Logística de distribución de Bimbo en Monterrey (Introducción)

Mi interés en logística y transportación de productos se remonta a hace alrededor de un año, cuando le pregunté a mi padre (cuyo empleo es gerencia de producción en Grupo Bimbo Monterrey) ¿porqué no podíamos comprar "Roles Glaseados" en la tienda de la esquina, pero si en expendios más grandes como el de la propia fábrica? A lo que me contestó que la causa era que ese producto en específico se producía en la planta de Guadalajara y se distribuía a otras partes de la República, por lo que el número de piezas era limitado en Monterrey. Los productos externos llegaban a la fábrica y se distribuían a los diversos expendios y los expendios distribuían a la demás tiendas.

Los principios de logística se basan en hacer llegar un producto de un lugar a otro de la manera más eficiente, es por esto que el objetivo de esta exploración es modelar todos los caminos posibles que se pueden usar para distribuir Roles Glaseados en el centro de Monterrey mediante la teoría de grafos, para poder obtener el camino más rápido que conecte los expendios, encontrando la ruta más óptima.

Modelación (Monterrey)

La teoría de grafos es un campo de estudios dentro del área de Matemáticas Discretas, el cuál estudia la estructura de los grafos. Un grafo es un conjunto de nodos/vértices (V) y pares (E) que normalmente son representados por aristas/arcos.

Existen dos tipos de grafos, los dirigidos y los no dirigidos. En estos primeros, los pares E son ordenados mientras que en los no dirigidos no son ordenados. Esto se puede ver de la siguiente manera:

Grafo Dirigido

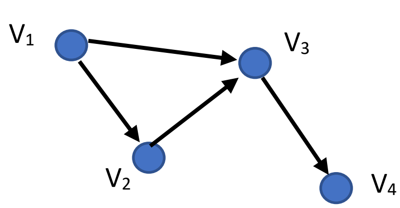


Figura 1. La figura muestra un grafo dirigido simple

Grafo no Dirigido

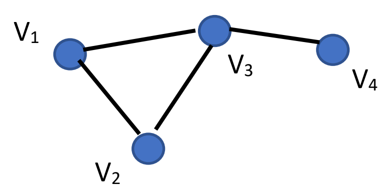


Figura 2. La figura muestra un grafo no dirigido simple

- En un grafo no dirigido G(V,E) a los vértices conectados por una arista se les denomina como extremos adyacentes, en el ejemplo anterior V_1 y V_2 son adyacentes por tanto la arista (V_1 , V_2) es incidente en ambos.
- En un grafo dirigido G(V,E), a los vértices conectados por un arco se les denomina extremos, V_1 es el extremo inicial del arco (V_1, V_2) y V_2 es el extremo final, en esta estructura un extremo final no puede conectarse al inicial directamente, debe de existir otro camino de arcos que lo permita.

Ahora bien para modelar la conexión que existe entre los diversos expendios y la fábrica de Bimbo en Monterrey, tomando en cuenta que la red de distribución de productos se hace de manera global. Se considera lo siguiente:

$$G = (V, E), si\ E' \subseteq E\ y\ E' \neq \emptyset\ \therefore G[E']$$

"G" es el grafo global, si se escoge un subconjunto de arcos (E´) que forman parte del original, y los únicos vértices son los extremos de estos, distintos al vacío, se denomina subgrafo inducido $G[E']$. Por tanto la distribución de Monterrey sería un subgrafo inducido del grafo dirigido global en dónde cada vértice es un expendio (o la fábrica) y los arcos representarían el recorrido de un expendio a otro (incluyendo la fábrica).

Existen 11 expendios principales que Bimbo Norte Monterrey surte directamente y se enlistan de la siguiente manera:

A. Expendio Bimbo, Av. Félix U. Gómez 1322, Terminal, 64580 Monterrey, N.L.
B. Pan Bimbo expendio, Calle Benito Juárez 971, Centro, 64000 Monterrey, N.L.
C. Bimbo, Juan Ignacio Ramón 223, Centro, 64000 Monterrey, N.L.
D. Expendio Bimbo, Av. Cuauhtémoc 1113, Centro, 64000 Monterrey, N.L.
E. Expendio Pan Bimbo, Av. Cristóbal Colón #1000, Industrial, 64490 Monterrey, N.L.
F. Expendio Bimbo, 64190, Av Rodrigo Gómez 11-C, Central, Monterrey, N.L.
G. Expendio Bimbo, Av Abraham Lincoln 3725, Mitras Nte., 64320 Monterrey, N.L.
H. Bimbo Expendio, Calle Pardillo 5958, Paseo de Las Mitras, 64117 Monterrey, N.L.
I. Bimbo Expendio, Tierra y Libertad Sector Centro, 64246 Monterrey, N.L.
J. Pan Bimbo expendio, Calle Burgos 201, Iturbide, 66420 San Nicolás de los Garza, N.L.
K. Expendio de Pan Bimbo, José María Arteaga 219, Centro de Guadalupe, 67100 Guadalupe, N.L.

X. Bimbo, Av. Félix U. Gómez 4203, Coyoacán, 64510 Monterrey, N.L. (Fábrica)

Pueden ser ubicados en el siguiente mapa utilizando el software de Google Maps:

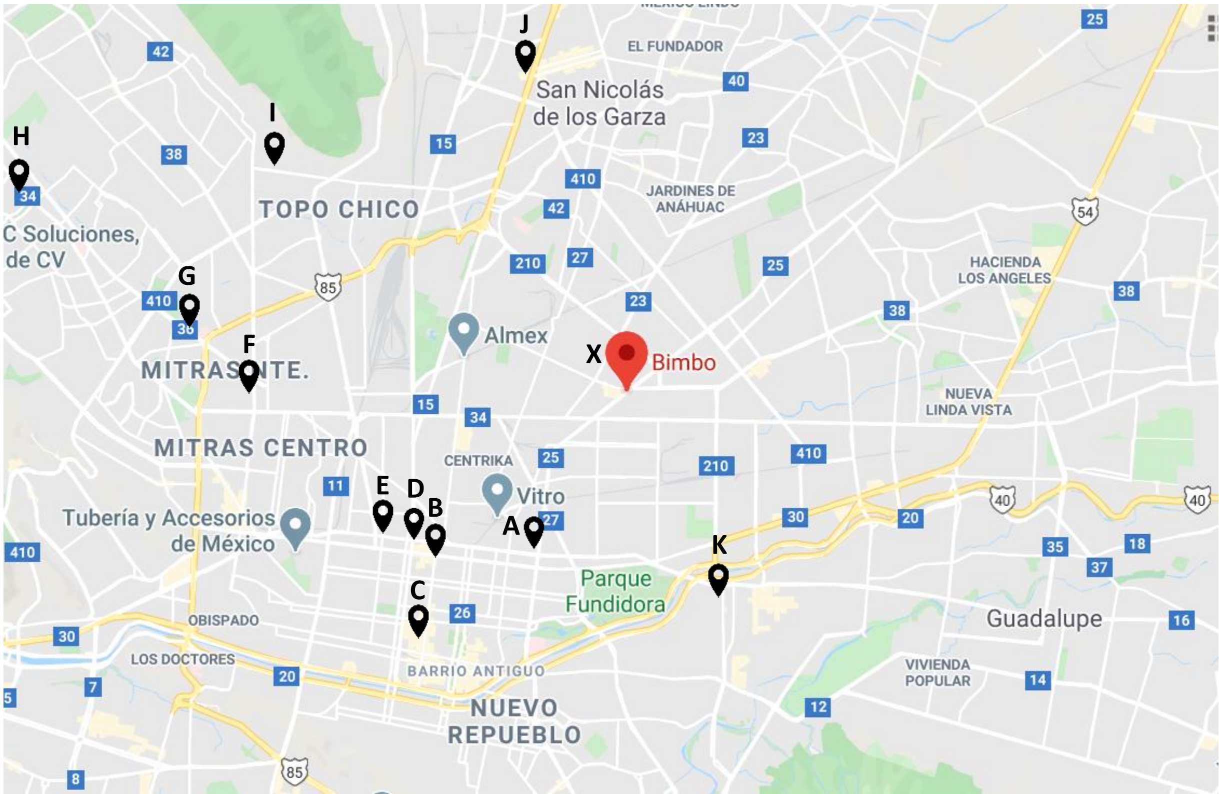


Figura 3. La figura muestra los expendios principales de Bimbo Monterrey Norte, además de la propia fábrica

Es importante notar que el tiempo para llegar al punto A del punto B, no necesariamente debe de ser el mismo para llegar al punto B del punto A, debido a leyes de tránsito, naturaleza de algunas calles, doble sentido, semáforos, etc. Sin embargo, se puede llegar a cualquier expendio desde cualquier otro. Por consiguiente:

$$d_e(A) = d_s(A) = 11$$

Cada vértice tendrá el mismo grado de entrada y de salida, el cuál será igual al número total de vértices menos 1, esto quiere decir que el número de arcos que saldrán y llegarán de cualquier vértice (eje. A) será 11.

Además:

$$\sum_{v \in V} d_e(v) = \sum_{v \in V} d_s(v) = card(E) = 121$$

La suma de todos los grados de entrada es igual a la suma de los grados de salida (fórmula anterior) la cuál es la misma que el número de arcos. Esto se debe a que cada arco suma uno más a la sumatoria de grado de entrada y otro de salida.

Por tanto G[E´] es el siguiente (fuente propia):

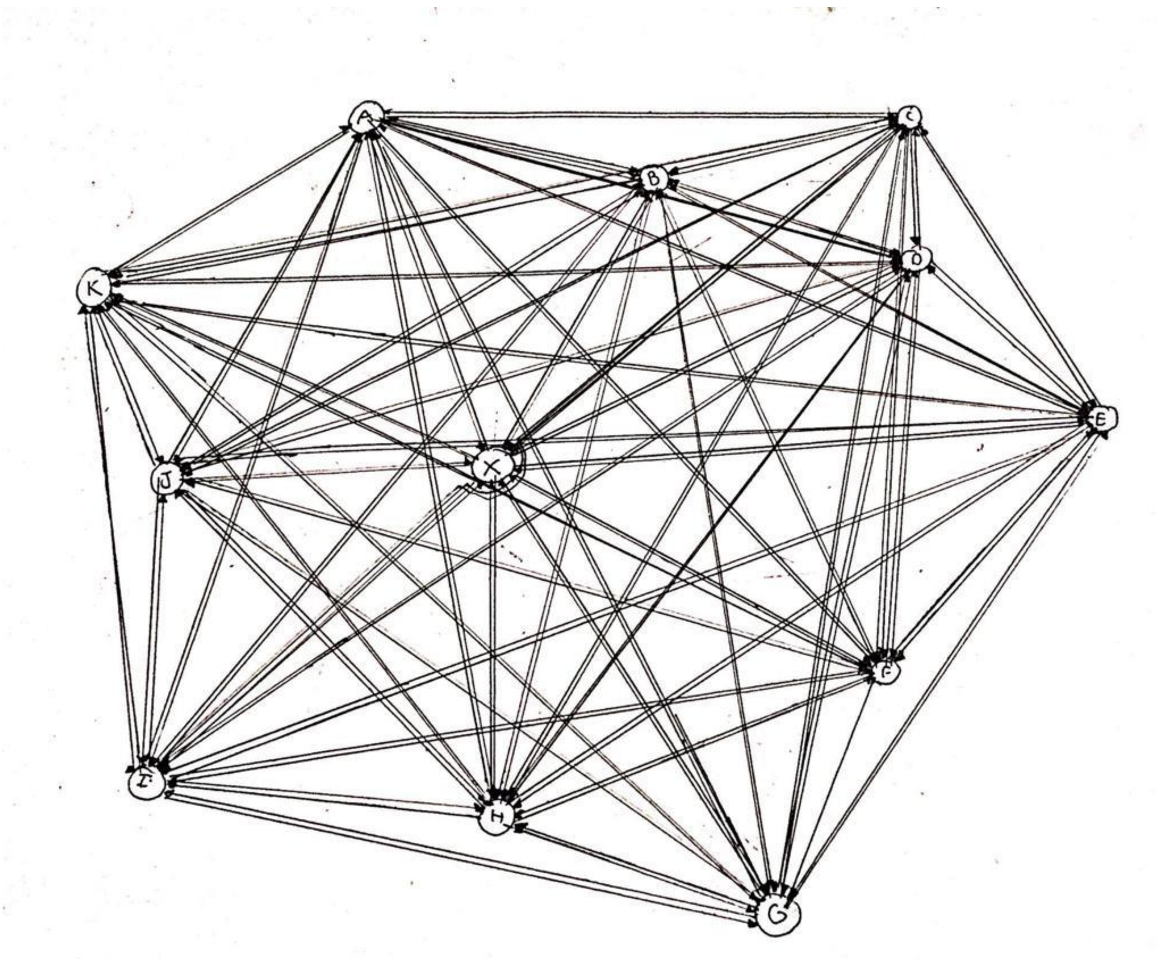


Figura 4. La figura muestra el subgrafo inducido de los principales expendios de Bimbo Monterrey Norte (Fuente Propia)

