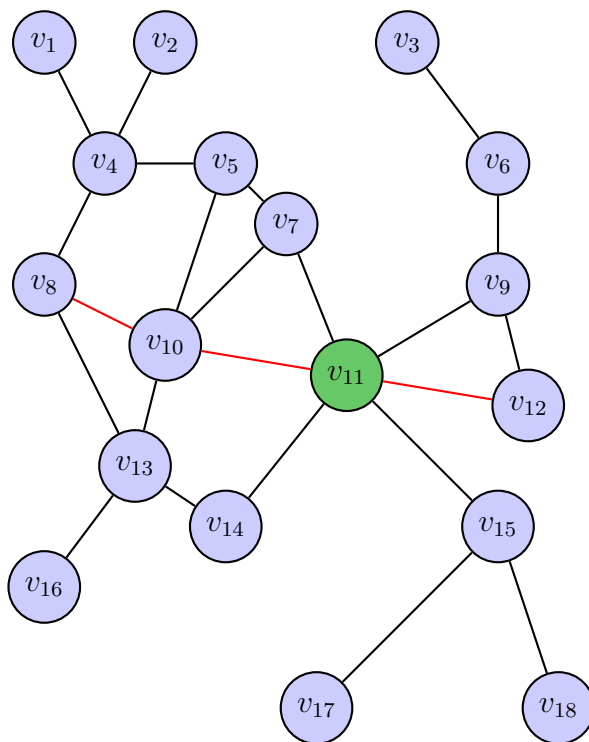


FIGURA 3.7

3.4. Planteamiento del problema

Encontrar el centro, mediante el uso de la teoría de gráficas, es beneficioso para los problemas de ubicación de servicios de emergencia, donde el objetivo es eliminar la distancia en el peor de los casos.

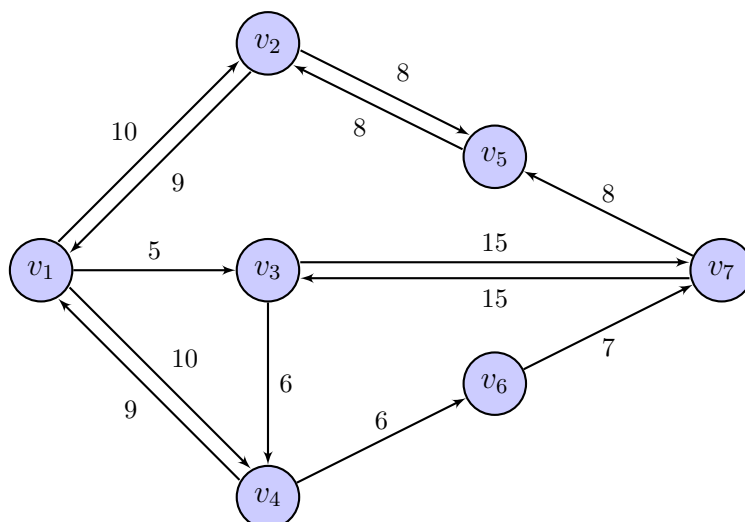
Sin embargo se tomarán en cuenta casos no tan particulares (como los vistos en esta sección), tal es el caso en el que se considere tiempo, distancia o ciertas condiciones que el terreno pueda tener; abriendo así el paso a problemas más apegados a la realidad.

Se modelarán casos pequeños con un desarrollo a detalle para lograr así un buen entendimiento del problemas, así como también se modelarán casos más complejos, que para su solución se plantearán algoritmos para resolverlos de manera más rápida.

3.5. Modelación matemática del problema

Problema 3.5.1. Se desea establecer un centro veterinario, en los principales ranchos ovinocultores en el municipio de Atlacomulco, Estado de México, dadas las recientes preocupaciones acerca de la calidad de la carne y las posibles nuevas enfermedades ovinas, han decidido, de la mano con el gobierno estatal, construir un centro veterinario. Para ello se han dado a la tarea de decidir que rancho albergará las instalaciones de éste centro, tal que desde cualquier otro rancho se tenga una mínima distancia.

Este problema se modelará utilizando la teoría de gráficas de la siguiente manera: sea G la multidigráfica tal que se dibuja un vértice por cada rancho, flechas en representación de conexiones directas entre algunos pares de ranchos, y pesos en las flechas en representación de las distancias entre cada conexión directa, en razón 10 kilómetros igual a 1. La multidigráfica es la siguiente:



El problema se reduce a encontrar la distancia entre cada par de puntos, y los puntos centrales de la multidigráfica.

Solución 3.5.2. Se tienen un total de 7 ranchos y 13 caminos, si consideramos ambas direcciones de cada conexión directa. Para encontrar las distancias entre cada par de puntos de la multidigráfica usaremos el algoritmo de Floyd-Warshall. Recordando la definición 1.3.20, hacemos la matriz de pesos de G , la cual será W_0 , la matriz inicial:

$$W_0 = \begin{array}{c|cccccccc} & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 \\ \hline v_1 & 0 & 10 & 5 & 10 & - & - & - \\ v_2 & 9 & 0 & - & - & 8 & - & - \\ v_3 & - & - & 0 & 6 & - & - & 15 \\ v_4 & 9 & - & - & 0 & - & 6 & - \\ v_5 & - & 8 & - & - & 0 & - & - \\ v_6 & - & - & - & - & - & 0 & 7 \\ v_7 & - & - & 15 & - & 8 & - & 0 \end{array}$$

Las matrices W_0 y S_0 :

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	0	10	5	10	-	-	-
v_2	9	0	-	-	8	-	-
v_3	-	-	0	6	-	-	15
v_4	9	-	-	0	-	6	-
v_5	-	8	-	-	0	-	-
v_6	-	-	-	-	-	0	7
v_7	-	-	15	-	8	-	0

W_0

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	1	2	3	4	5	6	7
v_2	1	2	3	4	5	6	7
v_3	1	2	3	4	5	6	7
v_4	1	2	3	4	5	6	7
v_5	1	2	3	4	5	6	7
v_6	1	2	3	4	5	6	7
v_7	1	2	3	4	5	6	7

S_0

Iteración 1: Las matrices W_1 y S_1 :

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	0	10	5	10	-	-	-
v_2	9	0	14	19	8	-	-
v_3	-	-	0	6	-	-	15
v_4	9	19	14	0	-	6	-
v_5	-	8	-	-	0	-	-
v_6	-	-	-	-	-	0	7
v_7	-	-	15	-	8	-	0

W_1

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	1	2	3	4	5	6	7
v_2	1	2	1	1	5	6	7
v_3	1	2	3	4	5	6	7
v_4	1	1	1	4	5	6	7
v_5	1	2	3	4	5	6	7
v_6	1	2	3	4	5	6	7
v_7	1	2	3	4	5	6	7

S_1

Iteración 2: Las matrices W_2 y S_2 :

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	0	10	5	10	18	-	-
v_2	9	0	14	19	8	-	-
v_3	-	-	0	6	-	-	15
v_4	9	19	14	0	27	6	-
v_5	17	8	22	27	0	-	-
v_6	-	-	-	-	-	0	7
v_7	-	-	15	-	8	-	0

W_2

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	1	2	3	4	2	6	7
v_2	1	2	1	1	5	6	7
v_3	1	2	3	4	5	6	7
v_4	1	1	1	4	2	6	7
v_5	2	2	2	2	5	6	7
v_6	1	2	3	4	5	6	7
v_7	1	2	3	4	5	6	7

S_2

Iteración 3: Las matrices W_3 y S_3 :

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	0	10	5	10	18	-	20
v_2	9	0	14	19	8	-	29
v_3	-	-	0	6	-	-	15
v_4	9	19	14	0	27	6	29
v_5	17	8	22	27	0	-	37
v_6	-	-	-	-	-	0	7
v_7	-	-	15	21	8	-	0

W_3

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	1	2	3	4	2	6	3
v_2	1	2	1	1	5	6	3
v_3	1	2	3	4	5	6	7
v_4	1	1	1	4	2	6	3
v_5	2	2	2	2	5	6	3
v_6	1	2	3	4	5	6	7
v_7	1	2	3	3	5	6	7

S_3

Iteración 4: Las matrices W_4 y S_4 :

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	0	10	5	10	18	15	20
v_2	9	0	14	19	8	25	29
v_3	16	25	0	6	33	12	15
v_4	9	19	14	0	27	6	29
v_5	17	8	22	27	0	33	37
v_6	-	-	-	-	-	0	7
v_7	30	40	15	21	8	27	0

W_4

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	1	2	3	4	2	4	3
v_2	1	2	1	1	5	4	3
v_3	4	4	3	4	4	4	7
v_4	1	1	1	4	2	6	3
v_5	2	2	2	2	5	4	3
v_6	1	2	3	4	5	6	7
v_7	4	4	3	3	5	4	7

S_4

Iteración 5: Las matrices W_5 y S_5 :

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	0	10	5	10	18	16	20
v_2	9	0	14	19	8	25	29
v_3	15	25	0	6	33	12	15
v_4	9	19	14	0	27	6	29
v_5	17	8	22	27	0	33	37
v_6	-	-	-	-	-	0	7
v_7	25	16	15	21	8	27	0

W_5

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	1	2	3	4	2	4	3
v_2	1	2	1	1	5	4	3
v_3	4	4	3	4	4	4	7
v_4	1	1	1	4	2	6	3
v_5	2	2	2	2	5	4	3
v_6	1	2	3	4	5	6	7
v_7	5	5	3	3	5	4	7

S_5

Iteración 6: Las matrices W_6 y S_6 :

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	0	10	5	10	18	16	20
v_2	9	0	14	19	8	25	29
v_3	15	25	0	6	33	12	15
v_4	9	19	14	0	27	6	13
v_5	17	8	22	27	0	33	37
v_6	-	-	-	-	-	0	7
v_7	25	16	15	21	8	27	0

W_6

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	1	2	3	4	2	4	3
v_2	1	2	1	1	5	4	3
v_3	4	4	3	4	4	4	7
v_4	1	1	1	4	2	6	6
v_5	2	2	2	2	5	4	3
v_6	1	2	3	4	5	6	7
v_7	5	5	3	3	5	4	7

S_6

Iteración 7: Las matrices W_7 y S_7 :

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	0	10	5	10	18	16	20
v_2	9	0	14	19	8	25	29
v_3	15	25	0	6	23	12	15
v_4	9	19	14	0	21	6	13
v_5	17	8	22	27	0	33	37
v_6	32	23	22	28	15	0	7
v_7	25	16	15	21	8	27	0

W_7

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	1	2	3	4	2	4	3
v_2	1	2	1	1	5	4	3
v_3	4	4	3	4	7	4	7
v_4	1	1	1	4	7	6	6
v_5	2	2	2	2	5	4	3
v_6	7	7	7	7	7	6	7
v_7	5	5	3	3	5	4	7

S_7

Tenemos como resultado a las matrices finales W_f , la cual es la matriz de distancias D , y S_f la matriz final de sucesores.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	0	10	5	10	18	16	20
v_2	9	0	14	19	8	25	29
v_3	15	25	0	6	23	12	15
v_4	9	19	14	0	21	6	13
v_5	17	8	22	27	0	33	37
v_6	32	23	22	28	15	0	7
v_7	25	16	15	21	8	27	0

W_f

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	1	2	3	4	2	4	3
v_2	1	2	1	1	5	4	3
v_3	4	4	3	4	7	4	7
v_4	1	1	1	4	7	6	6
v_5	2	2	2	2	5	4	3
v_6	7	7	7	7	7	6	7
v_7	5	5	3	3	5	4	7

S_f

Ahora necesitamos encontrar el centro de la multidigráfica.

Calculamos las excentricidades de cada vértice:

- $e(v_1) = \text{máx}\{d(v_1, v_2), d(v_1, v_3), d(v_1, v_4), d(v_1, v_5), d(v_1, v_6), d(v_1, v_7)\}$
 $= \text{máx}\{10, 5, 10, 18, 16, 20\} = 20$
 $\implies e(v_1) = 20$
- $e(v_2) = \text{máx}\{d(v_2, v_1), d(v_2, v_3), d(v_2, v_4), d(v_2, v_5), d(v_2, v_6), d(v_2, v_7)\}$
 $= \text{máx}\{9, 14, 19, 8, 25, 29\} = 29$
 $\implies e(v_2) = 29$
- $e(v_3) = \text{máx}\{d(v_3, v_1), d(v_3, v_2), d(v_3, v_4), d(v_3, v_5), d(v_3, v_6), d(v_3, v_7)\}$
 $= \text{máx}\{15, 25, 6, 23, 12, 15\} = 25$
 $\implies e(v_3) = 25$
- $e(v_4) = \text{máx}\{d(v_4, v_1), d(v_4, v_2), d(v_4, v_3), d(v_4, v_5), d(v_4, v_6), d(v_4, v_7)\}$
 $= \text{máx}\{9, 19, 14, 21, 6, 13\} = 21$
 $\implies e(v_4) = 21$
- $e(v_5) = \text{máx}\{d(v_5, v_1), d(v_5, v_2), d(v_5, v_3), d(v_5, v_4), d(v_5, v_6), d(v_5, v_7)\}$
 $= \text{máx}\{17, 8, 22, 27, 33, 37\} = 37$
 $\implies e(v_5) = 37$
- $e(v_6) = \text{máx}\{d(v_6, v_1), d(v_6, v_2), d(v_6, v_3), d(v_6, v_4), d(v_6, v_5), d(v_6, v_7)\}$
 $= \text{máx}\{32, 23, 22, 28, 15, 7\} = 32$
 $\implies e(v_6) = 32$
- $e(v_7) = \text{máx}\{d(v_7, v_1), d(v_7, v_2), d(v_7, v_3), d(v_7, v_4), d(v_7, v_5), d(v_7, v_6)\}$
 $= \text{máx}\{25, 16, 15, 21, 8, 27\} = 27$
 $\implies e(v_7) = 27$

El radio de G , (definición 3.1.3):

$$\begin{aligned} \text{rad}(G) &= \text{mín}\{e(v_1), e(v_2), e(v_3), e(v_4), e(v_5), e(v_6), e(v_7)\} \\ &= \text{mín}\{20, 29, 25, 21, 37, 32, 27\} = 20 \end{aligned}$$

Ahora el centro de G :

$$e(v_1) = 20 \text{ y } rad(G) = 20$$

$$\implies cen(G) = \{v \in V : e(v) = rad(G)\} = \{v_1\}$$

Solución: Por lo tanto, el centro de G es $\{v_1\}$, por lo que el lugar óptimo para la construcción del centro veterinario es en el vértice v_1 .

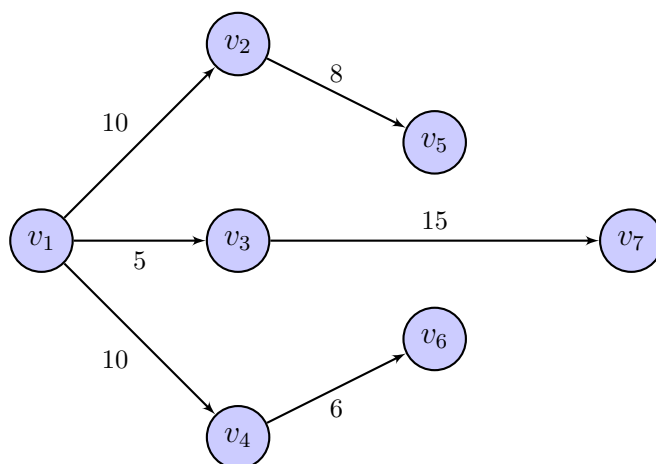
Además veremos la ruta más corta desde el vértice v_1 a los demás vértices. Si nos fijamos en el primer renglón de la matriz W_f , veremos la distancia del vértice v_1 a los otros vértices de G .

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	0	10	5	10	18	16	20

Y también el primer renglón de la matriz S_f , veremos los vértices que forman el camino de peso mínimo.

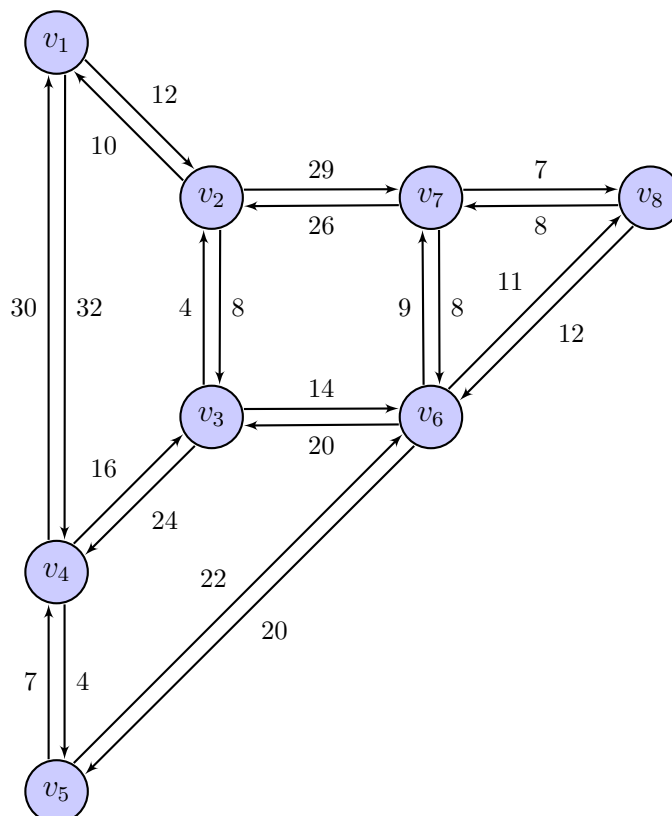
	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	1	2	3	4	2	4	3

Por lo que la arborescencia desde el vértice v_1 es la siguiente:



Problema 3.5.3. En ciudad de Juárez, Chihuahua, derivado de los altos índices de inmigrantes que han llegado, se desea establecer un centro de abastecimiento de recursos para las fuerzas de seguridad. Para ello se han dado a la tarea de decidir en que estación de policía será en la que éste centro será construido, tal que desde cualquier otra estación de policía se tenga una mínima distancia.