Se puede observar que hay muchas maneras de ir del nodo X al A, ya sea pasando por otros nodos o directamente, a estos se les llaman cadenas, sin embargo para la exploración se busca encontrar una manera de conectar todos los vértices, por lo cual se llamaría ciclo ya que es una cadena en la que no se debe de repetir ningún arco ni vértice (a excepción de X) con una longitud de cadena de 12, además de regresar a la fábrica después de efectuar este proceso. Por ejemplo el ciclo:

$$\{X, A, B, C, D, E, F, G, H, I, J, K, X\}$$

G[E´] también es fuertemente conexo, debido a que todos los nodos se encuentran en la misma clase de equivalencia, porque puedo llegar de cualquier expendio a cualquier otro y de manera contraria.

Grafo Hamiltoniano

Para que se pueda resolver el problema, primero se debe consolidar al G[E´] como hamiltoniano. Cualquier grafo es hamiltoniano si contiene un ciclo hamiltoniano, esto quiere decir que debe de contener un ciclo Y que comprenda todos los vértices del grafo sin repetir aristas ni vértices. Y en el G[E´] es cierto ya que cumple el teorema de Dirac:

$$G = (V, E), \qquad |V| = n \geq 3$$

$$\delta \geq \frac{n}{2} \Rightarrow hamiltoniano$$

Cualquier grafo no dirigido (nótese que G[E´] es dirigido, sin embargo hay dos arcos que conectan cualquiera dos nodos con dirección contraria, por tanto la unión de ambos arcos asemeja una arista) con un número mayor de 3 vértices es hamiltoniano si el grado mínimo es mayor o igual al número de vértices divididos entre dos. Por tanto:

$$\delta \geq \frac{n}{2}, \qquad 11 \geq \frac{12}{2} \quad \therefore \quad \mathrm{G[E']} \Rightarrow \mathrm{hamiltoniano}$$

El grado mínimo es 11 debido a que el vértice con menor grado de salida es 11.

Resolución (Monterrey)

Al corroborar que el subgrafo inducido es hamiltoniano, existen diversos ciclos que conectan todos los vértices y X es el nodo final e inicial. Sin embargo el número total de ciclos en esta distribución de productos es de:

$$11! = 39{,}916{,}800$$

Para encontrar el ciclo más rápido se utilizó un algoritmo en Java (Realizado por mí, ver Apéndice #1) que calcula el tiempo necesario para que un camión efectúe cada uno de estos trayectos (en el lapso de 9:00 a.m. a 10:00 a.m.) dados los siguientes datos (obtenidos de Google Maps):

(**Nota:** A cada arco se le asignó un valor en minutos)

Tabla 1. Tiempo de trayecto							
Arco	Tiempo (min)	Arco	Tiempo (min)	Arco	Tiempo (min)	Arco	Tiempo (min)
XA	12	CX	17	FX	17	IX	18
XB	15	CA	13	FA	18	IA	22
XC	18	CB	6	FB	16	IB	18
XD	18	CD	6	FC	20	IC	22
XE	20	CE	8	FD	14	ID	17
XF	17	CF	16	FE	14	IE	17
XG	16	CG	15	FG	6	IF	12
XH	26	CH	22	FH	13	IG	11
XI	19	CI	19	FI	12	IH	14
XJ	16	CJ	18	FJ	13	IJ	12
XK	13	CK	12	FK	19	IK	23
AX	10	DX	13	GX	18	JX	13
AB	5	DA	10	GA	22	JA	18
AC	9	DB	4	GB	20	JB	18
AD	6	DC	8	GC	25	JC	22
AE	8	DE	4	GD	20	JD	20
AF	17	DF	12	GE	21	JE	18
AG	18	DG	13	GF	8	JF	13
AH	25	DH	22	GH	10	JG	13
AI	20	DI	15	GI	8	JH	22
AJ	20	DJ	13	GJ	16	JI	12
AK	8	DK	13	GK	21	JK	17
BX	12	EX	16	HX	25	KX	13
BA	8	EA	14	HA	30	KA	13
BC	4	EB	7	HB	28	KB	14
BD	3	EC	11	HC	30	KC	18
BE	4	ED	5	HD	27	KD	14
BF	14	EF	10	HE	28	KE	17
BG	16	EG	13	HF	18	KF	22
BH	22	EH	19	HG	13	KG	21
BI	16	EI	12	HI	15	KH	31
BJ	16	EJ	16	HJ	22	KI	23
BK	9	EK	13	HK	25	KJ	20

Tabla 1. La table muestra el tiempo necesario para ir de un vértice a cualquier otro

El programa es iterativo dónde primeramente se permutan todas las combinaciones posibles de los vértices, y después se calcula el tiempo total que tarda cada ciclo. (**Nota:** Siempre se comienza y termina en X)

$$\sum_{i=1}^{12} V_i V_{i+1} = Tiempo_{total}$$

Por ejemplo, al permutar se obtiene la siguiente cadena:

$$\{XADGBCIJEFHKX\}$$

El programa divide la cadena en parejas adyacente y posteriormente calcula el total:

$$\{XA, AD, DG, GB, BC, CI, IJ, JE, EF, FH, HK, KX\}$$

$$Tiempo_{total} = XA + AD + DG + GB + BC + CI + IJ + JE + EF + FH + HK + KX$$

$$Tiempo_{total} = 12 + 6 + 13 + 20 + 4 + 19 + 12 + 18 + 10 + 13 + 25 + 13 = 165$$

Entonces a un camión le toma 165 minutos abastecer los expendios y regresar a la fábrica dado el ciclo anterior.

```
XKJIHGFECADBX  : 140
XKJIHGFECBADX  : 138
XKJIHGFECBDAX  : 131
XKJIHGFECDABX  : 138
XKJIHGFECDBAX  : 133
XKJIHGFEDABCX  : 135
XKJIHGFEDACBX  : 136
XKJIHGFEDBACX  : 137
XKJIHGFEDBCAX  : 130
XKJIHGFEDCABX  : 137
XKJIHGFEDCBAX  : 131
```

Figura 5. Ejemplo de algunas permutaciones de los expendios junto con su valor en minutos (Fuente propia)

Después de efectuar el algoritmo se encontró que el camino más rápido esta dado por el siguiente ciclo {XKABCDEFGHIJK} y le tomará alrededor de 111 minutos en efectuarlo más el tiempo de carga y descarga.