Este problema se modelará utilizando la teoría de gráficas de la siguiente manera: sea G la multidigráfica, tal que se dibuja un vértice por cada estación de policía, ya sea municipal, estatal o federal, flechas en representación de conexiones directas entre algunos pares de estaciones, y pesos en las flechas en representación de el tiempo promedio recorrido entre cada conexión directa, el cual es el tiempo en minutos entre cada estación según datos de Google maps. La multidigráfica es la siguiente:



Las estaciones planteadas son las siguientes:

- v_1 : Estación de Policía Distrito Delicias, 32130, C. 16 de Septiembre 247, Bellavista, Cd Juárez, Chih.
- v_2 : Dirección de la Policía Estatal Única, Eje Vial Juan Gabriel 1360, Condesa de Alamos San Lorenzo, 32340 Cd Juárez, Chih.
- v_3 : Secretaría de Seguridad Pública Municipal Distrito Universidad, Pedro N. Gracia 2160 Partido N, Díaz, 32575 Cd Juárez, Chih.
- v_4 : Policía Federal, Nuevo Hipódromo, 32685 Cd Juárez, Chih.

- v_5 : Policía Federal, Aeropuerto Internacional Abraham González, Cd Juárez, Chih.
- v_6 : Estación de Policía Distrito Sur, Av, C. Valle del Cedro 578, Morelos III, 32574 Cd Juárez, Chih.
- v_7 : Policía Municipal Distrito Oriente, Río Amacuzac s/n, Las Arcadas, 32563 Cd Juárez, Chih.
- v_8 : Estación de Policía Distrito Valle, Iturbide 802, María Isabel, 32560 Cd Juárez, Chih.

El problema se reduce a encontrar los puntos centrales de la multidigráfica, los cuales nos permitirán resolver el problema, y además podremos encontrar el tiempo mínimo de una estación a otra, así como la ruta a seguir.

Solución 3.5.4. Se tienen un total de 8 vértices y 22 flechas, si consideramos ambas direcciones de cada conexión directa. Para encontrar las distancias entre cada par de puntos de la multidigráfica, usaremos el algoritmo de Floyd-Warshall. Recordando la Definición 1.3.20 hacemos la matriz de pesos de G , la cual será W_0 , la matriz inicial.

$$W_0 = \begin{array}{c|cccccccc} & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 \\ \hline v_1 & 0 & 12 & - & 32 & - & - & - & - \\ v_2 & 10 & 0 & 8 & - & - & - & 29 & - \\ v_3 & - & 4 & 0 & 24 & - & 14 & - & - \\ v_4 & 30 & - & 16 & 0 & 4 & - & - & - \\ v_5 & - & - & - & 7 & 0 & 22 & - & - \\ v_6 & - & - & 20 & - & 20 & 0 & 9 & 11 \\ v_7 & - & 26 & - & - & - & 8 & 0 & 7 \\ v_8 & - & - & - & - & - & 12 & 8 & 0 \end{array}$$

Las matrices W_0 y S_0 :

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	0	12	-	32	-	-	-	-
v_2	10	0	8	-	-	-	29	-
v_3	-	4	0	24	-	14	-	-
v_4	30	-	16	0	4	-	-	-
v_5	-	-	-	7	0	22	-	-
v_6	-	-	20	-	20	0	9	11
v_7	-	26	-	-	-	8	0	7
v_8	-	-	-	-	-	12	8	0

W_0

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	1	2	3	4	5	6	7	8
v_2	1	2	3	4	5	6	7	8
v_3	1	2	3	4	5	6	7	8
v_4	1	2	3	4	5	6	7	8
v_5	1	2	3	4	5	6	7	8
v_6	1	2	3	4	5	6	7	8
v_7	1	2	3	4	5	6	7	8
v_8	1	2	3	4	5	6	7	8

S_0

Iteración 1: Las matrices W_1 y S_1 :

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	0	12	-	32	-	-	-	-
v_2	10	0	8	42	-	-	29	-
v_3	-	4	0	24	-	14	-	-
v_4	30	42	16	0	4	-	-	-
v_5	-	-	-	7	0	22	-	-
v_6	-	-	20	-	20	0	9	11
v_7	-	26	-	-	-	8	0	7
v_8	-	-	-	-	-	12	8	0

W_1

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	1	2	3	4	5	6	7	8
v_2	1	2	3	1	5	6	7	8
v_3	1	2	3	4	5	6	7	8
v_4	1	1	3	4	5	6	7	8
v_5	1	2	3	4	5	6	7	8
v_6	1	2	3	4	5	6	7	8
v_7	1	2	3	4	5	6	7	8
v_8	1	2	3	4	5	6	7	8

S_1

Iteración 2: Las matrices W_2 y S_2 :

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	0	12	20	32	-	-	41	-
v_2	10	0	8	42	-	-	29	-
v_3	14	4	0	24	-	14	33	-
v_4	30	42	16	0	4	-	71	-
v_5	-	-	-	7	0	22	-	-
v_6	-	-	20	-	20	0	9	11
v_7	36	26	34	68	-	8	0	7
v_8	-	-	-	-	-	12	8	0

W_2

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	1	2	2	4	5	6	2	8
v_2	1	2	3	1	5	6	7	8
v_3	2	2	3	4	5	6	2	8
v_4	1	1	3	4	5	6	2	8
v_5	1	2	3	4	5	6	7	8
v_6	1	2	3	4	5	6	7	8
v_7	2	2	2	2	5	6	7	8
v_8	1	2	3	4	5	6	7	8

S_2

Iteración 3: Las matrices W_3 y S_3 :

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	0	12	20	32	-	34	41	-
v_2	10	0	8	32	-	22	29	-
v_3	14	4	0	24	-	14	33	-
v_4	30	20	16	0	4	30	49	-
v_5	-	-	-	7	0	22	-	-
v_6	34	24	20	44	20	0	9	11
v_7	36	26	34	58	-	8	0	7
v_8	-	-	-	-	-	12	8	0

W_3

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	1	2	2	4	5	3	2	8
v_2	1	2	3	3	5	3	7	8
v_3	2	2	3	4	5	6	2	8
v_4	1	3	3	4	5	3	3	8
v_5	1	2	3	4	5	6	7	8
v_6	3	3	3	3	5	6	7	8
v_7	2	2	2	3	5	6	7	8
v_8	1	2	3	4	5	6	7	8

S_3

Iteración 4: Las matrices W_4 y S_4 :

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	0	12	20	32	36	34	41	-
v_2	10	0	8	32	36	22	29	-
v_3	14	4	0	24	28	14	33	-
v_4	30	20	16	0	4	30	49	-
v_5	37	27	23	7	0	22	56	-
v_6	34	24	20	44	20	0	9	11
v_7	36	26	34	58	62	8	0	7
v_8	-	-	-	-	-	12	8	0

W_4

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	1	2	2	4	4	3	2	8
v_2	1	2	3	3	4	3	7	8
v_3	2	2	3	4	4	6	2	8
v_4	1	3	3	4	5	3	3	8
v_5	4	4	4	4	5	6	4	8
v_6	3	3	3	3	5	6	7	8
v_7	2	2	2	3	4	6	7	8
v_8	1	2	3	4	5	6	7	8

S_4

Iteración 5: Las matrices W_5 y S_5 :

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	0	12	20	32	36	34	41	-
v_2	10	0	8	32	36	22	29	-
v_3	14	4	0	24	28	14	33	-
v_4	30	20	16	0	4	26	49	-
v_5	37	27	23	7	0	22	56	-
v_6	34	24	20	27	20	0	9	11
v_7	36	26	34	58	62	8	0	7
v_8	-	-	-	-	-	12	8	0

W_5

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	1	2	2	4	4	3	2	8
v_2	1	2	3	3	4	3	7	8
v_3	2	2	3	4	4	6	2	8
v_4	1	3	3	4	5	5	3	8
v_5	4	4	4	4	5	6	4	8
v_6	3	3	3	5	5	6	7	8
v_7	2	2	2	3	4	6	7	8
v_8	1	2	3	4	5	6	7	8

S_5

Iteración 6: Las matrices W_6 y S_6 :

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	0	12	20	32	36	34	41	45
v_2	10	0	8	32	36	22	29	33
v_3	14	4	0	24	28	14	23	25
v_4	30	20	16	0	4	26	35	37
v_5	37	27	23	7	0	22	31	33
v_6	34	24	20	27	20	0	9	11
v_7	36	26	28	35	28	8	0	7
v_8	46	36	32	39	32	12	8	0

W_6

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	1	2	2	4	4	3	2	6
v_2	1	2	3	3	4	3	7	6
v_3	2	2	3	4	4	6	6	6
v_4	1	3	3	4	5	5	6	6
v_5	4	4	4	4	5	6	6	6
v_6	3	3	3	5	5	6	7	8
v_7	2	2	6	6	6	6	7	8
v_8	6	6	6	6	6	6	7	8

S_6

Iteración 7: Las matrices W_7 y S_7 :

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	0	12	20	32	36	34	41	45
v_2	10	0	8	32	36	22	29	33
v_3	14	4	0	24	28	14	23	25
v_4	30	20	16	0	4	26	35	37
v_5	37	27	23	7	0	22	31	33
v_6	34	24	20	27	20	0	9	11
v_7	36	26	28	35	28	8	0	7
v_8	44	34	32	39	32	12	8	0

W_7

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	1	2	2	4	4	3	2	6
v_2	1	2	3	3	4	3	7	6
v_3	2	2	3	4	4	6	6	6
v_4	1	3	3	4	5	5	6	6
v_5	4	4	4	4	5	6	6	6
v_6	3	3	3	5	5	6	7	8
v_7	2	2	6	6	6	6	7	8
v_8	7	7	6	6	6	6	7	8

S_7

Iteración 8: Las matrices W_8 y S_8 :

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	0	12	20	32	36	34	41	45
v_2	10	0	8	32	36	22	29	33
v_3	14	4	0	24	28	14	23	25
v_4	30	20	16	0	4	26	35	37
v_5	37	27	23	7	0	22	31	33
v_6	34	24	20	27	20	0	9	11
v_7	36	26	28	35	28	8	0	7
v_8	44	34	32	39	32	12	8	0

W_8

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	1	2	2	4	4	3	2	6
v_2	1	2	3	3	4	3	7	6
v_3	2	2	3	4	4	6	6	6
v_4	1	3	3	4	5	5	6	6
v_5	4	4	4	4	5	6	6	6
v_6	3	3	3	5	5	6	7	8
v_7	2	2	6	6	6	6	7	8
v_8	7	7	6	6	6	6	7	8

S_8

Tenemos como resultado a las matrices finales W_f , la cual es la matriz de distancias, y S_f la matriz final de sucesores.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	0	12	20	32	36	34	41	45
v_2	10	0	8	32	36	22	29	33
v_3	14	4	0	24	28	14	23	25
v_4	30	20	16	0	4	26	35	37
v_5	37	27	23	7	0	22	31	33
v_6	34	24	20	27	20	0	9	11
v_7	36	26	28	35	28	8	0	7
v_8	44	34	32	39	32	12	8	0

W_f

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	1	2	2	4	4	3	2	6
v_2	1	2	3	3	4	3	7	6
v_3	2	2	3	4	4	6	6	6
v_4	1	3	3	4	5	5	6	6
v_5	4	4	4	4	5	6	6	6
v_6	3	3	3	5	5	6	7	8
v_7	2	2	6	6	6	6	7	8
v_8	7	7	6	6	6	6	7	8

S_f

Ahora necesitamos encontrar el centro de la multidigráfica.

Calculamos las excentricidades de cada vértice:

- $e(v_1) = \max\{d(v_1, v_2), d(v_1, v_3), d(v_1, v_4), d(v_1, v_5), d(v_1, v_6), d(v_1, v_7), d(v_1, v_8)\}$
 $= \max\{12, 20, 32, 36, 34, 41, 45\} = 45$

$$\implies e(v_1) = 45$$

- $e(v_2) = \max\{d(v_2, v_1), d(v_2, v_3), d(v_2, v_4), d(v_2, v_5), d(v_2, v_6), d(v_2, v_7), d(v_2, v_8)\}$
 $= \max\{10, 8, 32, 36, 22, 29, 33\} = 36$

$$\implies e(v_2) = 36$$

- $e(v_3) = \max\{d(v_3, v_1), d(v_3, v_2), d(v_3, v_4), d(v_3, v_5), d(v_3, v_6), d(v_3, v_7), d(v_3, v_8)\}$
 $= \max\{14, 4, 24, 28, 14, 23, 25\} = 28$

$$\implies e(v_3) = 28$$

- $e(v_4) = \max\{d(v_4, v_1), d(v_4, v_2), d(v_4, v_3), d(v_4, v_5), d(v_4, v_6), d(v_4, v_7), d(v_4, v_8)\}$
 $= \max\{30, 20, 16, 4, 26, 35, 37\} = 37$

$$\implies e(v_4) = 37$$

- $e(v_5) = \max\{d(v_5, v_1), d(v_5, v_2), d(v_5, v_3), d(v_5, v_4), d(v_5, v_6), d(v_5, v_7), d(v_5, v_8)\}$
 $= \max\{37, 27, 23, 7, 22, 31, 33\} = 37$

$$\implies e(v_5) = 37$$

- $e(v_6) = \max\{d(v_6, v_1), d(v_6, v_2), d(v_6, v_3), d(v_6, v_4), d(v_6, v_5), d(v_6, v_7), d(v_6, v_8)\}$
 $= \max\{34, 24, 20, 27, 20, 9, 11\} = 34$

$$\implies e(v_6) = 45$$

- $e(v_7) = \max\{d(v_7, v_1), d(v_7, v_2), d(v_7, v_3), d(v_7, v_4), d(v_7, v_5), d(v_7, v_6), d(v_7, v_8)\}$
 $= \max\{36, 26, 28, 35, 28, 8, 7\} = 36$

$$\implies e(v_7) = 36$$

- $e(v_8) = \max\{d(v_8, v_1), d(v_8, v_2), d(v_8, v_3), d(v_8, v_4), d(v_8, v_5), d(v_8, v_6), d(v_8, v_7)\}$
 $= \max\{44, 34, 32, 39, 32, 12, 8\} = 44$

$$\implies e(v_8) = 44$$

El radio de G , (definición 3.1.3):

$$\begin{aligned} rad(G) &= \min\{e(v_1), e(v_2), e(v_3), e(v_4), e(v_5), e(v_6), e(v_7), e(v_8)\} \\ &= \min\{45, 36, 28, 37, 37, 34, 36, 44\} = 28 \end{aligned}$$

Ahora el centro de G :

$$e(v_3) = 28 \text{ y } rad(G) = 28$$

$$\implies cen(G) = \{v \in V : e(v) = rad(G)\} = \{v_3\}$$

Solución: Por lo tanto, el centro de G es $\{v_3\}$, por lo que el lugar óptimo para la construcción del centro de abastecimiento de recursos es el vértice v_3 , y el tiempo de respuesta máximo desde éste vértice es de 28 minutos.

Además veremos la ruta más corta desde el vértice v_3 a los demás vértices. Al fijarnos en el tercer renglón de la matriz W_f , veremos el menor tiempo del vértice v_3 a los otros vértices de G .

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_3	14	4	0	24	28	14	23	25

Y también gracias a la matriz S_f , veremos los vértices que forman el camino de peso mínimo (menor tiempo). Por lo que la arborescencia (definición 1.3.12) desde el vértice v_3 es la siguiente:

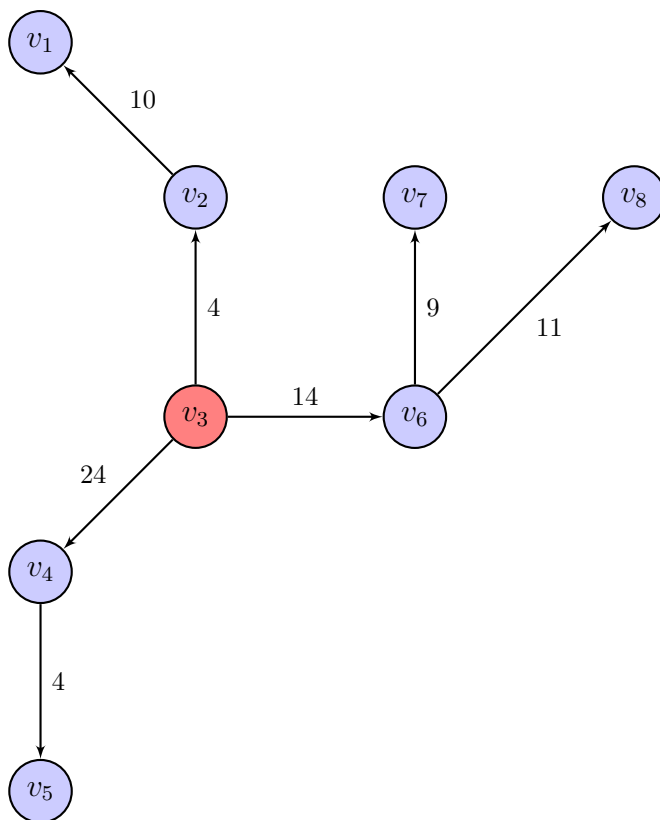


FIGURA 3.8

3.6. Desarrollo del código (Python)

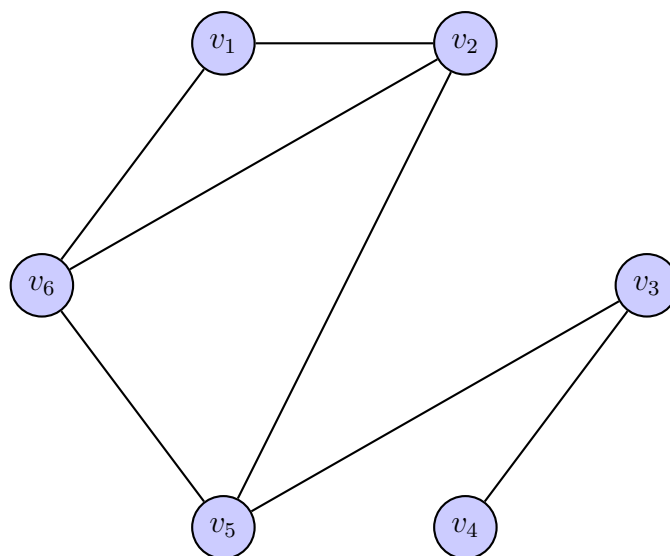
Ya sabemos como funciona el algoritmo de Floyd-Warshall, sin embargo, el proceso incluso para digráficas pequeñas, tal es el caso del Ejemplo 3.4.1., es muy tardado. Es por ello que se desarrollará un código para resolver de forma sencilla el problema inicial planteado.

La plataforma en la que será desarrollado el programa es *Jupyter Notebook*, la cual es una aplicación web para crear documentos computacionales; en ésta encontramos distintas bibliotecas útiles en ciencia de datos.

Para desarrollar el programa usaremos la biblioteca *NetworkX*, la cual sirve para el estudio de gráficas y análisis de redes, misma que nos ofrece diversas herramientas para el desarrollo de problemas en los que tengamos gráficas grandes, por ejemplo, gráficas que representen una colonia, en la que cada intersección de dos o más calles es un vértice, y cada calle es una arista, o una red de abastecimiento de agua en los que cada toma de agua sea un vértice, y las conexiones entre cada toma las aristas.

3.6.1. Construcción de gráficas, digráficas y multidigráficas con pesos

Código 3.6.1. Iniciaremos planteando la siguiente gráfica simple:



Antes de comenzar, añadimos la biblioteca *NetworkX*,

```
[1]: import networkx as nx
```

Establecemos al conjunto de vértices V , y al conjunto de aristas A ,

```
[2]: V= {1,2,3,4,5,6}
```

```
[3]: A=[(1,2), (3,4), (5,6), (2,6), (5,3), (5,2), (1,6)]
```

Ahora definamos a G la gráfica que queremos construir, y añadamos a los vértices y aristas,

```
[4]: G=nx.Graph()
```

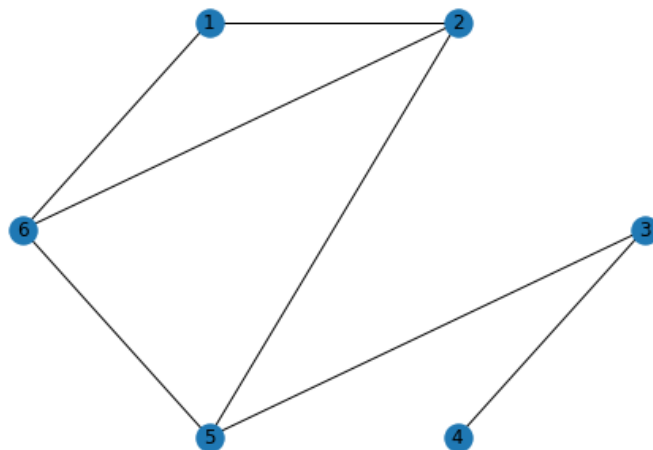
```
[5]: G.add_nodes_from(V)
```

```
[6]: G.add_edges_from(A)
```

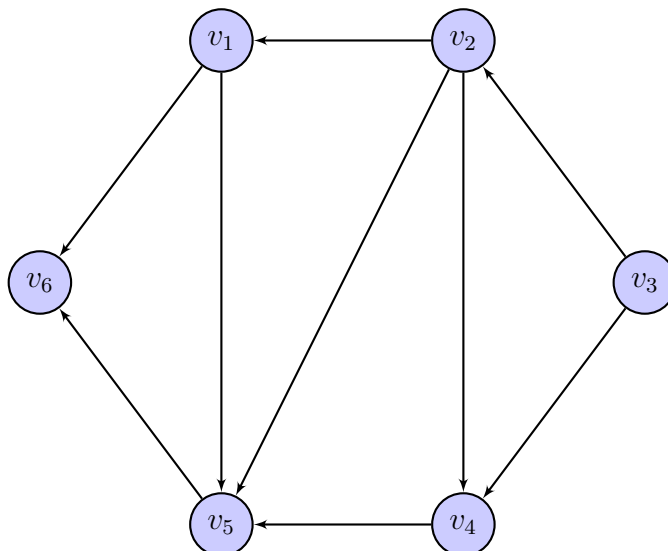
Para ver la gráfica escribimos lo siguiente (el renglón 7 sólo sirve para que los vértices estén fijos, de tal manera que visualmente sea lo mismo que la gráfica planteada, y no es necesario agregarlos para el correcto funcionamiento del programa). El comando `nx.draw` sirve para dibujar la gráfica, se abre un paréntesis y se especifica la gráfica a dibujar, en éste caso G , el comando de `pos` se refiere a la posición de los vértices, y el comando `with_labels` se refiere a si agregar (`True`) o no (`False`) las etiquetas de los vértices.

```
[7]: pos = {1:[0,4], 2:[4,4], 3:[7,0], 4:[4,-4], 5:[0,-4], 6:[-3,0]}
```

```
[8]: nx.draw(G, pos=pos, with_labels=True)
```



Código 3.6.2. Ahora pasamos a la construcción de una gráfica dirigida, partiremos de la siguiente digráfica:



Añadimos la biblioteca de *NetworkX*,

```
[1]: import networkx as nx
```

Establecemos al conjunto de vértices V , y al conjunto de flechas F ,

```
[4]: V= {1,2,3,4,5,6}
```

```
[5]: F=[(2,1), (3,4), (5,6), (1,5), (2,4), (2,5), (3,2), (4,5), (1,6)]
```

Definimos a la digráfica G y añadamos los vértices y flechas,

```
[6]: G=nx.DiGraph()
```

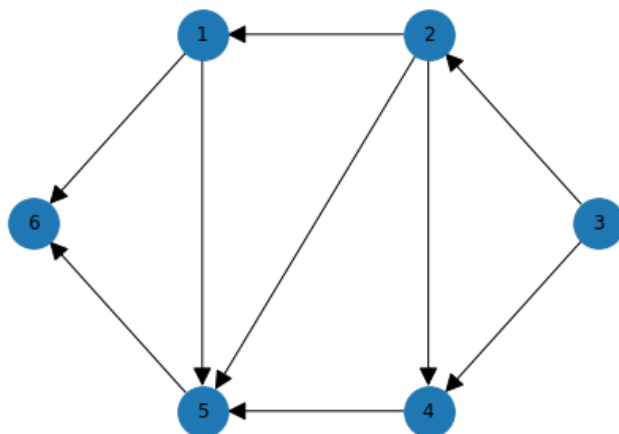
```
[7]: G.add_nodes_from(V)
```

```
[8]: G.add_edges_from(F)
```

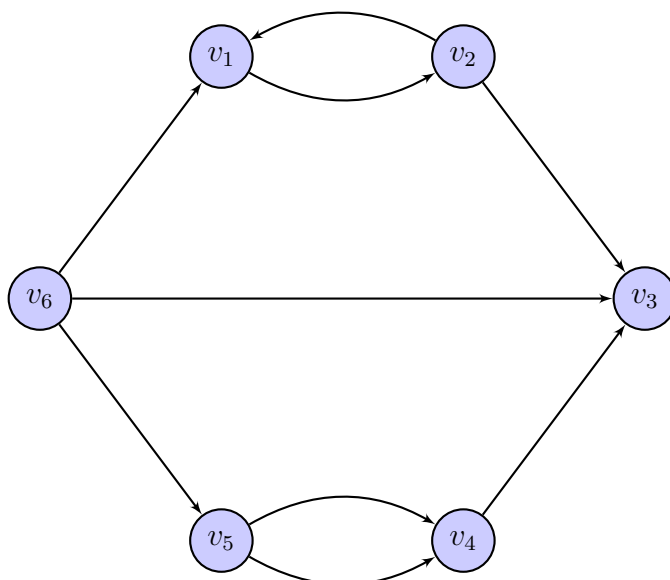
El comando *node_size* sirve para definir el tamaño de los vértices, y el comando *arrowsize* sirve para definir el tamaño de las puntas de las flechas. Para ver la gráfica escribimos lo siguiente

```
[9]: pos = {1:[0,4], 2:[4,4], 3:[7,0], 4:[4,-4], 5:[0,-4], 6:[-3,0]}
```

```
[10]: nx.draw(G, pos=pos, with_labels=True, node_size=1000, arrowsize=25)
```



Código 3.6.3. Ahora pasamos a una multidigráfica, partiendo de la siguiente multidigráfica,



Añadimos la biblioteca *NetworkX*,

```
[1]: import networkx as nx
```

Establecemos al conjunto de vértices V , y al conjunto de flechas F ,

```
[2]: V= {1,2,3,4,5,6}
```

```
[3]: F= [(6,1), (6,5), (6,3), (1,2), (2,1), (5,4), (5,4), (2,3), (4,3)]
```

Definimos a la multidigráfica G y añadamos los vértices y flechas (al añadir las flechas aparecerá un renglón que nos señalará si se repite alguna flecha).

```
[4]: G=nx.MultiDiGraph()
```

```
[5]: G.add_nodes_from(V)
```

```
[6]: G.add_edges_from(F)
```

```
[6]: [0, 0, 0, 0, 0, 0, 1, 0, 0]
```

La cantidad de veces que se repite cada arista se puede ver si escribimos lo siguiente (en donde los primeros dos valores corresponden a la flecha y el tercero a las veces que se repite)

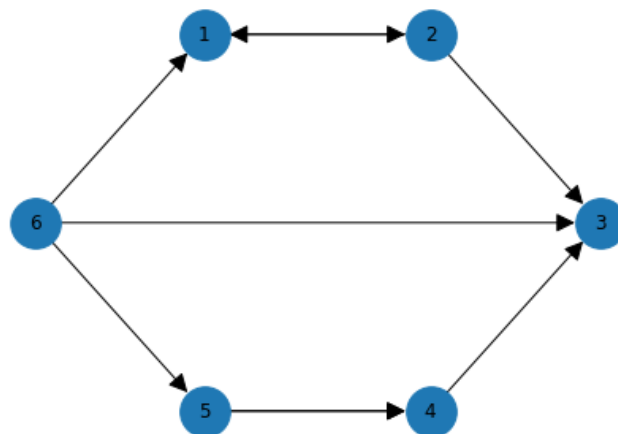
```
[7]: print(G.edges)
```

```
[(1, 2, 0), (2, 1, 0), (2, 3, 0), (4, 3, 0), (5, 4, 0), (5, 4, 1),  
(6, 1, 0), (6, 5, 0), (6, 3, 0)]
```

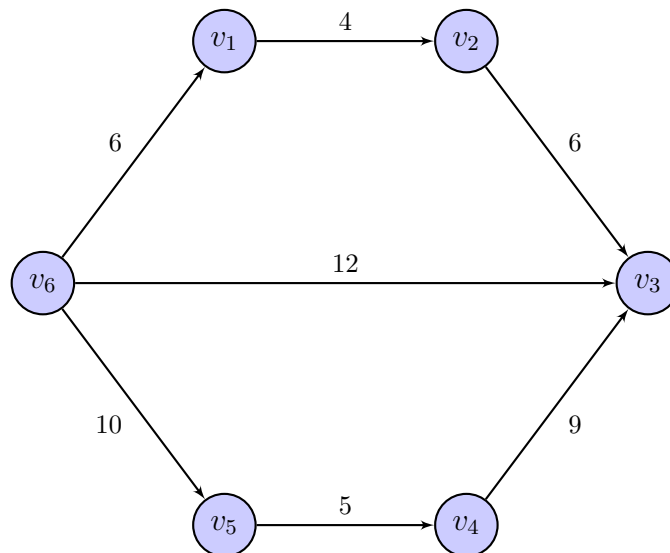
Para ver la gráfica escribimos lo siguiente

```
[8]: pos = {1:[0,4], 2:[4,4], 3:[7,0], 4:[4,-4], 5:[0,-4], 6:[-3,0]}
```

```
[9]: nx.draw(G, pos=pos, with_labels=True, node_size=1000, arrowsize=25)
```



Código 3.6.4. Ahora veremos el código para una digráfica con pesos (mismo que si se tiene una multidigráfica con pesos), partiendo de la siguiente



Añadimos la biblioteca de *NetworkX*,

```
[1]: import networkx as nx
```

Establecemos al conjunto de vértices V , y al conjunto de flechas F ,

```
[2]: V= {1,2,3,4,5,6}
```

```
[3]: F= [(6,1,6), (6,5,10), (6,3,12), (1,2,4), (5,4,5), (2,3,6), (4,3,9)]
```

Definimos a la digráfica G y añadimos los vértices y flechas,

```
[4]: G=nx.DiGraph()
```

```
[5]: G.add_nodes_from(V)
```

```
[6]: G.add_weighted_edges_from(F)
```

Definimos el peso (*weight*) de las flechas,

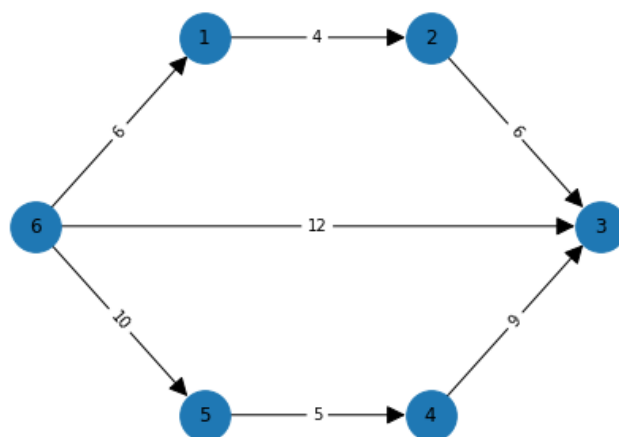
```
[7]: weight=nx.get_edge_attributes(G, 'weight')
```

Para ver la gráfica escribimos lo siguiente, el texto que aparece al escribir el inciso 9, nos menciona los datos de cada flecha

```
[8]: pos={1:[0,4], 2:[4,4], 3:[7,0], 4:[4,-4], 5:[0,-4], 6:[-3,0]}
```

```
[9]: nx.draw(G, pos=pos, with_labels=True, node_size=1000, arrowsize=25)
     nx.draw_networkx_edge_labels(G, pos, edge_labels= weight)
```

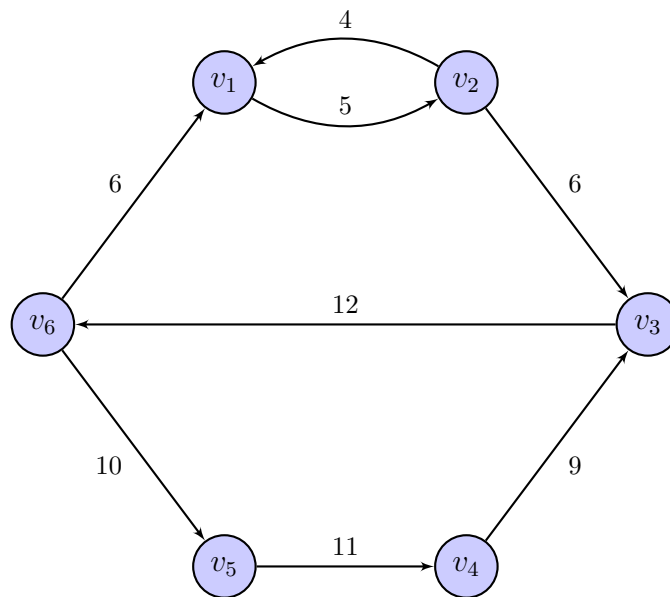
```
[9]: {(1, 2): Text(2.0, 4.0, '4'),
      (2, 3): Text(5.5, 2.0, '6'),
      (4, 3): Text(5.5, -2.0, '9'),
      (5, 4): Text(2.0, -4.0, '5'),
      (6, 1): Text(-1.5, 2.0, '6'),
      (6, 5): Text(-1.5, -2.0, '10'),
      (6, 3): Text(2.0, 0.0, '12')}
```



3.6.2. Puntos centrales de la digráfica

El código para encontrar los puntos centrales de la digráfica presenta una condición particular, la cual es que la digráfica sea fuertemente conexa, tal condición surge del proceso para encontrar la excentricidad de cada vértice, la cual depende de la distancia, la cual a su vez depende de la existencia de un camino dirigido.

Código 3.6.5. Tomando la siguiente multidigráfica con pesos, obtendremos el centro con la ayuda de NetworkX,



Escribimos a la multidigráfica igual que en el código 3.6.4, y quitamos la parte de la representación ya que no es necesaria,

```
[1]: import networkx as nx
```

```
[2]: V={1,2,3,4,5,6}
```

```
[3]: F= [(6,1,6), (6,5,10), (3,6,12), (1,2,5), (2,1,4), (5,4,11),
(2,3,6), (4,3,9)]
```

```
[4]: G=nx.DiGraph()
G.add_nodes_from(V)
G.add_weighted_edges_from(F)
```

```
[5]: weight=nx.get_edge_attributes(G, 'weight')
```

Ahora verifiquemos que la multidigráfica es fuertemente conexa, ésto para corroborar que cada par de vértices están conectados, lo cual también nos asegura que no se tendrán distancias infinitas, en caso de ser fuertemente conexa saldrá *True*, de lo contrario *False*.

```
[6]: nx.is_strongly_connected(G)
```

```
[6]: True
```

Para obtener el centro de la multidigráfica, si se tienen gráficas con peso en las flechas o

aristas se escribe *barycenter*, en el caso contrario se escribe *center*. Para ello el paquete de *NetworkX* corre el algoritmo de Dijkstra para cada vértice, una muestra del cálculo hecho para cada vértice, utilizando el comando *shortest_path_length* (colocando en el espacio *source* el vértice de origen y el método para encontrar el camino de peso mínimo, en éste caso el algoritmo de Dijkstra (1.4.1), es el siguiente:

```
[7]: nx.shortest_path_length(G, source=1, weight='weight',
method='dijkstra')
```

```
[7]: {1: 0, 2: 5, 3: 11, 6: 23, 5: 33, 4: 44}
```

```
[8]: nx.shortest_path_length(G, source=2, weight='weight',
method='dijkstra')
```

```
[8]: {2: 0, 1: 4, 3: 6, 6: 18, 5: 28, 4: 39}
```

```
[9]: nx.shortest_path_length(G, source=3, weight='weight',
method='dijkstra')
```

```
[9]: {3: 0, 6: 12, 1: 18, 5: 22, 2: 23, 4: 33}
```

```
[10]: nx.shortest_path_length(G, source=4, weight='weight',
method='dijkstra')
```

```
[10]: {4: 0, 3: 9, 6: 21, 1: 27, 5: 31, 2: 32}
```

```
[11]: nx.shortest_path_length(G, source=5, weight='weight',
method='dijkstra')
```

```
[11]: {5: 0, 4: 11, 3: 20, 6: 32, 1: 38, 2: 43}
```

```
[12]: nx.shortest_path_length(G, source=6, weight='weight',
method='dijkstra')
```

```
[12]: {6: 0, 1: 6, 5: 10, 2: 11, 3: 17, 4: 21}
```

Podemos apreciar que obtendremos las distancias desde el vértice origen (*source*), hacia cada vértice de la multidigráfica, ordenadas de menor a mayor.

Y por último el comando para obtener el centro

```
[13]: nx.barycenter(G, weight='weight')
```

```
[13]: [6]
```

Y obtendremos el resultado, el cual en éste caso es el vértice 6.