Mientras que el camino que demora más tiempo esta dado por el siguiente ciclo: {XEJDHAIKFBGCX} y le toma 247 minutos, más el tiempo de carga y descarga.

```
39916800
XKABCDEFGHIJX  : 111
XEJDHAIKFBGCX  : 247
BUILD SUCCESSFUL (total time: 80 minutes 8 seconds)
```

Figura 6. Impresión de pantalla del camino más rápido y corto, además del número de las permutaciones (Fuente Propia).

En un mapa se pueden visualizar de la siguiente manera:

Figura 7. Camino más rápido para la distribución (Fuente Propia).

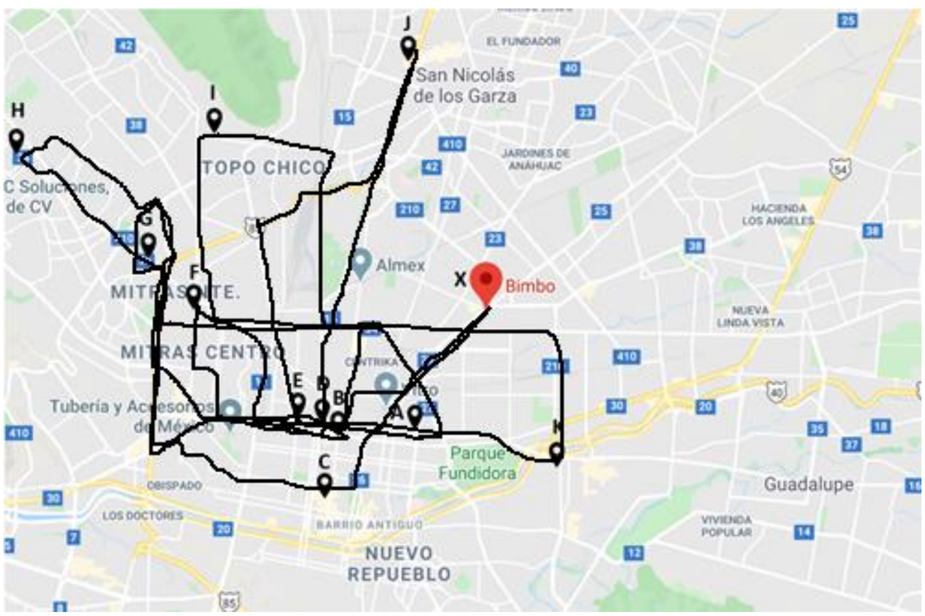


Figura 8. Camino con más demora para la distribución (Fuente Propia).

Modelación (Guadalajara-Monterrey)

Ahora bien, sucede algo distinto si no se tiene que pasar por todos los vértices y eso es lo que ocurre con el trayecto de la planta de Guadalajara a Monterrey, ya que sólo se necesita llegar de un lugar a otro.

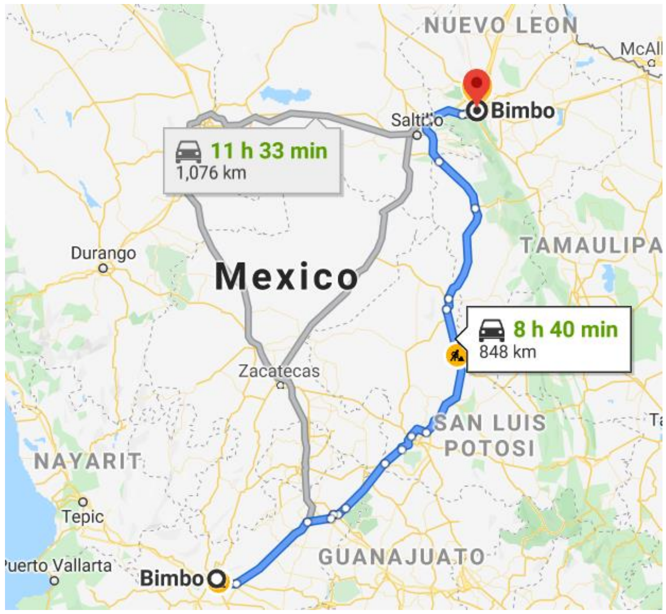


Figura 9. Ruta Bimbo Guadalajara-Bimbo Monterrey (Fuente: Google Maps)

En este caso no se analizará el camino más rápido, sino el más barato, tomando en cuenta el coste del combustible de un camión de 8 ejes (Modelo utilizado por Bimbo), además del gasto por cada caseta en el tramo estipulado. Esta estructura se modelaría a partir de un grafo ponderado en dónde se busca la ruta con menor peso del vértice Y (Bimbo Guadalajara) a X (Bimbo Norte Monterrey).

Un grafo ponderado es aquél en el que sus aristas/arcos tienen peso. El peso es un valor que indica el costo/distancia/tiempo que costará recorrer la sección entre ambos vértices. Es fundamental considerar que a diferencia de los grafos dirigidos, la suma de todos los grados de los vértices debe de ser igual al doble de la cantidad de vértices.

$$\sum_{v \in V} d(v) = 2|V|$$

Existen 9 segmentos que permiten la conexión de Y (Bimbo Guadalajara) y X (Bimbo Norte Monterrey), se encuentran enlistados a continuación:

XA. Guadalajara (Jalisco) – El Desperdicio (Jalisco)

AD. El Desperdicio (Jalisco) – Matehuala (San Luis Potosí)

AB. El Desperdicio (Jalisco) – Morelos (Zacatecas)

BC. Morelos (Zacatecas) – San Tiburcio (Zacatecas)

BE. Morelos (Zacatecas) – Saltillo (Coahuila)

CE. San Tiburcio (Zacatecas) – Saltillo (Coahuila)

DC. Matehuala (San Luis Potosí) – San Tiburcio (Zacatecas)

DE. Matehuala (San Luis Potosí) – Saltillo (Coahuila)

EY. Saltillo (Coahuila) – Monterrey (Nuevo León)

El costo del combustible de esta clase de camiones en Guadalajara es de $21.00 el litro (diésel) y aproximadamente utiliza 35 litros por cada 100 km (Telematics, 2019). Además, la información sobre el coste de las casetas y la distancia de las rutas fue recabada del sitio web: Ruta Punto a Punto que esta sustentada por la Secretaría de Comunicaciones y Transportes del gobierno mexicano. Por lo cual, para obtener el peso total de una arista se utiliza la siguiente ecuación:

$$Peso = Costo_{casetas} + \left(\frac{Distancia \times 35}{100} \times 21\right)$$

Utilizando la ecuación anterior se obtuvieron los siguientes resultados:

Tabla 2. Peso total de las aristas de G(V,E)				
Tramo	Costo de Casetas ($)	Kilómetros (km)	Costo de Combustible ($)	Peso Total
XA	1,005.00	149	1,095.15	2,100.15
AD	632.00	398	2,925.30	3,557.30
AB	1,476.00	378	2,778.30	4,254.30
BC	-	188	1,381.80	1,381.80
BE	-	350	2,572.50	2,572.50
CE	-	162	1,190.70	1,190.70
DC	-	124	911.40	911.40
DE	287.00	262	1,925.70	2,212.70
EY	-	70	514.50	514.50

Tabla 2. Se muestra el peso total que tendrá cada arista, basado en el costo del combustible y el de las casetas.

Por lo que el nuevo grafo se modelaría así:

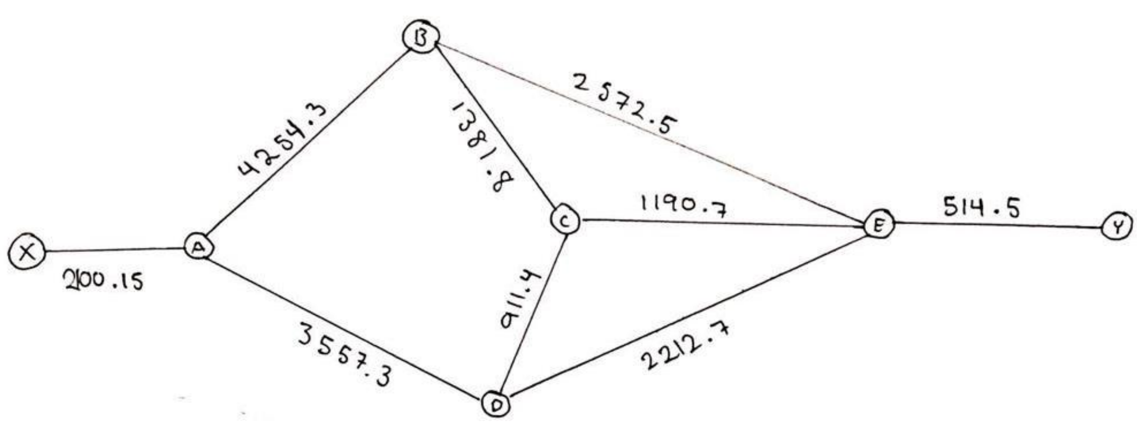


Figura 10. Grafo con pesos Guadalajara – Monterrey (Fuente Propia).

Resolución (Guadalajara-Monterrey)

Para encontrar el camino con menor peso se utilizará un algoritmo creado en 1959 por Edsger Dijkstra el cuál consiste en explorar los caminos con mínimo peso entre cualesquiera dos vértices (en este caso ambas fábricas).

Algoritmo de Dijkstra

El algoritmo funciona de la siguiente manera:

Teniendo dos vértices A y B, el camino más corto (menor peso) es aquél camino dónde la sumatoria de los pesos de las aristas que conectan ambos vértices es la menor de todos los caminos posibles.

El primer paso del algoritmo de Dijkstra es hacer una tabla en dónde podamos modelar los pasos del algoritmo.

Tabla 3. Algoritmo de Dijkstra		
Vértice	Menor Peso	Valor Temporal
X		
A		
B		
C		
D		
E		
Y		

Tabla 3. Se muestra la estructura del algoritmo de Dijkstra

Posteriormente se recorren todos los nodos adyacentes del nodo actual (se empieza en X), excepto los nodos no marcados, que son aquellos en los que no se ha encontrado si son los más eficientes. Creando un peso temporal dado por:

$$p(v_i) = p_{v_1} + p(v_1, v_i)$$

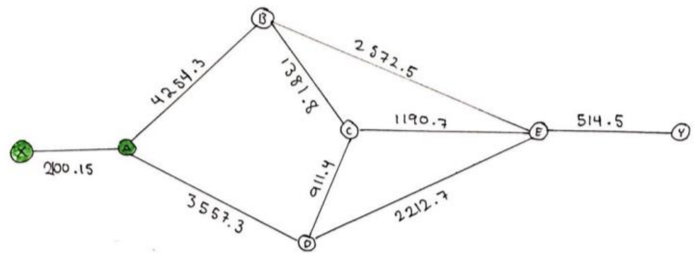
Dónde el peso temporal es igual al peso necesario para llegar a un vértice de un camino antes seleccionado, más el peso de la nueva arista.

Mientras que un camino especial es aquél que se traza a partir de los nodos ya marcados.

Cuando se realiza este proceso para el nodo actual, se avanza al nodo con menor peso encontrado y se realiza el mismo proceso, especificando todos los valores en la tabla. Por lo tanto para Bimbo se soluciona de la siguiente manera (tomando la Figura 10):

X sólo tiene cómo vértice adyacente a A, por tanto este valor temporal se convertirá en el menor peso, por tanto se queda marcado. Posteriormente el nodo A se convierte en el nodo actual y se analizarán sus nodos adyacentes

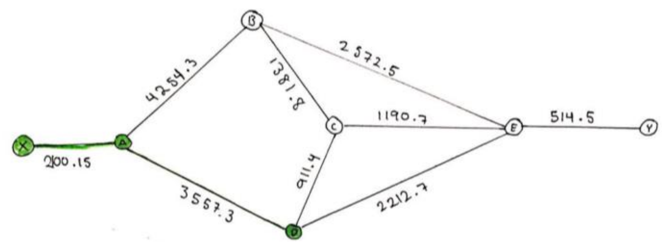
Tabla 3. Algoritmo de Dijsktra		
Vértice	Menor Peso	Valor Temporal
X	0	0
A		2100.15
B		
C		
D		
E		
Y		



A se queda marcado y tiene a B y D adyacentes, por tanto se calcula el valor temporal con la fórmula de arriba que es el peso acumulado.