Ejemplo 3.6.1. A continuación se probará el código en un problema previamente hecho, el problema 3.5.3. Escribimos el código, el cual se quitará la parte para la representación gráfica ya que solo buscamos el centro,

```
[1]: import networkx as nx
```

```
[2]: V={1,2,3,4,5,6,7,8}
```

```
[3]: F= [(1,2,12), (2,1,10),  
        (1,4,32), (4,1,30),  
        (2,3,8), (3,2,4),  
        (2,7,29), (7,2,26),  
        (3,4,24), (4,3,16),  
        (3,6,14), (6,3,20),  
        (4,5,4), (5,4,7),  
        (5,6,22), (6,5,20),  
        (6,7,9), (7,6,8),  
        (6,8,11), (8,6,12),  
        (7,8,7), (8,7,8),  
        ]
```

```
[4]: G=nx.DiGraph()  
G.add_nodes_from(V)  
G.add_weighted_edges_from(F)
```

```
[5]: weight=nx.get_edge_attributes(G, 'weight')
```

Se verifica la información de la multidigráfica, así como se verifica que ésta sea fuertemente conexa

```
[6]: print(nx.info(G))
```

Name:

Type: DiGraph

Number of nodes: 8

Number of edges: 22

Average in degree: 2.7500

Average out degree: 2.7500

```
[7]: nx.is_strongly_connected(G)
```

```
[7]: True
```

Y por último obteniendo el mismo resultado que en el problema 3.5.3,

```
[8]: nx.barycenter(G, weight='weight')
```

```
[8]: [3]
```

el cual es el vértice v_3 , además se obtiene en el caso del problema 3.5.3 el tiempo mínimo a recorrer si escribimos lo siguiente,

```
[9]: nx.shortest_path_length(G, source=3, weight='weight',
method='dijkstra')
```

```
[9]: {3: 0, 2: 4, 6: 14, 1: 14, 7: 23, 4: 24, 8: 25, 5: 28}
```

3.6.3. Arborescencia: digráfica de caminos mínimos

Por último buscamos la arborescencia generada desde los puntos centrales de la digráfica, para ello nos guiaremos del Código 3.6.5. Gracias a la arborescencia obtenemos los caminos mínimos desde los puntos centrales de la digráfica

Código 3.6.6. Teniendo en cuenta los datos de la digráfica,

```
[1]: import networkx as nx
```

```
[2]: V={1,2,3,4,5,6}
```

```
[3]: F= [(6,1,6),(6,5,10),(3,6,12),(1,2,5),(2,1,4),(5,4,11),
(2,3,6),(4,3,9)]
```

```
[4]: G=nx.DiGraph()
G.add_nodes_from(V)
G.add_weighted_edges_from(F)
```

```
[5]: weight=nx.get_edge_attributes(G, 'weight')
```

```
[6]: nx.is_strongly_connected(G)
```

```
[6]: True
```

```
[7]: nx.barycenter(G, weight='weight')
```

```
[7]: [6]
```

Ahora escribimos lo siguiente para obtener todos los caminos mínimos desde el vértice 6, así como las longitudes de cada uno de éstos,

```
[8]: p1 = nx.shortest_path(G, source=6, weight="weight")
```

```
[9]: longitud = nx.shortest_path_length(G, source=6, weight="weight")
```

```
[10]: print("Todos los caminos mínimos desde 6:", p1)
print("Longitud de los caminos mínimos: ", longitud)
```

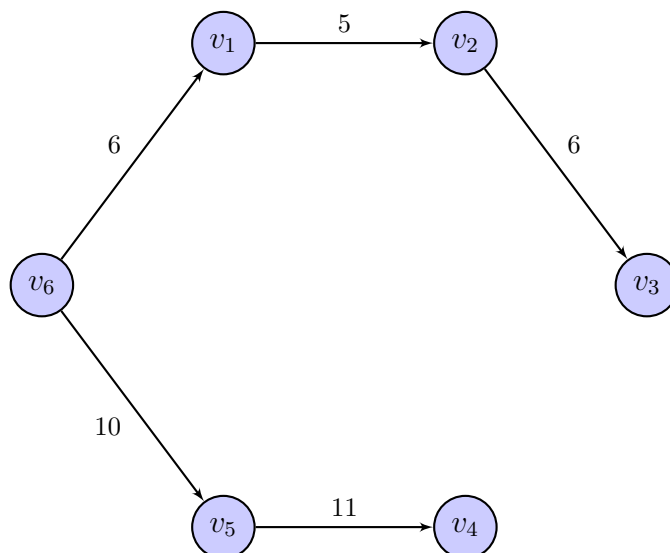
Todos los caminos mínimos desde 6:

```
{6: [6], 1: [6, 1], 5: [6, 5], 2: [6, 1, 2], 4: [6, 5, 4],
3: [6, 1, 2, 3]}
```

Longitud de los caminos mínimos:

```
{6: 0, 1: 6, 5: 10, 2: 11, 3: 17, 4: 21}
```

Obteniendo así la arborescencia, la cual es la siguiente:



Para concluir con éste capítulo se planteará un problema relacionado al gas natural en México, utilizando datos del libro: El Gas Natural, y su geografía industrial en México. [11]

Problema 3.6.1. En 2011 México ocupó el lugar número 11 en el consumo internacional de gas natural; aproximadamente 60% del valor agregado industrial de México se origina a empresas que usan gas natural para sus procesos de transformación productivos.

La creciente demanda de gas natural destinado principalmente a la generación de energía eléctrica y el consumo industrial, ha provocado presiones al SNG (Sistema Nacional de Gasoductos) llevando algunos tramos del mismo a un punto de saturación con capacidades disponibles menores al 10%. En noviembre del 2011 la Secretaría de Energía presentó el estado en el que se encontraba el SNG por regiones de entrega; en el cual el SNG se divide en 5 regiones.

Nos enfocaremos en la región Noreste, ya que ésta es de suma importancia para la industria nacional. Es una región productora e importadora de gas natural desde Estados Unidos, abastece a otras regiones del país; aquí en 2011 se generó el 17,85% del PIB nacional, así como el 20,11% de la inversión total de la industria.

En la región Noreste, los principales municipios consumidores de gas natural para la industria, la comprenden los municipios de Chihuahua, Torreón, Ramos Arizpe, Monclova, Monterrey, San Nicolás de los Garza, Apodaca, Reynosa, Altamira y Ciudad Madero, de los cuales Monterrey es el principal consumidor a nivel nacional y San

Nicolás de los Garza el tercero; éstos 10 municipios aportan el 79,77 % del consumo de la región.

La Secretaría de Energía busca construir un centro de control y monitoreo de los gasoductos en los estados de la región Noreste, los cuales son Chihuahua, Durango, Coahuila de Zaragoza, Nuevo León y Tamaulipas. Además de que se contará con un equipo especializado en la contención de desastres naturales (como daños estructurales producidos por el movimiento sísmico) o provocados, que surjan de la distribución del gas natural, es por ello que se espera que la zona de construcción esté en un punto tal que desde cualquier otro (zona industrial), la distancia de atención sea la mínima posible.

Los datos de las zonas industriales a considerar son los siguientes, ordenadas por estado, municipio y localidad, así como el consumo 2008 – 2011 de GN (gas natural) en MMpcd (millones de pies cúbicos diarios) :

Estado	Municipio	Localidad	Consumo de GN
Chihuahua	Ahumada	Miguel Ahumada	0,006
Chihuahua	Aldama	Juan Aldama	1,000
Chihuahua	Allende	Valle Ignacio de Allende	0,010
Chihuahua	Camargo	Santa Rosalía de Camargo	0,261
Chihuahua	Cuahutémoc	Cuahutémoc	7,222
Chihuahua	Chihuahua	Chihuahua	14,831
Chihuahua	Delicias	Delicias	1,763
Chihuahua	Jiménez	José Mariano Jiménez	0,220
Chihuahua	Juárez	Juárez	6,911
Chihuahua	Rosales	Santa Cruz Rosales	0,014
Chihuahua	Saucillo	Saucillo	0,234
Chihuahua	Santa Isabel	Santa Isabel	0,005
Coahuila	Castaños	Castaños	0,106
Coahuila	Francisco I. Madero	Francisco I. Madero	0,054
Coahuila	General Cepeda	General Cepeda	0,009

Coahuila	Matamoros	San Antonio del Coyote	0,428
Coahuila	Monclova	Monclova	23,431
Coahuila	Parras	Parras	2,655
Coahuila	Piedras Negras	Piedras Negras	0,933
Coahuila	Ramos Arizpe	Ramos Arizpe	12,130
Coahuila	San Pedro	San Pedro	0,451
Coahuila	Sierra Mojada	Laguna del Rey	0,439
Coahuila	Torreón	Torreón	34,161
Durango	Cuencamé	Cuencamé	0,044
Durango	Durango	Durango	4,109
Durango	Gómez Palacio	Gómez Palacio	8,771
Durango	Lerdo	Lerdo	0,690
Durango	Mapimí	Ceballos	0,086
Durango	Peñon Blanco	Peñon Blanco	0,007
Durango	Tlahualilo	Tlahualilo de Zaragoza	0,015
Nuevo León	Apodaca	Apodaca	8,878
Nuevo León	Caderyta de Jiménez	Caderyta de Jiménez	8,553
Nuevo León	China	China	0,068
Nuevo León	San Pedro Garza García	San Pedro Garza García	6,638
Nuevo León	General Bravo	General Bravo	0,025
Nuevo León	General Escobedo	Cd. General Escobedo	2,222
Nuevo León	Mina	Mina	0,020
Nuevo León	Monterrey	Monterrey	91,428
Nuevo León	Pesquería	Pesquería	2,410
Nuevo León	San Nicolás de los Garza	San Nicolás de los Garza	60,166
Nuevo León	Hidalgo	Hidalgo	0,126
Nuevo León	Santa Catarina	Santa Catarina	7,832
Tamaulipas	Abasolo	Abasolo	0,019
Tamaulipas	Aldama	Aldama	0,041
Tamaulipas	Altamira	Altamira	29,658
Tamaulipas	Ciudad Madero	Ciudad Madero	8,800

Tamaulipas	Matamoros	Matamoros	7,892
Tamaulipas	Nuevo Laredo	Nuevo Laredo	0,826
Tamaulipas	Reynosa	Reynosa	9,078
Tamaulipas	San Fernando	San Fernando	0,053
Tamaulipas	Soto La Marina	Soto la Marina	0,019
Tamaulipas	Tampico	Tampico	0,990

CUADRO 3.1: Zona industrial región Noreste

Además se planea que la construcción esté en una zona cuyo consumo de gas natural sea mayor o igual a 5 MMpcd.

Este problema se modelará utilizando la teoría de gráficas de la siguiente manera:

- Para fines prácticos se considera una gráfica simple $G = (V, A)$, es decir que se cumple que $d(a, b) = d(b, a)$ para todo $a, b \in V$.
- Se considera un vértice por localidad en la que se encuentre una zona industrial, cada vértice se establecerá en el lugar que cumpla con los siguientes criterios:
 1. Estar en un municipio de la zona industrial Noreste.
 2. Estar cerca del paso de gasoducto.
 3. Estar cerca de una zona o parque industrial, se buscará la más importante siguiendo los datos del libro: El Gas Natural, y su geografía industrial en México. [11]

Las direcciones de las zonas o parques industriales, siguiendo los datos de sus ramas principales de consumo, están en la tabla 3.2.

- No se consideran todas las aristas entre cada par vértices, si no que se procurará seguir la ruta de los gasoductos mostrada en el mapa 4.16 (los cuales se encuentran en el capítulo 5. *Figuras y mapas*), sean éstos privados o del estado; así como la ruta de la carretera federal.
- Los datos se tomarán de Google maps.
- Se usará una escala a razón de $10\text{km} = 1 \text{ u}$ (unidad).

A cada municipio se le asignará un número, siguiendo los datos de la tabla 3.1, así como también se darán las direcciones de las zonas o parques industriales:

	Municipio	Zona o parque industrial
1.	Ahumada	Agroquimicos Santa Rita S.A. DE C.V, 32840 Villa Ahumada, Chih.
2.	Aldama	Parque Industrial Aldama, Carmen Serdán #81, Francisco Villa, 32900 Juan Aldama, Chih.
3.	Allende	La Alegría Grupo Agroempresarial, Hidalgo S/N, La Cordereña, 33920 Valle de Ignacio Allende, Chih.
4.	Camargo	Parque Industrial El Soldado, Ben Pérez #7, Santa Rosalía, 33759 Santa Rosalía, Chih.
5.	Cuahutémoc	CASTMETAL FWF de México, Avenida Rio Tutuaca S/N,, Cuauhtémoc, Parque Industrial Xalostoc
6.	Chihuahua	Complejo Industrial Chihuahua
7.	Delicias	ALPURA Delicias, Av Plutarco Elías Calles #700, Primero de Mayo, 33088 Delicias, Chih.
8.	Jiménez	The dal Internacional SA de CV, División del Nte. #10, Obrera, 33980 José Mariano Jiménez, Chih.
9.	Juárez	Parque Industrial Omega, Omega, 32410 Cd Juárez, Chih.
10.	Rosales	Santa Cruz de Rosales, 33120 Chih.
11.	Saucillo	Agroindustrial Fernández, Av. 3a y Quinta, Altavista, 33620 Saucillo, Chih.
12.	Santa Isabel	Viñedo El Molino Don Tomás, 33270 Santa Isabel, Chih.

13.	Castaños	Fasemex, Carretera 57 Km 178, California, 25870 Castaños, Coah.
14.	Francisco I. Madero	Zona Industrial, 27908 Francisco I. Madero, Coah.
15.	General Cepeda	General Cepeda, 25950 Coah.
16.	Matamoros	Laguna Industrial Park, Carr. Torreón - Matamoros km. 13.5, Sin Nombre de Col 2, 27440 Matamoros, Coah.
17.	Monclova	Parque industrial de Monclova, Lib. Lic. Carlos Salinas de Gortari s/n, 25815 Monclova, Coah.
18.	Parras	Desarrollo Industrial Cactus Valley Parras, 16 de Septiembre S/N, Zona Industrial, 27989 Parras de la Fuente, Coah.
19.	Piedras Negras	Hendrickson Spring México, Ave. Rassini #801, Colonia Nicolas Bravo, 26030 Piedras Negras, Coah.
20.	Ramos Arizpe	VYNMSA Ramos Arizpe Industrial Park, Industrial Park, Carretera antigua Saltillo-Monclova km 11 VYNMSA, Sector II, 25908 Ramos Arizpe, Coah.
21.	San Pedro	Wrangler San Pedro, Sin Nombre de Col 2, San Pedro, Coah.
22.	Sierra Mojada	Química del Rey, Laguna del Rey, Coah.
23.	Torreón	Parque Industrial las Américas, 27278, Torreón, Coah.
24.	Cuencamé	Minerales de Avino (MIASA), Carretera Cuencamé-Torreón Km.2, 35800 Cuencamé, Dgo.
25.	Durango	Ciudad Industrial, 34208 Durango, Dgo.

26.	Gómez Palacio	Parque Industrial Lagunero, 35078, Gómez Palacio, Dgo.
27.	Lerdo	Siete leguas, Blvd. del Tecnológico, Plácido Domingo, 35159 Cd Lerdo, Dgo.
28.	Mapimí	Planta incubadora Citra, 35236 Mapimí, Dgo.
29.	Peñón Blanco	34743 Peñón Blanco, Dgo.
30.	Tlahualilo	OMJC Planta Tlahualilo, Chihuahua 110, Ceceda, Las Colonias, 35290 Tlahualilo de Zaragoza, Dgo.
31.	Apodaca	Parque Industrial Apodaca, 66603, Av. Parque Industrial Monterrey 13, Monterrey, 66603 Cd Apodaca, N.L.
32.	Cadereyta de Jiménez	Refinería Cadereyta, Carr. Libre Monterrey - Reynosa Km. 36.5, Sin Nombre de Col 3, 67488 Cadereyta Jiménez, N.L.
33.	China	Industrias Manufactureras De Tamaulipas, S.A . De C.V., Arroyo Las Lajas 102, Col. Vista Hermosa, 67050 China, N.L.
34.	San Pedro Garza García	IFM efector S de RL de CV, Ave. Arq. Pedro Ramírez Vázquez 200-4 Planta Baja, Col. Valle Oriente, 66269, San Pedro Garza García, N.L.
35.	General Bravo	Bravo Beef Master, 67000 Gral Bravo, N.L.
36.	General Escobedo	Divisiones de papel Escobedo, C. Zuazua 312, Lázaro Cárdenas, 66053 Cd Gral Escobedo, N.L.
37.	Mina	Triturados Mina Sa de Cv, Calle sin nombre, 65100 Mina, N.L.

38.	Monterrey	Owens ilinois (Vidriera Monterrey Vimos), Magallanes 517, Treviño, 64570 Monterrey, N.L.
39.	Pesquería	Centro Industrial Ternium, Carr. Santa Ma- ría La Floreña Km 15, Rancho La Joya, Eji- do La Victoria, 66650 Pesquería, N.L.
40.	San Nicolás de los Garza	Parque Industrial Barragán, 66425, San Ni- colás de los Garza, N.L.
41.	Hidalgo	CEMEX, Planta Cemento Hidalgo, Galea- na 300 Sur; Zona Centro, Hidalgo, Código Postal 65600, Nuevo León, México
42.	Santa Catarina	Industrias Vago De México, S.A. de C.V. Miguel Hidalgo y Costilla 743-A, La Fama, 66100 Santa Catarina, N.L.
43.	Abasolo	Semillas y Granos de Abasolo, SA de CV, Abasolo, KM 1, 87760 Abasolo, Tamps.
44.	Aldama	Aldama, 89670 Tamps.
45.	Altamira	Altamira Servicios de Infraestructura S.A. de C.V., Ave Desarrollo Textil No. 3000 Eji- do Santa Amalia, 89602 Altamira, Tamps.
46.	Ciudad Madero	Refinería PEMEX (Av. Álvaro Obregón), Av. Álvaro Obregón No.3014, Emilio Carranza, 89530 Cd Madero, Tamps.
47.	Matamoros	Parque Industrial del Norte, 87316 Heroica Matamoros, Tamps.
48.	Nuevo Laredo	Parque Industrial FINSA Nuevo Laredo, Finsa Industrial Park, 88275 Nuevo Lare- do, Tamps.
49.	Reynosa	Parque Industrial Reynosa, 88757 Reynosa, Tamps.

50.	San Fernando	Cooperativa de consumo agropecuario loma de santa cruz S.C.L. de C.V., Victoria km 170-7, 87600 San Fernando, Tamps.
51.	Soto La Marina	Empacadora Ganadera de Tamaulipas, Estación Manuel - Matamoros Km 144.5, 87670 Soto la Marina, Tamps.
52.	Tampico	Lacteos De Tamaulipas S.A. De C.V., Agustín de Iturbide 1210, Zona Centro, 89137 Tampico, Tamps.