Se puede observar que el peso más bajo es de A a D, por tanto este se marca.

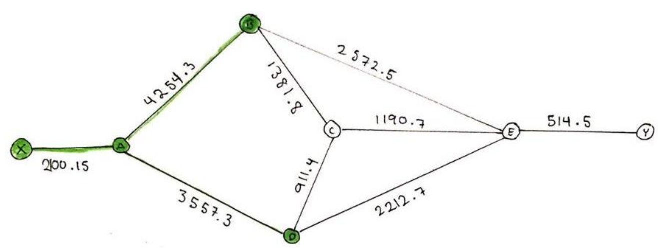
Tabla 3. Algoritmo de Dijsktra		
Vértice	Menor Peso	Valor Temporal
X	0	0
A	2100.15	2100.15
B		6354.45
C		
D		5657.45
E		
Y		



D tiene adyacencia con C y E, por tanto se calcula el peso acumulado en ambos.

Se observa que el mínimo peso es de B de 6354.45, por tanto se marca y se evalúa.

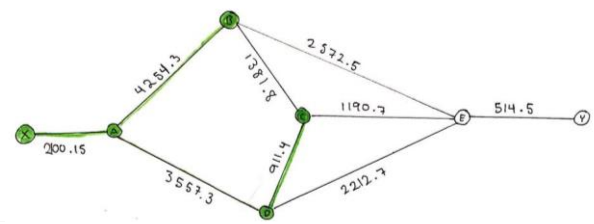
Tabla 3. Algoritmo de Dijsktra		
Vértice	Menor Peso	Valor Temporal
X	0	0
A	2100.15	2100.15
B		6354.45
C		6568.85
D	5657.45	5657.45
E		7870.15
Y		



Con el nodo actual en B se calcula el peso hacia C y E, sin embargo ya que es menor el valor temporal que se tenía de los pasos anteriores, se mantienen igual.

Ahora bien, el siguiente valor temporal a evaluar es el de C desde A

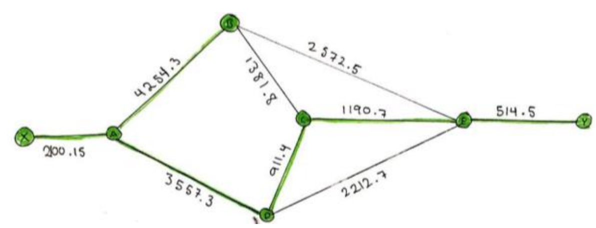
Tabla 3. Algoritmo de Dijsktra			
Vértice	Menor Peso	Valor Temporal	
X	0	0	
A	2100.15	2100.15	
B	6354.45	6354.45	
C		6568.85	7736.25
D	5657.45	5657.45	
E		7870.15	8926.95
Y			



Debido a que C adyacente con B da un valor temporal mayor al que se tenía, no se toma en cuenta. Sin embargo, la conexión al punto E resulta más eficiente, por tanto se actualiza.

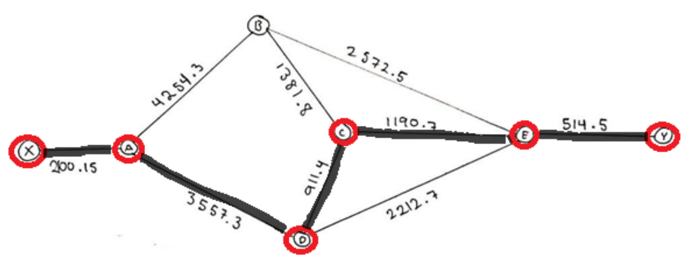
El siguiente valor para analizar es el de E, sin embargo, debido a que es el único nodo que tiene adyacencia con Y, ya se encontró el camino más corto para llegar de una fábrica a otra.

Tabla 3. Algoritmo de Dijsktra			
Vértice	Menor Peso	Valor Temporal	
X	0	0	
A	2100.15	2100.15	
B	6354.45	6354.45	7950.65
C	6568.85	6568.85	
D	5657.45	5657.45	
E		7870.15	7759.55
Y			



El camino más eficiente se forma a partir de {XADCEY} y se gastará un total de $8274.05 para completar el trayecto.

Tabla 3. Algoritmo de Dijsktra		
Vértice	Menor Peso	Valor Temporal
X	0	0
A	2100.15	2100.15
B	6354.45	6354.45
C	6568.85	6568.85
D	5657.45	5657.45
E	7759.55	7759.55
Y	8274.05	8274.05



Lo que se vería en un mapa de la siguiente manera:

Figura 11. Recorrido con menor gasto para un camión de 8 ejes de Bimbo Guadalajara a Bimbo Monterrey

Conclusión

En conclusión, se logró modelar la ruta óptima para poder surtir Roles Glaseados desde Bimbo Guadalajara a los diversos expendios de Bimbo Monterrey. El costo sería de $8274.05 para hacer llegar el producto a Monterrey y tomaría alrededor de 111 minutos distribuirlo a todos los expendios. Se puede observar la importancia en cuanto a eficiencia de utilizar la teoría de grafos para encontrar los caminos más cortos o con menor peso, y en caso de hacer un mal plan de logística existirían grandes pérdidas económicas. El teorema de Dirac y el algoritmo de Dijkstra resultan dos grandes herramientas para la mejora en la distribución de productos y fueron puestos a prueba en esta exploración.

Finalmente, considero que es interesante poder combinar la informática con las matemáticas, ya que con el software se pueden abstraer muchos elementos de la vida real y se pueden analizar en un plano más crítico. Resultó un gran reto poder modelar el problema en grafos y aún más crear un programa que me ayudara a hacerlo, ya que cómo no tenía muchos conocimientos en el área de Matemáticas Discretas, tuve que aprender la teoría desde cero y ponerla en práctica. Sin embargo, es enriquecedor saber cómo se llevan a cabo procesos de logística, y me llena de curiosidad conocer como funcionaría una red internacional y más aún el tema de redes neuronales.