CUADRO 3.2: Numeración de localidades

Los mapas de los municipios con su respectiva numeración se pueden apreciar en los Mapas 4.17, 4.18, 4.19, 4.20 y 4.21.

Las aristas a considerar son las siguientes, con su respectivo peso ($w(a_i)$ con $a_i \in A$ y $w(a_i) = 1 = 10 \text{ km}$):

a_i	Ruta	#	$w(a_i)$
a_1	Juárez-Ahumada	(9, 1)	13
a_2	Ahumada-Chihuahua	(1, 6)	17.7
a_3	Chihuahua-Aldama	(6, 2)	4
a_4	Chihuahua-Cuahutémoc	(6, 5)	11.5
a_5	Chihuahua-Santa Isabel	(6, 12)	5.9
a_6	Chihuahua-Rosales	(6, 10)	10.5
a_7	Aldama-Rosales	(2, 10)	10.4
a_8	Aldama-Camargo	(2, 4)	18.1
a_9	Aldama-Saucillo	(2, 11)	13.4
a_{10}	Aldama-Delicias	(2, 7)	11.2
a_{11}	Cuahutémoc-Santa Isabel	(5, 12)	5.8
a_{12}	Rosales-Delicias	(10, 7)	1.4
a_{13}	Rosales-Saucillo	(10, 11)	3.6
a_{14}	Saucillo-Camargo	(11, 4)	5.2
a_{15}	Saucillo-Allende	(11, 3)	17.1

a_{16}	Delicias-Allende	(7, 3)	19
a_{17}	Camargo-Sierra Mojada	(4, 22)	24.9
a_{18}	Camargo-Allende	(4, 3)	12.3
a_{19}	Camargo-Jiménez	(4, 8)	7
a_{20}	Allende-Jiménez	(3, 8)	5.8
a_{21}	Jiménez-Sierra Mojada	(8, 22)	17.9
a_{22}	Jiménez-Mapimí	(8, 28)	21.2
a_{23}	Sierra Mojada-Tlahualillo	(22, 30)	19.7
a_{24}	Sierra Mojada-Francisco I. Madero	(22, 14)	17
a_{25}	Mapimí-Tlahualillo	(28, 30)	8.6
a_{26}	Mapimí-Lerdo	(28, 27)	5.5
a_{27}	Mapimí-Gómez Palacio	(28, 26)	5.7
a_{28}	Tlahualillo-Francisco I. Madero	(30, 14)	5.5
a_{29}	Tlahualillo-Gómez Palacio	(30, 26)	7.5
a_{30}	Durango-Peñón Blanco	(25, 29)	13.2
a_{31}	Durango-Cuencamé	(25, 24)	14
a_{32}	Peñón Blanco-Cuencamé	(29, 24)	5
a_{33}	Cuencamé-Lerdo	(24, 27)	9.6
a_{34}	Cuencamé-Torreón	(24, 23)	12.3
a_{35}	Lerdo-Gómez Palacio	(27, 26)	1.1
a_{36}	Lerdo-Torreón	(27, 23)	2.3
a_{37}	Torreón-Matamoros	(23, 16)	0.8
a_{38}	Torreón-Gómez Palacio	(23, 26)	1.4
a_{39}	Francisco I. Madero-San Pedro	(14, 21)	3.4
a_{40}	San Pedro-Parras	(21, 18)	12.4
a_{41}	Parras-Ramos Arizpe	(18, 20)	14.7
a_{42}	Parras-General Cepeda	(18, 15)	8.3
a_{43}	General Cepeda-Ramos Arizpe	(15, 20)	6.8
a_{44}	Ramos Arizpe-Castaños	(20, 13)	17.4
a_{45}	Castaños-Monclova	(13, 17)	2.2
a_{46}	Monclova-Piedras Negras	(17, 19)	23.6

a_{47}	Piedras Negras-Nuevo Laredo	(19, 48)	18.8
a_{48}	Castaños-Mina	(13, 37)	14
a_{49}	Mina-Hidalgo	(37, 41)	0.8
a_{50}	Ramos Arizpe-Santa Catarina	(20, 42)	7
a_{51}	Santa Catarina-Mina	(42, 37)	8
a_{52}	General Escobedo-Mina	(36, 37)	3.4
a_{53}	Santa Catarina-San Pedro Garza García	(42, 34)	1.4
a_{54}	San Pedro Garza García-Monterrey	(34, 38)	0.9
a_{55}	Monterrey-San Nicolás de los Garza	(38, 40)	1.2
a_{56}	Monterrey-General Escobedo	(38, 36)	1.5
a_{57}	General Escobedo-Apodaca	(36, 31)	2.1
a_{58}	Monterrey-Cadereyta de Jiménez	(38, 32)	4.3
a_{59}	San Nicolás de los Garza-Apodaca	(40, 31)	1.8
a_{60}	Apodaca-Pesquería	(31, 39)	2.9
a_{61}	Pesquería-Cadereyta de Jiménez	(39, 32)	3
a_{62}	Cadereyta de Jiménez-China	(32, 33)	7.9
a_{63}	General Bravo-Reynosa	(35, 49)	10.3
a_{64}	Reynosa-Matamoros	(49, 47)	7.9
a_{65}	Reynosa-China	(49, 33)	11.8
a_{66}	Reynosa-San Fernando	(49, 50)	15.1
a_{67}	China-San Fernando	(33, 50)	20.2
a_{68}	San Fernando-Abasolo	(50, 43)	10.1
a_{69}	San Fernando-Soto la Marina	(50, 51)	13.2
a_{70}	Abasolo-Soto la Marina	(43, 51)	5
a_{71}	Soto la Marina-Aldama	(51, 44)	11
a_{72}	Aldama-Altamira	(44, 45)	8.7
a_{73}	Altamira-Tampico	(45, 52)	3.4
a_{74}	Altamira-Ciudad Madero	(45, 46)	3.2
a_{75}	Tampico-Ciudad Madero	(52, 46)	0.8

CUADRO 3.3: Distancias

Ahora nos fijamos en las columnas $\#$ y $w(a_i)$, las cuales serán las que escribiremos en el

código (el cual se basará en los códigos 3.6.4, 3.6.5 y 3.6.6).

Solución: Agregamos el código de la misma manera, sólo cambiamos el apartado de *DiGraph* por *Graph*, ya que tenemos una gráfica simple.

```
[1]: import networkx as nx
```

```
[2]: V={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,
26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,
49,50,51,52}
```

```
[3]: A= [(9,1,13),(1,6,17.7),(6,2,4),(6,5,11.5),(6,12,5.9),
(6,10,10.5),(2,10,10.4),(2,4,18.1),(2,11,13.4),(2,7,11.2),
(5,12,5.8),(10,7,1.4),(10,11,3.6),(11,4,5.2),(11,3,17.1),
(7,3,19),(4,22,24.9),(4,3,12.3),(4,8,7),(3,8,5.8),(8,22,17.9),
(8,28,21.2),(22,30,19.7),(22,14,17),(28,30,8.6),(28,27,5.5),
(28,26,5.7),(30,14,5.5),(30,26,7.5),(25,29,13.2),(25,24,14),
(29,24,5),(24,27,9.6),(24,23,12.3),(27,26,1.1),(27,23,2.3),
(23,16,0.8),(23,26,1.4),(14,21,3.4),(21,18,12.4),(18,20,14.7),
(18,15,8.3),(15,20,6.8),(20,13,17.4),(13,17,2.2),(17,19,23.6),
(19,48,18.8),(13,37,14),(37,41,0.8),(20,42,7),(42,37,8),(36,37,3.4),
(42,34,1.4),(34,38,0.9),(38,40,1.2),(38,36,1.5),(36,31,2.1),
(38,32,4.3),(40,31,1.8),(31,39,2.9),(39,32,3),(32,33,7.9),
(35,49,10.3),(49,47,7.9),(49,33,11.8),(49,50,15.1),(33,50,20.2),
(50,43,10.1),(50,51,13.2),(43,51,5),(51,44,11),(44,45,8.7),
(45,52,3.4),(45,46,3.2),(52,46,0.8)]
```

```
[4]: G=nx.Graph()
G.add_nodes_from(V)
G.add_weighted_edges_from(A)
```

```
[5]: weight=nx.get_edge_attributes(G,'weight')
```

Verificamos que los datos sean correctos,


```
[6]: print(nx.info(G))
```

Name:

Type: Graph

Number of nodes: 52

Number of edges: 75

Average degree: 2.8846

```
[7]: nx.barycenter(G, weight='weight')
```

```
[7]: [20]
```

Obteniendo que el centro de la gráfica es el vértice 20, es decir, el municipio de Ramos Arizpe, ubicado en el estado de Coahuila de Zaragoza, con un consumo de 12.130 MMpcd, el cual cumple con la condición de un consumo mayor o igual a 5 MMpcd. Por lo tanto el lugar óptimo para la construcción del centro de control y monitoreo de los gasoductos es en el municipio de Ramos Arizpe, Coahuila de Zaragoza. A continuación se mostrarán todos los caminos mínimos desde Ramos Arizpe y su respectiva longitud.

```
[9]: p1 = nx.shortest_path(G, source=20, weight="weight")
```

```
[10]: longitud = nx.shortest_path_length(G, source=20, weight="weight")
```

```
[11]: print("Todos los caminos mínimos desde Ramos Arizpe: ", p1)
print("Longitud de los caminos mínimos: ", longitud)
```

Todos los caminos mínimos desde Ramos Arizpe: {

20: [20],

18: [20, 18],

15: [20, 15],

13: [20, 13],

42: [20, 42],

37: [20, 42, 34, 38, 36, 37],

34: [20, 42, 34],

38: [20, 42, 34, 38],

40: [20, 42, 34, 38, 40],

36: [20, 42, 34, 38, 36],
32: [20, 42, 34, 38, 32],
31: [20, 42, 34, 38, 40, 31],
39: [20, 42, 34, 38, 40, 31, 39],
33: [20, 42, 34, 38, 32, 33],
41: [20, 42, 34, 38, 36, 37, 41],
21: [20, 18, 21],
17: [20, 13, 17],
19: [20, 13, 17, 19],
49: [20, 42, 34, 38, 32, 33, 49],
50: [20, 42, 34, 38, 32, 33, 50],
14: [20, 18, 21, 14],
22: [20, 18, 21, 14, 22],
30: [20, 18, 21, 14, 30],
35: [20, 42, 34, 38, 32, 33, 49, 35],
47: [20, 42, 34, 38, 32, 33, 49, 47],
28: [20, 18, 21, 14, 30, 28],
26: [20, 18, 21, 14, 30, 26],
43: [20, 42, 34, 38, 32, 33, 50, 43],
51: [20, 42, 34, 38, 32, 33, 50, 51],
48: [20, 13, 17, 19, 48],
27: [20, 18, 21, 14, 30, 26, 27],
23: [20, 18, 21, 14, 30, 26, 23],
8: [20, 18, 21, 14, 22, 8],
24: [20, 18, 21, 14, 30, 26, 27, 24],
16: [20, 18, 21, 14, 30, 26, 23, 16],
4: [20, 18, 21, 14, 22, 4],
25: [20, 18, 21, 14, 30, 26, 27, 24, 25],
29: [20, 18, 21, 14, 30, 26, 27, 24, 29],
44: [20, 42, 34, 38, 32, 33, 50, 51, 44],
3: [20, 18, 21, 14, 22, 8, 3],
45: [20, 42, 34, 38, 32, 33, 50, 51, 44, 45],

11: [20, 18, 21, 14, 22, 4, 11],
 7: [20, 18, 21, 14, 22, 4, 11, 10, 7],
 2: [20, 18, 21, 14, 22, 4, 2],
 52: [20, 42, 34, 38, 32, 33,
 50, 51, 44, 45, 52],
 46: [20, 42, 34, 38, 32, 33, 50, 51, 44, 45, 46],
 10: [20, 18, 21, 14, 22, 4, 11, 10],
 6: [20, 18, 21, 14, 22, 4, 11, 10, 6],
 1: [20, 18, 21, 14, 22, 4, 11, 10, 6, 1],
 5: [20, 18, 21, 14, 22, 4, 11, 10, 6, 5],
 12: [20, 18, 21, 14, 22, 4, 11, 10, 6, 12],
 9: [20, 18, 21, 14, 22, 4, 11, 10, 6, 1, 9]}

Longitud de los caminos mínimos: {

20: 0,	15: 6.8,
42: 7,	34: 8.4,
38: 9.3,	40: 10.5,
36: 10.8,	31: 12.3,
32: 13.600000000000001,	37: 14.200000000000001,
18: 14.7,	41: 15.000000000000002,
39: 15.200000000000001,	13: 17.4,
17: 19.599999999999998,	33: 21.5,
21: 27.1,	14: 30.5,
49: 33.3,	30: 36.0,
47: 41.199999999999996,	50: 41.7,
19: 43.2,	26: 43.5,
35: 43.599999999999994,	28: 44.6,
27: 44.6,	23: 44.9,
16: 45.699999999999996,	22: 47.5,
43: 51.800000000000004,	24: 54.2,
51: 54.900000000000006,	29: 59.2,
48: 62.0,	8: 65.4,
44: 65.9,	25: 68.2,

3: 71.2,	4: 72.4,
45: 74.60000000000001,	11: 77.60000000000001,
46: 77.80000000000001,	52: 78.00000000000001,
10: 81.2,	7: 82.60000000000001,
2: 90.5,	6: 91.7,
12: 97.60000000000001,	5: 103.2,
1: 109.4,	9: 122.4}

Observación 3.6.7. Algunas longitudes y caminos podrían verse reducidos si se añaden aristas ajenas a las que inicialmente se plantearon.

Capítulo 4

Mejor ruta dada las condiciones del terreno

4.1. Planteamiento del problema

Caminar, andar en bicicleta, en auto o en autobús, son algunas de las actividades que se realizan día a día en las principales ciudades de México, por lo que en cada caso siempre se prevé la ruta más corta, sin embargo estas rutas casi nunca consideran la ruta en función del terreno o condiciones del camino. Por lo que se buscará obtener la mejor ruta considerando las condiciones del terreno, ya sea para decidir que ruta escoger para un recorrido turístico, la ruta óptima para el cableado eléctrico entre un par de municipios, o el trazado de carreteras o vías férreas.

4.2. Modelación matemática del problema

Problema 4.2.1. Una compañía de servicios de paquetería busca bajar los costos de transporte entre sus dos principales bodegas (representadas por los vértices v_1 y v_9 de la figura 4.1), para ello se han dado a la tarea de encontrar la ruta que reduzca sus costos de transporte de una manera eficiente.

Se tendrá una gráfica simple, representando las longitudes de las conexiones directas, se modelará utilizando la teoría de gráficas, de tal manera que los vértices representen las dos bodegas (v_1 y v_9) y los demás serán puntos de interés, las aristas las conexiones

directas, y los pesos en las aristas la distancia en kilómetros, en razón $10 \text{ km} = 1$. Así como también una tabla en las que se especificará el tipo de terreno (ciudad, carretera o mixto) de cada conexión directa. Además se hará una comparativa de costos de transporte, considerando los datos de la Comisión Nacional para el Uso Eficiente de la Energía [5]; consideraremos las siguientes condiciones para determinar el costo total de la nueva ruta:

- Se considerará al vehículo Sprinter Cargo Van Mercedes Benz, el cual es uno de los más usados en las empresas de paquetería, el cual tiene los siguientes rendimientos de diesel (promedio de rendimiento de los 6 modelos existentes [5]):
 - 10.9 km/l en ciudad
 - 14.1 km/l en carretera
 - 12.1 km/l mixto
- Se establecerá un gasto por litro de diesel de \$23.5 MXN;
- Se considerará que la empresa de paquetería cuenta con una flotilla de 60 vehículos, mismos que hacen dos recorridos completos (ida y vuelta) cada uno entre ambas bodegas, de los cuales alguno tendrá solo uno de los siguientes inconvenientes, mismos que no afectarán a todos los vehículos (se especificará la frecuencia de tales inconvenientes):
 - 1 de cada 20: vehículo con mantenimiento deficiente aumenta el consumo de combustible en 30 %;
 - 1 de cada 10: una presión incorrecta en las llantas aumenta el consumo de combustible en 5 %;

A continuación la gráfica con pesos $G = (V, A, w)$:

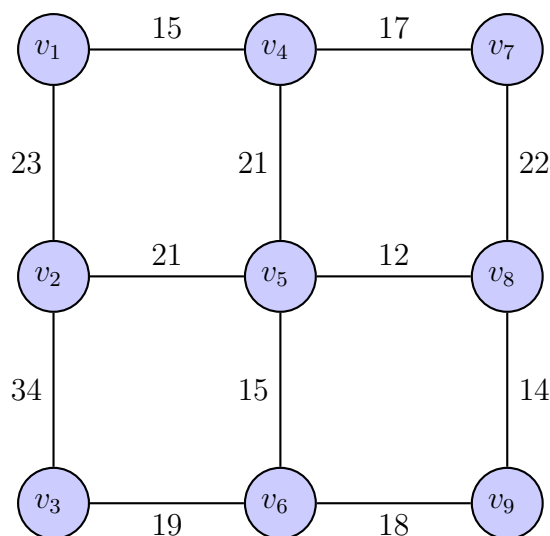


FIGURA 4.1

A continuación se especificará el tipo de terreno de cada una de las aristas de la gráfica $G = (V, A, w)$,

A	$w(a_i)$	Tipo de terreno
(v_1, v_4)	15	Mixto
(v_1, v_2)	23	Carretera
(v_4, v_7)	17	Mixto
(v_4, v_5)	21	Ciudad
(v_7, v_8)	22	Mixto
(v_2, v_5)	21	Ciudad
(v_2, v_3)	34	Carretera
(v_5, v_8)	12	Ciudad
(v_5, v_6)	15	Ciudad
(v_8, v_9)	14	Mixto
(v_3, v_6)	19	Carretera
(v_6, v_9)	18	Carretera

Ahora para considerar las condiciones del terreno se dividirá el peso de las aristas por el rendimiento de diesel, es decir, se obtendrá el consumo de diesel de cada conexión directa en razón $10 L= 1$, la cual se obtiene con la siguiente ecuación:

$$\frac{w(a_i)}{\text{rendimiento de diesel}} = \frac{\frac{km}{1}}{\frac{km}{L}} = \frac{km \cdot L}{km} = L$$

La gráfica a considerar es la siguiente:

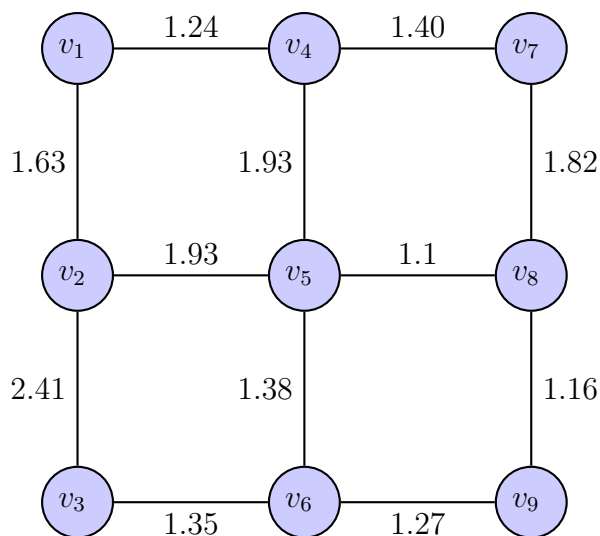


FIGURA 4.2

Solución 4.2.2. Ahora podemos reducir el problema a encontrar el camino de peso mínimo del vértice v_1 al v_9 , para ello utilizaremos el algoritmo de Dijkstra (1.4.1).

Paso 1. Sea $P = \emptyset$ y $Q = V$ y $\delta_v = \infty$ para todo $v \in V - v_1$.

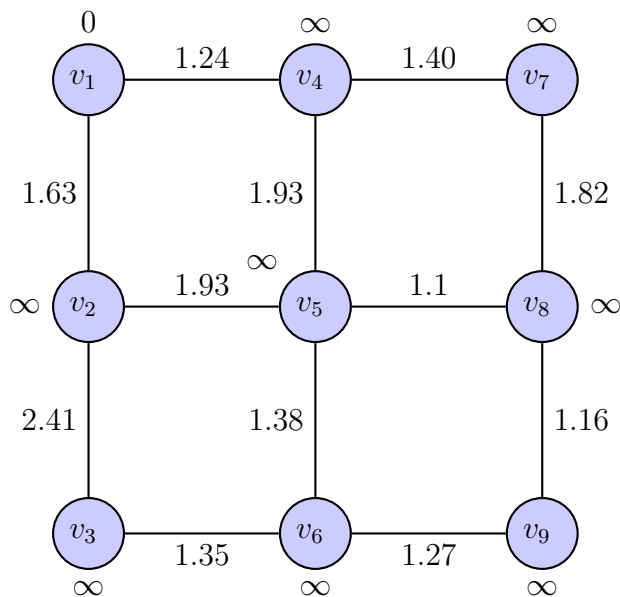


FIGURA 4.3

Paso 2. Como Q no es vacío,

Encontramos un vértice v_1 tal que $\delta_{v_1} = 0 = \min\{\delta_u : u \in Q\}$.

Sea $Q_0 = Q - v_1$ y $P_0 = P \cup v_1$, y como $Q_0 \cap N^+(v_1) = \{v_2, v_4\}$

$$\delta_{v_2} = \min\{\infty, 0 + 1.63\} = 1.63,$$

$$\delta_{v_4} = \min\{\infty, 0 + 1.24\} = 1.24$$

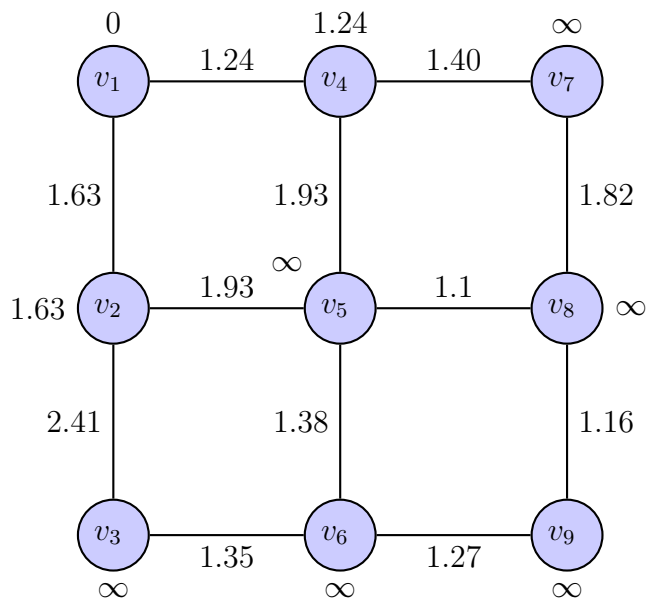


FIGURA 4.4

Paso 3. $Q_0 \neq \emptyset$, se va al paso 2.

Paso 2. Como Q_0 no es vacío,

Encontramos un vértice v_4 tal que $\delta_{v_4} = 1.24 = \min\{\delta_u : u \in Q_0\}$.

Sea $Q_1 = Q_0 - v_4$ y $P_1 = P_0 \cup v_4$, y como $Q_1 \cap N^+(v_4) = \{v_5, v_7\}$

$$\delta_{v_5} = \min\{\infty, 1.24 + 1.93\} = 3.17,$$

$$\delta_{v_7} = \min\{\infty, 1.24 + 1.40\} = 2.64$$

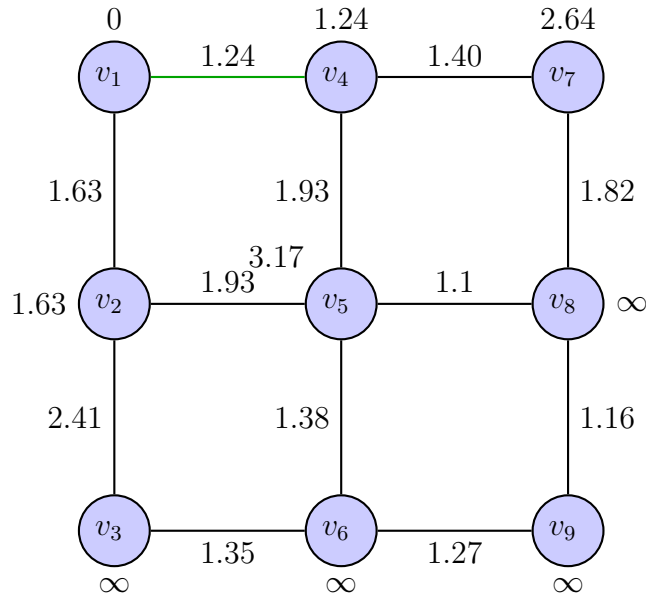


FIGURA 4.5

Paso 3. $Q_1 \neq \emptyset$, se va al paso 2.

Paso 2. Como Q_1 no es vacío,

Encontramos un vértice v_2 tal que $\delta_{v_2} = 1.63 = \min\{\delta_u : u \in Q_1\}$.

Sea $Q_2 = Q_1 - v_2$ y $P_2 = P_1 \cup v_2$, y como $Q_2 \cap N^+(v_2) = \{v_3, v_5\}$

$$\delta_{v_3} = \min\{\infty, 1.63 + 2.41\} = 4.04,$$

$$\delta_{v_5} = \min\{3.17, 1.63 + 1.93\} = 3.17$$

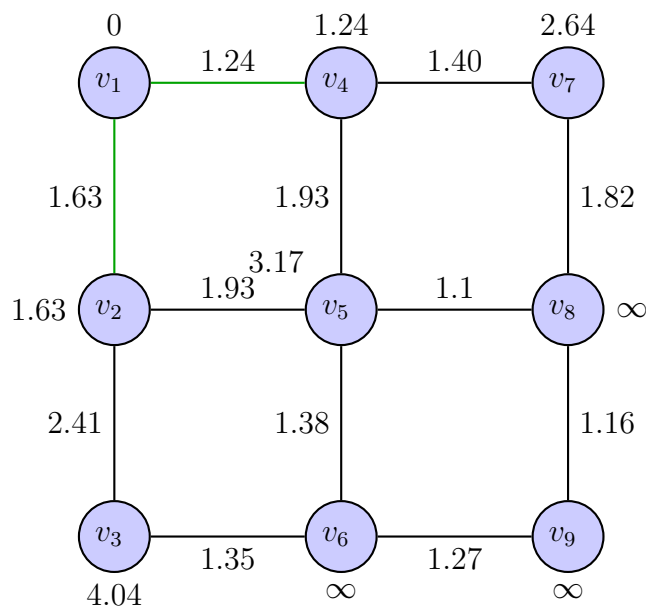


FIGURA 4.6

Paso 3. $Q_2 \neq \emptyset$, se va al paso 2.

Paso 2. Como Q_2 no es vacío,

Encontramos un vértice v_7 tal que $\delta_{v_7} = 2.64 = \min\{\delta_u : u \in Q_2\}$.

Sea $Q_3 = Q_2 - v_7$ y $P_3 = P_2 \cup v_7$, y como $Q_3 \cap N^+(v_7) = \{v_8\}$

$$\delta_{v_8} = \min\{\infty, 2.64 + 1.82\} = 4.46$$

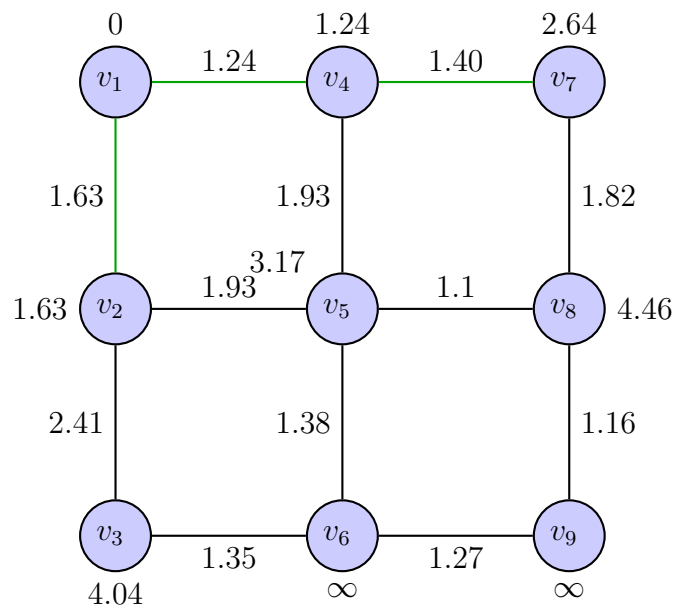


FIGURA 4.7

Paso 3. $Q_3 \neq \emptyset$, se va al paso 2.

Paso 2. Como Q_3 no es vacío,

Encontramos un vértice v_5 tal que $\delta_{v_5} = 3.17 = \min\{\delta_u : u \in Q_3\}$.

Sea $Q_4 = Q_3 - v_5$ y $P_4 = P_3 \cup v_5$, y como $Q_4 \cap N^+(v_5) = \{v_6, v_8\}$

$$\delta_{v_6} = \min\{\infty, 3.17 + 1.38\} = 4.55,$$

$$\delta_{v_8} = \min\{4.46, 3.17 + 1.1\} = 4.27$$

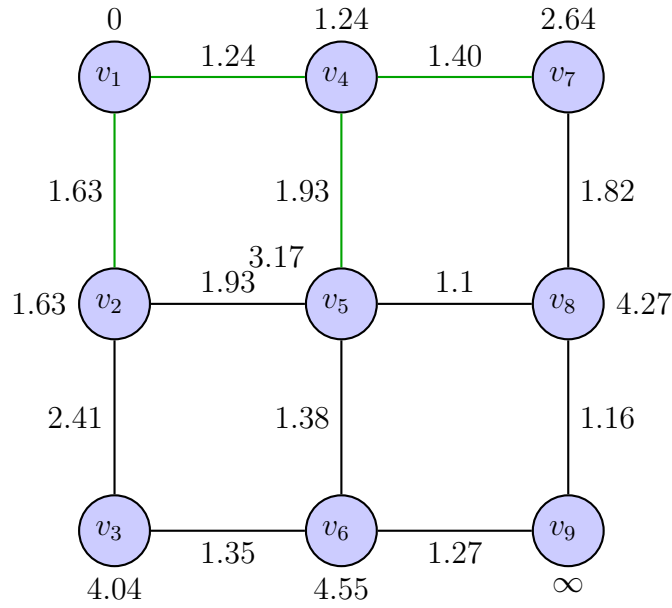


FIGURA 4.8

Paso 3. $Q_4 \neq \emptyset$, se va al paso 2.

Paso 2. Como Q_4 no es vacío,

Encontramos un vértice v_3 tal que $\delta_{v_3} = 4.04 = \min\{\delta_u : u \in Q_4\}$.

Sea $Q_5 = Q_4 - v_3$ y $P_5 = P_4 \cup v_3$, y como $Q_5 \cap N^+(v_3) = \{v_6\}$

$$\delta_{v_6} = \min\{4.55, 4.04 + 1.35\} = 4.55$$

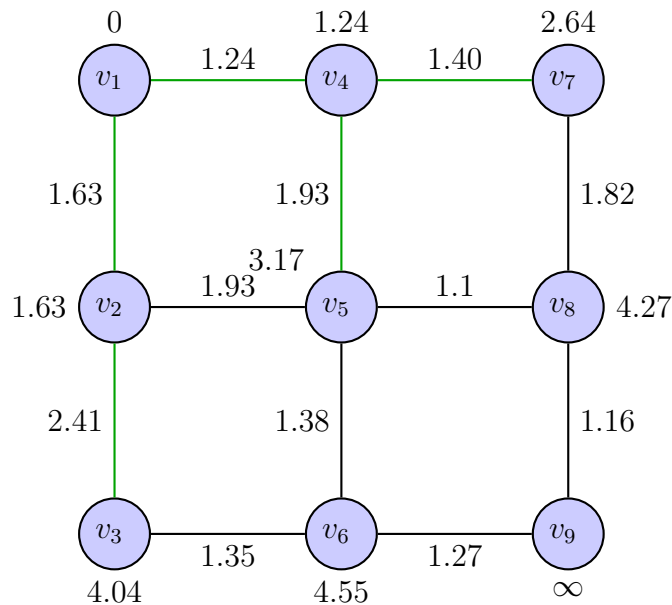


FIGURA 4.9

Paso 3. $Q_5 \neq \emptyset$, se va al paso 2.

Paso 2. Como Q_5 no es vacío,

Encontramos un vértice v_8 tal que $\delta_{v_8} = 4.27 = \min\{\delta_u : u \in Q_5\}$.

Sea $Q_6 = Q_5 - v_8$ y $P_6 = P_5 \cup v_8$, y como $Q_6 \cap N^+(v_8) = \{v_9\}$

$$\delta_{v_9} = \min\{\infty, 4.27 + 1.16\} = 5.43$$

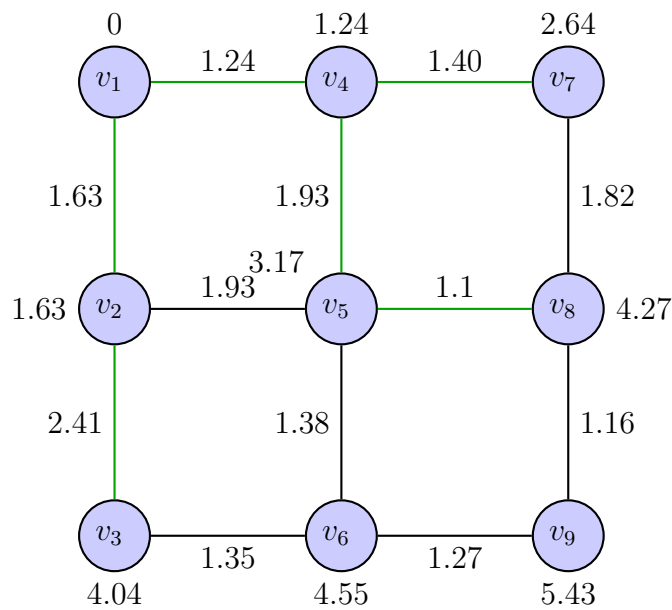


FIGURA 4.10

Paso 3. $Q_6 \neq \emptyset$, se va al paso 2.

Paso 2. Como Q_6 no es vacío,

Encontramos un vértice v_6 tal que $\delta_{v_6} = 4.55 = \min\{\delta_u : u \in Q_6\}$.

Sea $Q_7 = Q_6 - v_6$ y $P_7 = P_6 \cup v_6$, y como $Q_7 \cap N^+(v_6) = \{v_9\}$

$$\delta_{v_9} = \min\{5.43, 4.55 + 1.27\} = 5.43$$

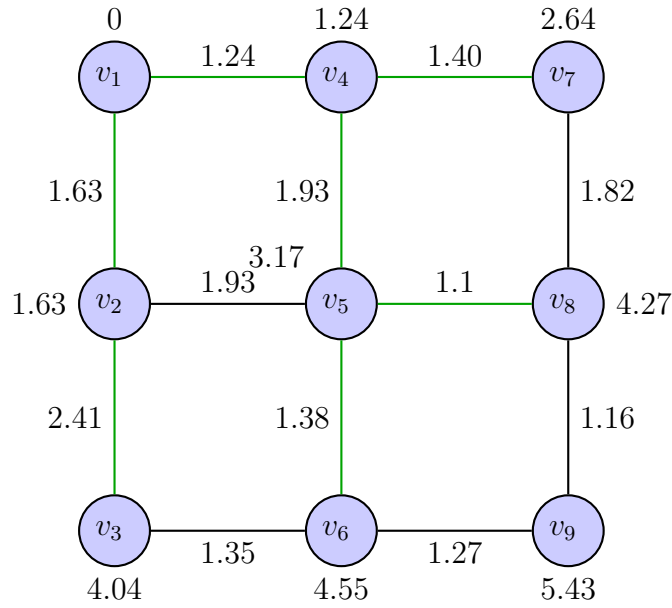


FIGURA 4.11

Paso 3. $Q_7 \neq \emptyset$, se va al paso 2.

Paso 2. Como Q_7 no es vacío,

Encontramos un vértice v_9 tal que $\delta_{v_9} = 5.43 = \min\{\delta_u : u \in Q_7\}$.