Bibliografía

Castillo, H.C. (2016). Tecnológico Nacional de México: MODULO 7 DEL MIL MODULO DE LOGISTA INVERSA. Recuperado el 3 de marzo del 2020 de: https://dspace.itcolima.edu.mx/bitstream/handle/123456789/229/Tesis-Bimbo-TERMINADA%20(1).docxHUGO.pdf;jsessionid=bepvc16pohlw1j3kk4s6xzxtj?sequence=1

Secretaría de Comunicaciones y Transporte. (2020). Rutas Punto a Punto. Recuperado el 2 de marzo del 2020 de: http://app.sct.gob.mx/sibuac_internet/ControllerUI?action=cmdEscogeRuta

Arroyo J. A., Aguerreberre R., Torres G. (2012). SCT: COSTOS DE OPERACIÓN BASE DE LOS VEHÍCULOS REPRESENTATIVOS DEL TRANSPORTE INTERURBANO 2012. Recuperado el 3 de marzo del 2020 de: https://imt.mx/archivos/Publicaciones/PublicacionTecnica/pt368.pdf

Telematics, T. (2019). Webfleet: ¿Conoces el consumo de diésel de un camión por km? Recuperado el 3 de marzo del 2020 de: https://www.webfleet.com/es_es/webfleet/blog/conoces-el-consumo-de-diesel-de-un-camion-por-km/

Gobierno de México. (2020). Precios de gasolinas y diésel reportados por los permisionarios. Recuperado el 3 de marzo del 2020 de: http://www.cre.gob.mx/ConsultaPrecios/GasolinasyDiesel/GasolinasyDiesel.html

UDB (2019). Tema: Algoritmos para la ruta más corta en un

Grafo. Recuperado el 3 de marzo del 2020 de: http://www.udb.edu.sv/udb_files/recursos_guias/informatica-ingenieria/programacion-iv/2019/ii/guia-10.pdf

Universidad Pamplona (2012). Teoría de Grafos. Recuperado el 2 de marzo del 2020 de: http://www.unipamplona.edu.co/unipamplona/portalIG/home_23/recursos/general/11072012/grafo3.pdf

Álvarez, M., Parra, J (2013). Universidad del Bío-Bío: Teoría de Grafos. Recuperado el 2 de marzo del 2020 de: http://repobib.ubiobio.cl/jspui/bitstream/123456789/1953/3/Alvarez_Nunez_Marcelino.pdf

Jórdan C. (2011). Conexión en grafos dirigidos | 6/42 | UPV. Recuperado el 2 de marzo del 2020 de: https://www.youtube.com/watch?v=IgJ9_jrhhNI