Sea $Q_8 = Q_7 - v_9$ y $P_8 = P_7 \cup v_9$, y como $Q_8 \cap N^+(v_9) = \emptyset$

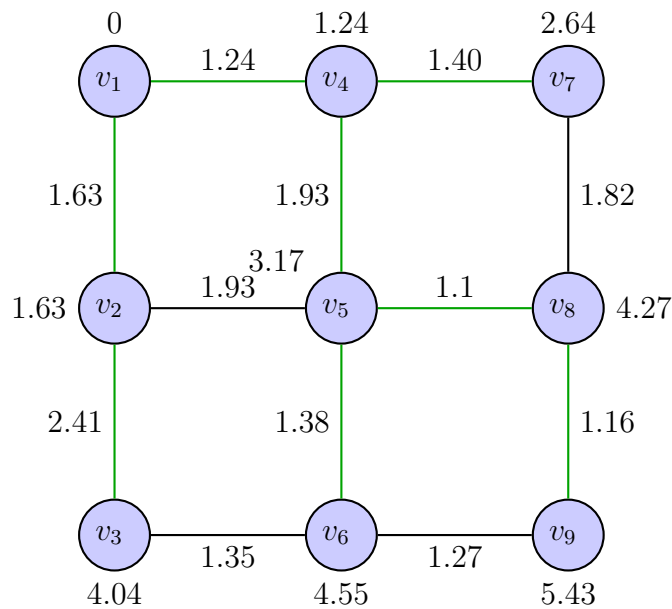


FIGURA 4.12

Paso 3. $P_8 = V$ y $Q_8 = \emptyset$, FIN.

Solución

Entonces el camino de mejor rendimiento desde el vértice v_1 al vértice v_9 , es el siguiente:

$$C = \{v_1, (v_1, v_4), v_4, (v_4, v_5), v_5, (v_5, v_8), v_8, (v_8, v_9), v_9\}$$

y su peso es de 5.43. Por ende, se tiene un consumo total de 54.3 L de diesel. Por lo tanto como cada vehículo hace dos recorridos (ida y vuelta), el consumo de diesel de cada vehículo es de 108.6 L. Además se tiene que tres vehículos tienen un aumento del 30 % y otros seis un aumento del 5 %, entonces el consumo es el siguiente:

$$x = 108.6$$

$$51x + 3 \cdot 1.3x + 6 \cdot 1.05x = 61.2x = 6646.32$$

Entonces tenemos un consumo total de 6646.32 L de diesel, a un precio de \$23.5 MXN por litro, se tiene un gasto de diesel de \$156,188.52 MXN.

Problema 4.2.3. La Secretaría de Cultura busca realzar la belleza y diversidad de tres localidades mayas del estado de Quintana Roo, debido a esto se planea la remodelación de una ex hacienda abandonada, la cual se convertirá en un centro cultural y recreativo. Para la logística de los transportes de cada localidad a la ex hacienda, se planea que se considere el terreno de la ruta, por seguridad de los habitantes.

Se tendrá una gráfica simple, representando las longitudes de las conexiones directas, se modelará utilizando la teoría de gráficas, de tal manera que los vértices representen las localidades (\mathcal{A} , \mathcal{B} y \mathcal{C}), la ex hacienda (\mathcal{H}) y los demás serán puntos de interés, las aristas las conexiones directas, y los pesos en las aristas la distancia en metros, a razón $100\text{m} = 1$. Así como también se especificará, en un tabla, los riesgos y la condición de cada conexión directa.

A continuación la gráfica con pesos $G = (V, A, w)$ (definición 1.3.15), la cual fue dividida en tres gráficas para una mejor visualización:

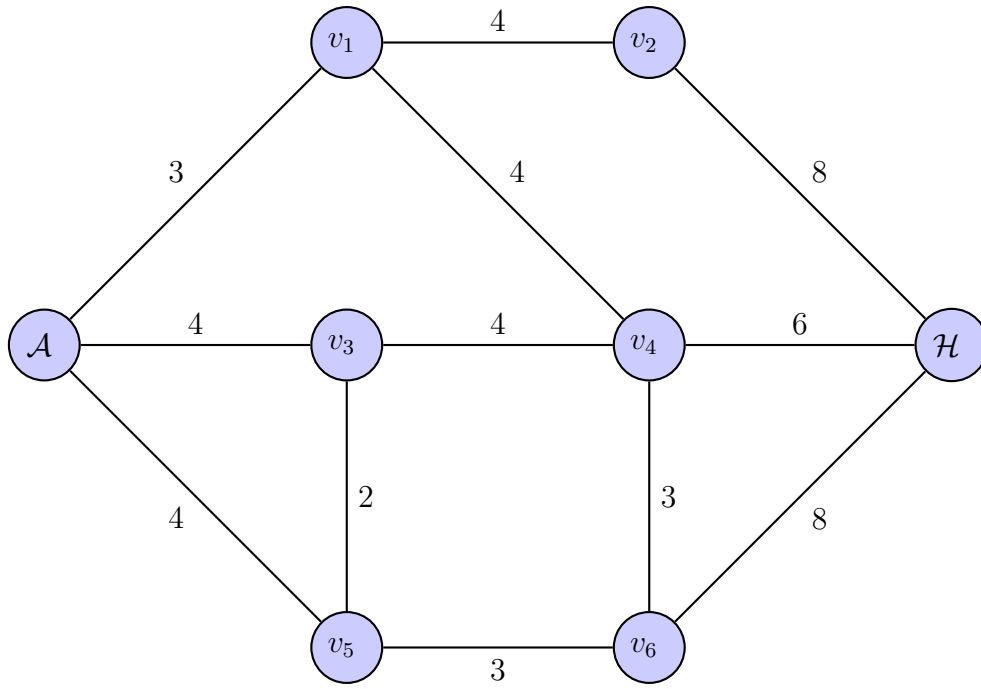


FIGURA 4.13: Gráfica de la localidad \mathcal{A} y las conexiones directas a la ex hacienda \mathcal{H}

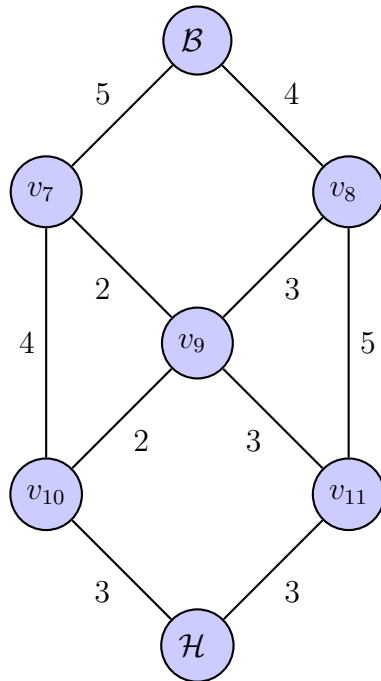


FIGURA 4.14: Gráfica de la localidad \mathcal{B} y las conexiones directas a la ex hacienda \mathcal{H}

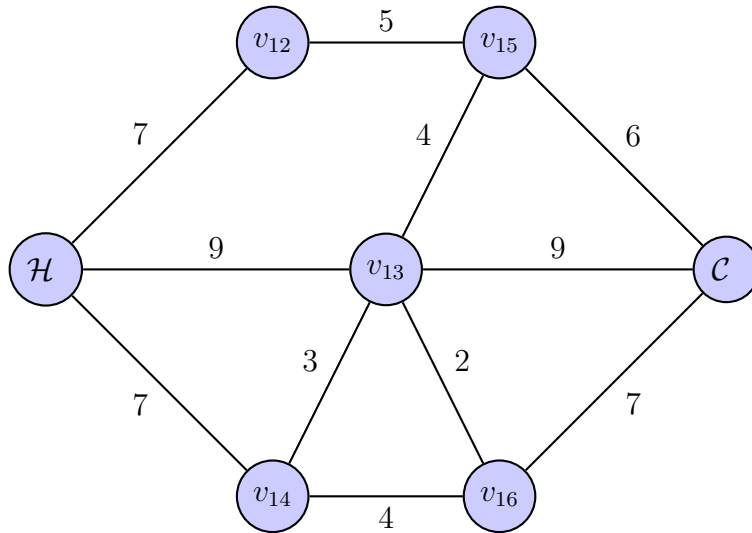


FIGURA 4.15: Gráfica de la localidad \mathcal{C} y las conexiones directas a la ex hacienda \mathcal{H}

A continuación se especificarán los riesgos, así como la tabla de el tipo de terreno de cada una de las aristas de la gráfica $G = (V, A, w)$. Se asignará un multiplicador por cada uno de los riesgos, ya que a mayor riesgo el costo aumenta:

- Riesgo de pinchar una llanta debido a baches o elementos punzocortantes en el camino:
 - Camino de terracería (CT), multiplicar por 1.5;
 - Camino de adoquín o asfalto (CA), multiplicar por 1.1.
- Riesgo de fracturas o deslaves en el camino debido a las lluvias:
 - Camino de terracería (CT), multiplicar por 2.5;
 - Camino de adoquín o asfalto (CA), multiplicar por 1.5;
 - Si el camino está cerca de un acantilado o montaña (CM), multiplicar por 5.

A	$w(a_i)$	Tipo de terreno	CM	Nuevo valor $w^*(a_i)$
(\mathcal{A}, v_1)	3	CT	No	$3 \cdot 1.5 \cdot 2.5 = 11.25$
(\mathcal{A}, v_3)	4	CT	No	$4 \cdot 1.5 \cdot 2.5 = 15$
(\mathcal{A}, v_5)	4	CA	No	$4 \cdot 1.1 \cdot 1.5 = 6.6$
(v_1, v_2)	4	CT	Sí	$4 \cdot 1.5 \cdot 2.5 \cdot 5 = 75$
(v_1, v_4)	4	CA	No	$4 \cdot 1.1 \cdot 1.5 = 6.6$

(v_2, \mathcal{H})	8	CA	No	$8 \cdot 1.1 \cdot 1.5 = 13.2$
(v_3, v_4)	4	CT	No	$4 \cdot 1.5 \cdot 2.5 = 15$
(v_3, v_5)	2	CT	Sí	$2 \cdot 1.5 \cdot 2.5 \cdot 5 = 37.5$
(v_4, v_6)	3	CT	Sí	$3 \cdot 1.5 \cdot 2.5 \cdot 5 = 56.25$
(v_4, \mathcal{H})	6	CA	No	$6 \cdot 1.1 \cdot 1.5 = 9.9$
(v_5, v_6)	3	CA	No	$3 \cdot 1.1 \cdot 1.5 = 4.95$
(v_6, \mathcal{H})	8	CT	No	$4 \cdot 1.5 \cdot 2.5 = 30$
(\mathcal{B}, v_7)	5	CA	No	$5 \cdot 1.1 \cdot 1.5 = 8.25$
(\mathcal{B}, v_8)	4	CA	No	$4 \cdot 1.1 \cdot 1.5 = 6.6$
(v_7, v_9)	2	CT	Sí	$2 \cdot 1.5 \cdot 2.5 \cdot 5 = 37.5$
(v_7, v_{10})	4	CA	No	$4 \cdot 1.1 \cdot 1.5 = 6.6$
(v_8, v_9)	3	CT	No	$3 \cdot 1.5 \cdot 2.5 = 11.25$
(v_8, v_{11})	5	CA	Sí	$5 \cdot 1.1 \cdot 1.5 \cdot 5 = 41.25$
(v_9, v_{10})	2	CT	No	$2 \cdot 1.5 \cdot 2.5 = 7.5$
(v_9, v_{11})	3	CT	No	$3 \cdot 1.5 \cdot 2.5 = 11.25$
(v_{10}, \mathcal{H})	3	CT	No	$3 \cdot 1.5 \cdot 2.5 = 11.25$
(v_{11}, \mathcal{H})	3	CA	No	$3 \cdot 1.1 \cdot 1.5 = 4.95$
(\mathcal{H}, v_{12})	7	CA	No	$7 \cdot 1.1 \cdot 1.5 = 11.55$
(\mathcal{H}, v_{13})	9	CA	No	$9 \cdot 1.1 \cdot 1.5 = 14.85$
(\mathcal{H}, v_{14})	7	CT	No	$7 \cdot 1.5 \cdot 2.5 = 26.25$
(v_{12}, v_{15})	5	CT	Sí	$5 \cdot 1.5 \cdot 2.5 \cdot 5 = 93.75$
(v_{13}, v_{15})	4	CT	No	$4 \cdot 1.5 \cdot 2.5 = 15$
(v_{13}, v_{16})	2	CT	No	$2 \cdot 1.5 \cdot 2.5 = 7.5$
(v_{14}, v_{16})	3	CT	No	$3 \cdot 1.5 \cdot 2.5 = 11.25$
(v_{13}, \mathcal{C})	9	CT	No	$9 \cdot 1.5 \cdot 2.5 = 33.75$
(v_{15}, \mathcal{C})	6	CT	No	$6 \cdot 1.5 \cdot 2.5 = 22.5$
(v_{16}, \mathcal{C})	7	CA	No	$7 \cdot 1.1 \cdot 1.5 = 11.55$

Nótese que al tomar en cuenta la calidad del terreno, se pueden cambiar demasiado los valores, es por ello que es necesario considerarlo al trazar la ruta óptima.

Ahora utilizaremos el programa de NetworkX para encontrar la solución de aplicar el algoritmo de Dijkstra, como se vió en el código 3.6.5, para encontrar el camino que represente un riesgo menor de las localidades (\mathcal{A} , \mathcal{B} y \mathcal{C}) a la ex hacienda (\mathcal{H}).

Solución 4.2.4. Agregamos el código de la misma manera, sólo cambiamos el apartado de *Digraph* por *Graph*, ya que tenemos una gráfica simple.

```
[1]: import networkx as nx
```

```
[2]: V= {'A', 'B', 'C', 'H', 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16}
```

```
[3]: A=[('A', 1, 11.25), ('A', 3, 15), ('A', 5, 6.6), (1, 2, 75), (1, 4, 6.6),
(2, 'H', 13.2), (3, 4, 15), (3, 5, 37.5), (4, 6, 56.25), (4, 'H', 9.9),
(5, 6, 4.95), (6, 'H', 30), ('B', 7, 8.25), ('B', 8, 6.6), (7, 9, 37.5),
(7, 10, 6.6), (8, 9, 11.25), (8, 11, 41.25), (9, 10, 7.5), (9, 11, 11.25),
(10, 'H', 11.25), (11, 'H', 4.95), ('H', 12, 11.55), ('H', 13, 14.85),
('H', 14, 26.25), (12, 15, 93.75), (13, 15, 15), (13, 16, 7.5), (14, 16, 11.25),
(15, 'C', 33.75), (16, 'C', 22.5), (16, 'C', 11.55)]
```

```
[4]: G=nx.Graph()
G.add_nodes_from(V)
G.add_weighted_edges_from(A)
```

```
[5]: weight=nx.get_edge_attributes(G, 'weight')
```

```
[6]: print(nx.info(G))
```

Graph with 20 nodes and 32 edges

```
[7]: s1=nx.shortest_path(G, source='A', target='H', weight='weight',
method='dijkstra')
s2=nx.shortest_path(G, source='B', target='H', weight='weight',
method='dijkstra')
s3=nx.shortest_path(G, source='C', target='H', weight='weight',
method='dijkstra')
```

```
[8]: print("Rutas de riesgo mínimo: ", s1,s2,s3)
```

Rutas de riesgo mínimo:

['A', 1, 4, 'H']

['B', 7, 10, 'H']

['C', 16, 13, 'H']

```
[9]: l1 = nx.shortest_path_length(G, source='A', target='H',
weight='weight')
l2 = nx.shortest_path_length(G, source='B', target='H',
weight='weight')
l3 = nx.shortest_path_length(G, source='C', target='H',
weight='weight')
```

```
[10]: print("Riesgo de cada ruta: ", l1,l2,l3)
```

Riesgo de cada ruta:

27.75

26.1

33.9

De tal modo que obtenemos las rutas de menor riesgo entre cada localidad y la exhaceinda, la cual será un centro cultural, además veremos la distancia de cada camino:

$$\mathcal{A} \longrightarrow v_1 \longrightarrow v_4 \longrightarrow \mathcal{H}$$

$$w(\mathcal{A}, v_1) + w(v_1, v_4) + w(v_4, \mathcal{H}) = 13 = 1300 \text{ m}$$

$$\mathcal{B} \longrightarrow v_7 \longrightarrow v_{10} \longrightarrow \mathcal{H}$$

$$w(\mathcal{B}, v_7) + w(v_7, v_{10}) + w(v_{10}, \mathcal{H}) = 12 = 1200 \text{ m}$$

$$\mathcal{C} \longrightarrow v_{16} \longrightarrow v_{13} \longrightarrow \mathcal{H}$$

$$w(\mathcal{C}, v_{16}) + w(v_{16}, v_{13}) + w(v_{13}, \mathcal{H}) = 18 = 1800 \text{ m}$$

Conclusiones

Esta tesis tiene como objetivo mostrar que utilizando la *teoría de gráficas* podemos encontrar una solución óptima a diversos problemas de tráfico y logística, tales como la asignación de rutas de transporte, ubicaciones óptimas de construcción para centros de emergencia y planeación de ruta más corta.

A través de la aplicación de la teoría de gráficas, se encontró la solución más eficiente a los problemas abordados, la cual incluye una base profunda y clara de conceptos, un análisis detallado del problema, y un código útil que permite resolver cualquier problema similar agregando los datos correspondientes.

Para corroborar la solución se plantearon problemas reales, enfocados a diversos espacios urbanos, desde la logística de pequeñas comunidades en Quintana Roo, hasta la logística en caso de emergencia para la red de gas natural de la región Noreste de México.

Esta investigación arroja una nueva realidad, la gran importancia de la aplicación de las matemáticas, en particular de la teoría de gráficas, para la solución óptima de problemas urbanos.

Figuras y mapas

Cada uno de los mapas presentados en ésta sección, y utilizados en el problema 3.6.1, fueron aportados por el Mtro. Jordy Micheli Thiri3n, quien es uno de los autores del libro: El gas Natural y su geografía industrial en México [11]; las modificaciones a la numeración de los mapas fueron hechas con aprobación de los autores, además dicha numeración coincide con la vista en los Cuadros 3.1 y 3.2 del problema ya mencionado.



FIGURA 4.16: Región Noreste

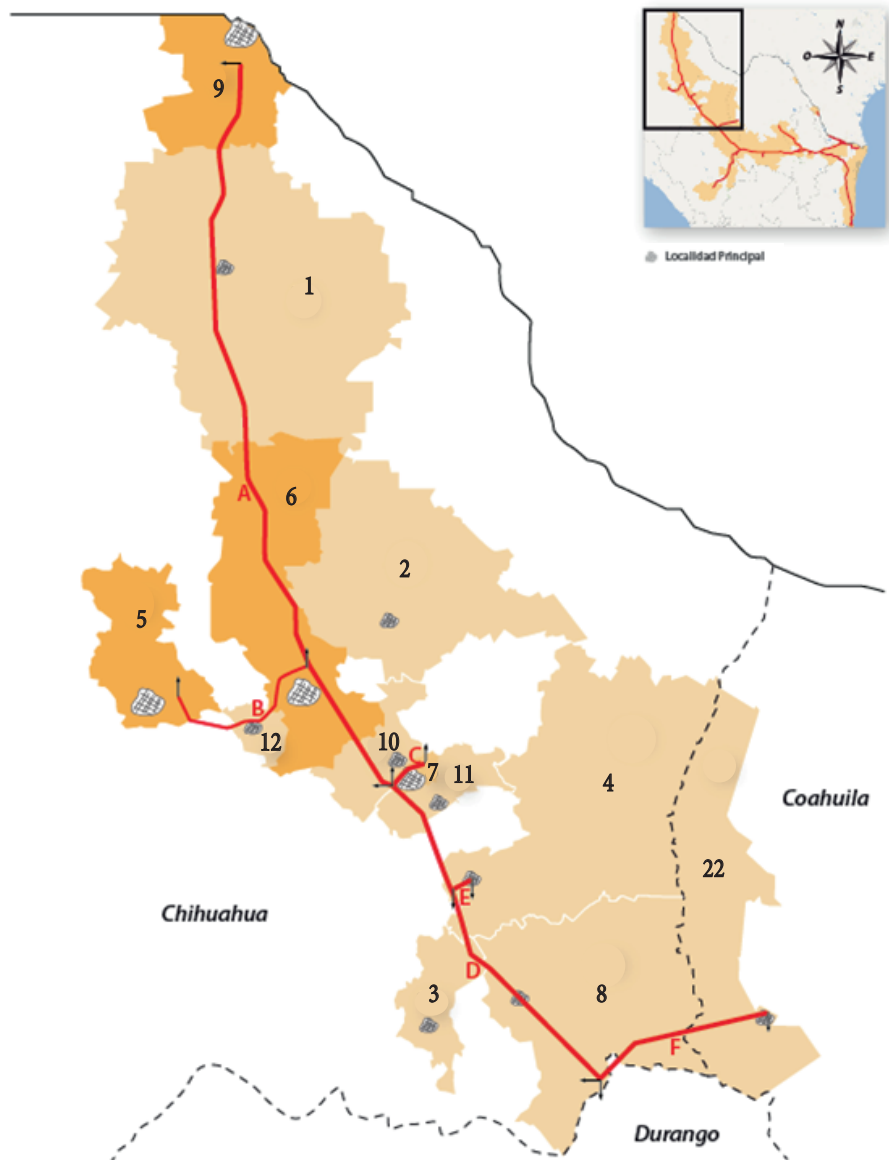


FIGURA 4.17: Región Noreste: mapa 1

SIMBOLOGÍA

Nivel de consumo de gas

- Alto (15 - 94.13)
- Medio Alto (5 - 14.99)
- Medio Bajo (1 - 4.99)
- Bajo (0 - 0.99)

Ductos

- A)** Chihuahua - CD. Juárez
- B)** Terminal de Avalos - Celulosa Anahuac
- C)** El Peñón - CFE Delicias
- D)** Reynosa - Chihuahua
- E)** La laguna - Petroquímica Camargo
- F)** Escalón - Química del Rey



FIGURA 4.18: Región Noreste: mapa 2

SIMBOLOGÍA

Nivel de consumo de gas

- Alto (15 - 94.13)
- Medio Alto (5 - 14.99)
- Medio Bajo (1 - 4.99)
- Bajo (0 - 0.99)

Ductos

- A)** Reynosa - Chihuahua
- B)** Chávez - Durango
- C)** Chávez - Gómez Palacio
- D)** Paila -Parras

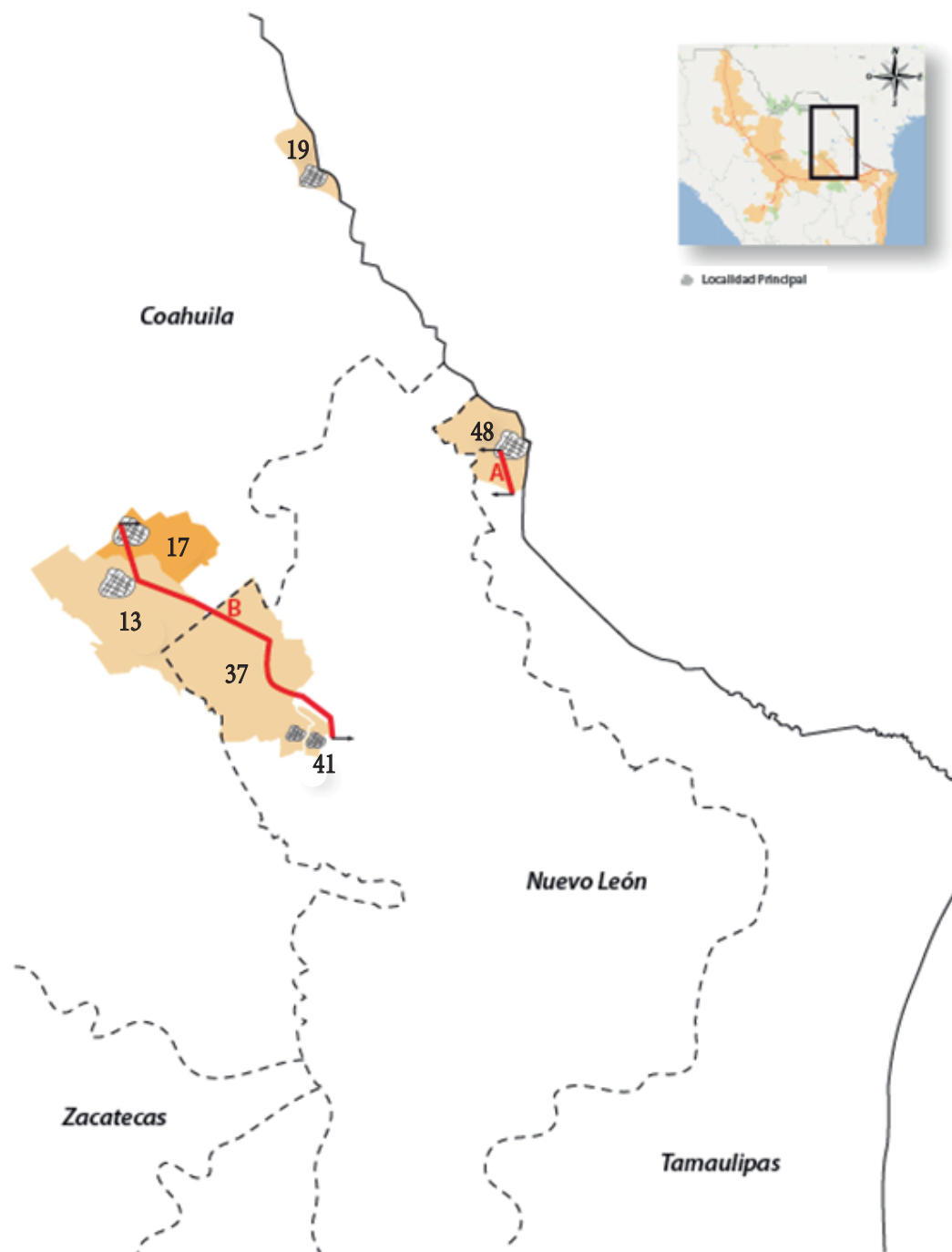


FIGURA 4.19: Región Noreste: mapa 3

SIMBOLOGÍA

Nivel de consumo de gas

- Alto (15 - 94.13)
- Medio Alto (5 - 14.99)
- Medio Bajo (1 - 4.99)
- Bajo (0 - 0.99)

Ductos

- A)** Pandura - Estación 6 (Laredo)
- B)** Escobedo - Castaños

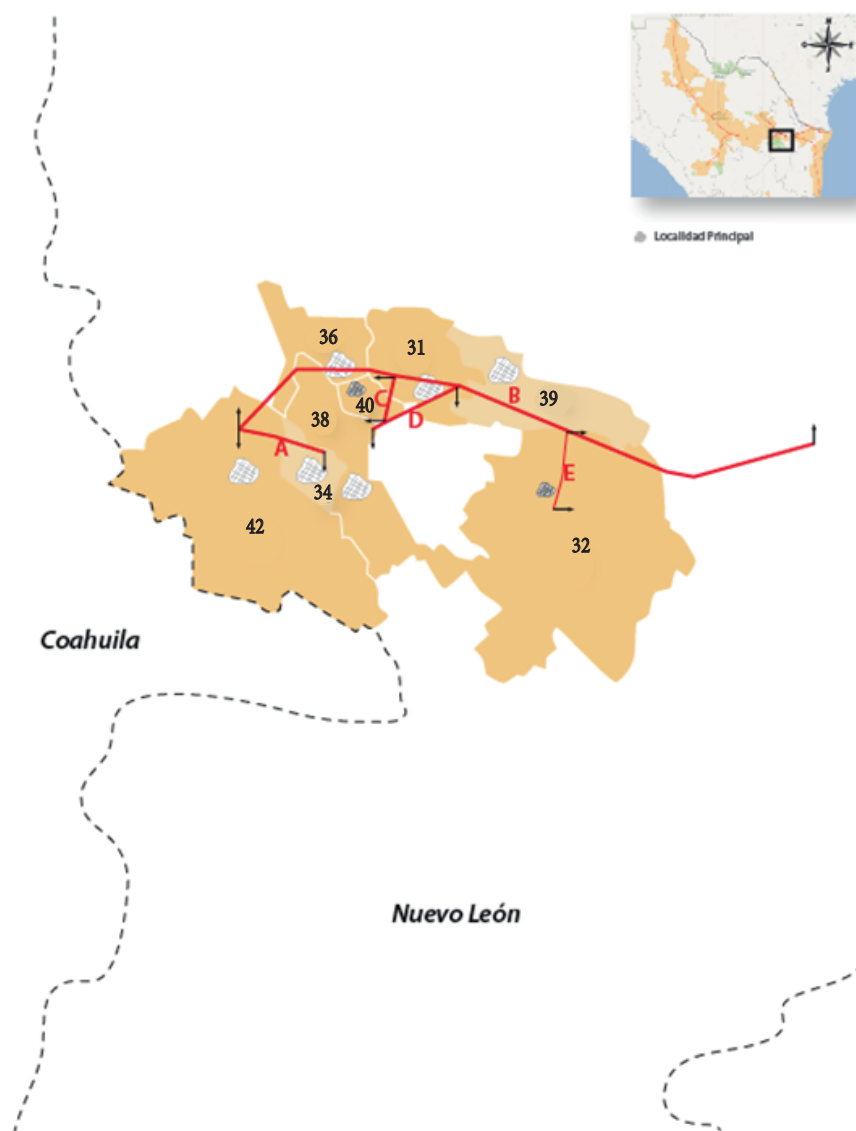


FIGURA 4.20: Región Noreste: mapa 4

SIMBOLOGÍA

Nivel de consumo de gas

- | | |
|--------------------------|-------------------------|
| ■ Alto (15 - 94.13) | ■ Medio Bajo (1 - 4.99) |
| ■ Medio Alto (5 - 14.99) | ■ Bajo (0 - 0.99) |

Ductos

- | | |
|--|--------------------------------------|
| A) Sta. Catarina - CFE San Jerónimo | D) Apodaca - EST. Medición 13 |
| B) Reynosa - Chihuahua | E) Pesquería - Cadereyta |
| C) Sta. Rosa - EST. Medición 13 | |

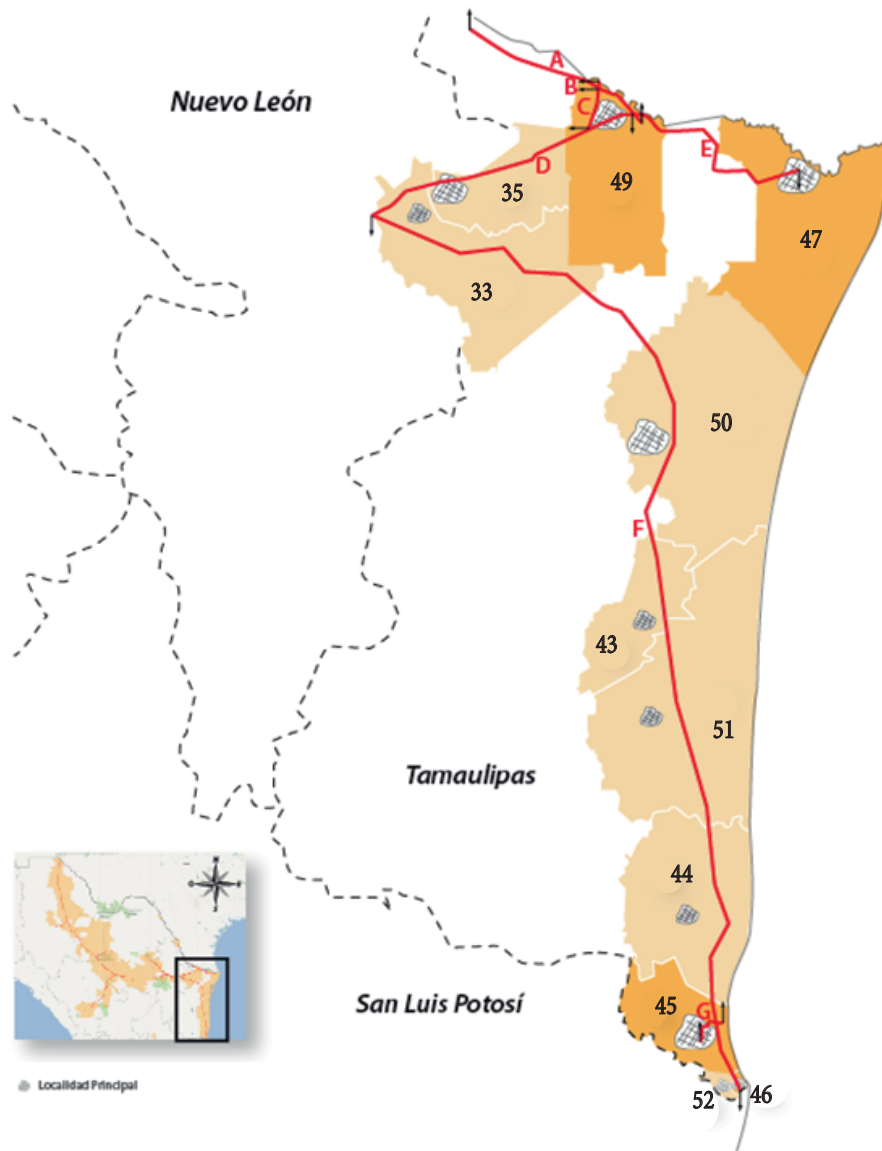


FIGURA 4.21: Región Noreste: mapa 5

SIMBOLOGÍA

Nivel de consumo de gas

- Alto (15 - 94.13)
- Medio Alto (5 - 14.99)
- Medio Bajo (1 - 4.99)
- Bajo (0 - 0.99)

Ductos

- A)** Miguel Alemán - Reynosa
- B)** Arguelles - MDR Bravo
- C)** EST 19 - Arguelles
- G)** Lomas del Real - CFE Colinas
- D)** Reynosa - Chihuahua
- E)** Reynosa - Matamoros
- F)** Cactus - San Fernando - Los Ramones

Bibliografía

- [1] J.M. Aldous, S. Best, and R.J. Wilson. *Graphs and Applications: An Introductory Approach*. The Open University. Springer London, Great Britain, 4^{ta} edición, 2003.
- [2] J. Anzano. *El proceso de urbanización en el mundo (Sección Temario de oposiciones de Geografía e Historia)*. Proyecto Clío 36. ISSN: 1139-6237, 2010. <http://clio.rediris.es/n36/oposicones/tema09.pdf> (Ingresado en 02/14/2024).
- [3] J. Bang-Jensen and G.Z. Gutin. *Digraphs: Theory, Algorithms and Applications*. Springer Monographs in Mathematics. Springer London, New York, 2^{da} edición, 2008.
- [4] J.A. Bondy and U.S.R. Murty. *Graph Theory with Applications*. American Elsevier Publishing Company, Great Britain, 1^{ra} edición, 1976.
- [5] CONUEE. *Catalogo de Rendimientos 2020*. https://www.gob.mx/cms/uploads/attachment/file/601693/Cat_logo_de_Rendimientos_2020_v20.12.pdf. (Ingresado en 02/14/2024).
- [6] Mónika Csikós, Daniel Hoske, and Torsten Ueckerdt. Graph theory. <https://www.math.kit.edu/iag6/lehre/graphtheory2019w/media/main.pdf>. (Ingresado en 02/14/2024).
- [7] Carlos A. Hernández G. and Diana Macías A. *Notas de Gráficas y juegos*. <https://drive.google.com/file/d/1RMJG1J56NWxBQzIWKELxDdkczakV088T/view?usp=sharing> (Ingresado en 02/14/2024).

- [8] Dorit S. Hochbaum. *Lecture Notes for IEOR 266: Graph Algorithms and Network Flows*. Department of Industrial Engineering and Operations Research, University of California, Berkeley, 2014.
- [9] Cousin J. *Topological organization of architectural space, Architectural Design*. Montreal: Presses de l'Université de Montreal, 1970.
- [10] Grason J. *Emerging Methods in Environmental Design and Planning*. Ed, G. T. Moore, M.I.T. Press, Cambridge, Mass., 1970.
- [11] Micheli J., Romero M., and Valle E. *Gas natural y su geografía industrial en México*. Universidad Autónoma Metropolitana, México, 1^a edición, 2013.
- [12] P.H. Levin and Building Research Station (Great Britain). *Use of Graphs to Decide the Optimum Layout of Buildings*. Building research current papers - Design series. Building Research Station, 1964.
- [13] Instituto Politécnico Nacional. *Inicio y Evolución del urbanismo*. <https://www.repositoriodigital.ipn.mx/bitstream/123456789/25348/1/7-InicioEvolucionUrbanismo.pdf>. (Ingreso en 02/14/2024).
- [14] C. St. J. A. Nash-Williams. *Edge-disjoint spanning trees of finite graphs*. Number 1. Journal of the London Mathematical Society, 1961.
- [15] Instituto Nacional para el Federalismo y el Desarrollo Municipal. *Crecimiento urbano y sostenibilidad: retos y desafíos para los municipios de México*. <https://www.gob.mx/inafed/articulos/crecimiento-urbano-y-sostenibilidad-retos-y-desafios-para-los-municipiosdemexico> (Ingresado en 02/14/2024).
- [16] Jaehyun Park. *Successor Matrix, Witness Matrix*. <https://theory.stanford.edu/~virgi/cs367/oldlects/lecture7.pdf>. (Ingresado en 02/14/2024).