Apéndices

Apéndices #1 (Programa en Java para el cálculo de cada camino)

```
public class Grafos {
    public int max;
    public int min;
    public String maxx;
    public String minn;
    public void main(String args[]){
        this.min=1000;
        this.max=100;
        String expendios []= {"A","B","C","D","E","F","G","H","I","J","K"};
        int permutaciones=0;
        for(int a=0; a<expendios.length; a++){
            for(int b=0; b<expendios.length; b++){
            for(int c=0; c<expendios.length; c++){
            for(int d=0; d<expendios.length; d++){
            for(int e=0; e<expendios.length; e++){
            for(int f=0; f<expendios.length; f++){
            for(int g=0; g<expendios.length; g++){
            for(int h=0; h<expendios.length; h++){
            for(int i=0; i<expendios.length; i++){
            for(int j=0; j<expendios.length; j++){
            for(int k=0; k<expendios.length; k++){

if(a!=b&&a!=c&&a!=d&&a!=e&&a!=f&&a!=g&&a!=h&&a!=i&&a!=j&&a!=k&&b!=a&&b!=c&&b!=d&&b!=e&&b!
=f&&b!=g&&b!=h&&b!=i&&b!=j&&b!=k&&c!=a&&c!=b&&c!=d&&c!=e&&c!=f&&c!=g&&c!=h&&c!=i&&c!=j&
&c!=k&&d!=a&&d!=b&&d!=c&&d!=e&&d!=f&&d!=g&&d!=h&&d!=i&&d!=j&&d!=k&&e!=a&&e!=b&&e!=c&&e!
=d&&e!=f&&e!=g&&e!=h&&e!=i&&e!=j&&e!=k&&f!=a&&f!=b&&f!=c&&f!=d&&f!=e&&f!=g&&f!=h&&f!=i&&f!
=j&&f!=k&&g!=a&&g!=b&&g!=c&&g!=d&&g!=e&&g!=f&&g!=h&&g!=i&&g!=j&&g!=k&&h!=a&&h!=b&&h!=c&
&h!=d&&h!=e&&h!=f&&h!=g&&h!=i&&h!=j&&h!=k&&i!=a&&i!=b&&i!=c&&i!=d&&i!=e&&i!=f&&i!=g&&i!=h&
&i!=j&&i!=k&&j!=a&&j!=b&&j!=c&&j!=d&&j!=e&&j!=f&&j!=g&&j!=h&&j!=i&&j!=k&&k!=a&&k!=b&&k!=c&&
k!=d&&k!=e&&k!=f&&k!=g&&k!=h&&k!=j&&k!=i){

permutaciones++;
                String camino = "X"+expendios[a]+expendios[b]+expendios[c]+expendios[d]+
                  expendios[e]+expendios[f]+expendios[g]+expendios[h]+
                        expendios[i]+expendios[j]+expendios[k]+"X ";
                sumarTiempo(camino);
```

```
            }}}}}}}}}}}}
System.out.println(permutaciones);
   }


   public void sumarTiempo(String h){
     int suma=0;
     for (int l=0; l<12; l++){
     if(h.substring(l, l+2)=="DB"||h.substring(l, l+2)=="DE"||h.substring(l, l+2)=="BC"||h.substring(l,l+2)=="BE"){
        suma=suma+4;
     }
     else if(h.substring(l, l+2)=="AB"||h.substring(l, l+2)=="ED"){
        suma=suma+5;
     }
     else if(h.substring(l, l+2)=="CB"||h.substring(l, l+2)=="CD"||h.substring(l, l+2)=="FG"||h.substring(l, l+2)=="AD"){
        suma=suma+6;
     }
     else if(h.substring(l, l+2)=="EB"){
        suma=suma+7;
     }
     else if(h.substring(l, l+2)=="CE"||h.substring(l, l+2)=="DC"||h.substring(l, l+2)=="AE"||h.substring(l,
l+2)=="GF"||h.substring(l, l+2)=="GI"||h.substring(l, l+2)=="AK"||h.substring(l, l+2)=="BA"){
        suma=suma+8;
     }
     else if(h.substring(l, l+2)=="AC"||h.substring(l, l+2)=="BK"){
        suma=suma+9;
     }
     else if(h.substring(l, l+2)=="AX"||h.substring(l, l+2)=="DA"||h.substring(l, l+2)=="GH"||h.substring(l, l+2)=="EF"){
        suma=suma+10;
     }
     else if(h.substring(l, l+2)=="IG"||h.substring(l, l+2)=="EC"){
        suma=suma+11;
     }
     else if(h.substring(l, l+2)=="XA"||h.substring(l, l+2)=="IF"||h.substring(l, l+2)=="FI"||h.substring(l,
l+2)=="IJ"||h.substring(l, l+2)=="CK"||h.substring(l, l+2)=="DF"||h.substring(l, l+2)=="JI"||h.substring(l,
l+2)=="BX"||h.substring(l, l+2)=="EI"){
        suma=suma+12;
     }
```

```
else if(h.substring(l, l+2)=="CA"||h.substring(l, l+2)=="FH"||h.substring(l, l+2)=="FJ"||h.substring(l, l+2)=="XK"||h.substring(l, l+2)=="DX"||h.substring(l, l+2)=="JX"||h.substring(l, l+2)=="DG"||h.substring(l, l+2)=="JF"||h.substring(l, l+2)=="JG"||h.substring(l, l+2)=="DJ"||h.substring(l, l+2)=="DK"||h.substring(l, l+2)=="KX"||h.substring(l, l+2)=="KA"||h.substring(l, l+2)=="EG"||h.substring(l, l+2)=="HG"||h.substring(l, l+2)=="EK"){

    suma=suma+13;

}

else if(h.substring(l, l+2)=="FD"||h.substring(l, l+2)=="FE"||h.substring(l, l+2)=="IH"||h.substring(l, l+2)=="EA"||h.substring(l, l+2)=="KB"||h.substring(l, l+2)=="KD"||h.substring(l, l+2)=="BF"){

    suma=suma+14;

}

else if(h.substring(l, l+2)=="XB"||h.substring(l, l+2)=="CG"||h.substring(l, l+2)=="DI"||h.substring(l, l+2)=="HI"){

    suma=suma+15;

}

 else if(h.substring(l, l+2)=="FB"||h.substring(l, l+2)=="CF"||h.substring(l, l+2)=="XG"||h.substring(l, l+2)=="XJ"||h.substring(l, l+2)=="GJ"||h.substring(l, l+2)=="EX"||h.substring(l, l+2)=="BG"||h.substring(l, l+2)=="BI"||h.substring(l, l+2)=="BJ"||h.substring(l, l+2)=="EJ"){

    suma=suma+16;

}

else if(h.substring(l, l+2)=="CX"||h.substring(l, l+2)=="FX"||h.substring(l, l+2)=="ID"||h.substring(l, l+2)=="XF"||h.substring(l, l+2)=="IE"||h.substring(l, l+2)=="AF"||h.substring(l, l+2)=="JK"||h.substring(l, l+2)=="KE"){

    suma=suma+17;

}

 else if(h.substring(l, l+2)=="IX"||h.substring(l, l+2)=="FA"||h.substring(l, l+2)=="XC"||h.substring(l, l+2)=="IB"||h.substring(l, l+2)=="XD"||h.substring(l, l+2)=="CJ"||h.substring(l, l+2)=="GX"||h.substring(l, l+2)=="JA"||h.substring(l, l+2)=="JB"||h.substring(l, l+2)=="JE"||h.substring(l, l+2)=="AG"||h.substring(l, l+2)=="KC"||h.substring(l, l+2)=="HF"){

    suma=suma+18;

}

else if(h.substring(l, l+2)=="XI"||h.substring(l, l+2)=="CI"||h.substring(l, l+2)=="FK"||h.substring(l, l+2)=="EH"){

    suma=suma+19;

}

else if(h.substring(l, l+2)=="FC"||h.substring(l, l+2)=="XE"||h.substring(l, l+2)=="GB"||h.substring(l, l+2)=="GD"||h.substring(l, l+2)=="JD"||h.substring(l, l+2)=="AI"||h.substring(l, l+2)=="AJ"||h.substring(l, l+2)=="KJ"){

    suma=suma+20;

}

else if(h.substring(l, l+2)=="GE"||h.substring(l, l+2)=="GK"||h.substring(l, l+2)=="KG"){

    suma=suma+21;

}
```

```
    else if(h.substring(l, l+2)=="IA"||h.substring(l, l+2)=="IC"||h.substring(l, l+2)=="CH"||h.substring(l,
l+2)=="GA"||h.substring(l, l+2)=="JC"||h.substring(l, l+2)=="DH"||h.substring(l, l+2)=="JH"||h.substring(l,
l+2)=="KF"||h.substring(l, l+2)=="BH"||h.substring(l, l+2)=="HJ"){
        suma=suma+22;
    }
     else if(h.substring(l, l+2)=="IK"||h.substring(l, l+2)=="KI"){
        suma=suma+23;
    }
      else if(h.substring(l, l+2)=="GC"||h.substring(l, l+2)=="AH"||h.substring(l, l+2)=="HX"||h.substring(l,
l+2)=="HK"){
        suma=suma+25;
    }
     else if(h.substring(l, l+2)=="XH"){
        suma=suma+26;
    }
    else if(h.substring(l, l+2)=="HD"){
        suma=suma+27;
    }
    else if(h.substring(l, l+2)=="HB"||h.substring(l, l+2)=="HE"){
        suma=suma+28;
    }
    else if(h.substring(l, l+2)=="HA"||h.substring(l, l+2)=="HC"){
        suma=suma+30;
    }
    else if(h.substring(l, l+2)=="KH"||h.substring(l, l+2)=="HE"){
        suma=suma+31;
    }
  }
  if(suma<this.min){
     this.minn=h;
     this.min=suma;
  }
  if(suma>this.max){
     this.maxx=h;
     this.max=suma;
  }
}}
```