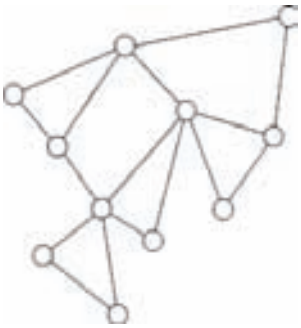
Implementación de un algoritmo para ubicar el centro de una red en problemas de localización

Dexmont Alejandro Peña Carrillo, Anel Berenice Reyes Ramírez, Ruth Marlen Ávila Guerrero, Roger Z. Ríos Mercado
Facultad de Ingeniería Mecánica y Eléctrica - UANL
{dexmont, anel, marlen, roger}@yalma.fime.uanl.mx



RESUMEN

La investigación de operaciones es la ciencia que brinda soporte a los problemas de toma de decisiones que surgen en diversos ámbitos. Una de las áreas de mayor interés, tanto práctico como científico, es la de los problemas de localización. Un problema típico tiene que ver con ubicar/construir instalaciones para brindar eficientemente un servicio. El problema se modela normalmente como una red, los nodos corresponden a clientes y/o puntos potenciales de ubicación de las instalaciones, y los arcos que unen a estos nodos corresponden a las relaciones entre dichos puntos. Este artículo trata el problema de cómo ubicar el centro absoluto en una red, el cual consiste en encontrar un punto en la red cuya distancia al nodo más lejano es mínima. Se realiza una implementación del algoritmo de Dvir y Handler, uno de los métodos existentes más eficientes para resolver este problema. Este trabajo ilustra la operación del algoritmo y muestra su eficiencia al resolver instancias de tamaño mayor a las reportadas originalmente por Dvir y Handler.

PALABRAS CLAVE

Investigación de operaciones, optimización de flujo en redes, localización, centro absoluto de una red.

ABSTRACT

Operations research is regarded as a scientific approach to decision-making problems arising in a number of scenarios. Location problems have become, in the practice and academically, one of the most interesting areas. One of the typical problems deals with locating or building facilities to efficiently deliver a service. The problem is modeled as a network, where nodes are associated to customers or potential facility location sites, and the arcs joining pair of nodes correspond to relationships between these nodes. This paper addresses the problem of locating the absolute center in a network, the point having a minimal distance to the farthest node in the network. An implementation of the Dvir-Handler algorithm, which is one of the most successful methods for this type of problems, is presented. The purpose of this work is both to illustrate how the method works and to corroborate its efficiency when tested on instances of considerably larger size than those reported by Dvir and Handler.

KEYWORDS

Operations research, network flow programming, location, absolute center of a network.

INTRODUCCIÓN

La investigación de operaciones es la ciencia que brinda soporte a problemas de toma de decisiones que surgen en diversos ámbitos industriales. En particular, una de la áreas de gran interés tanto práctico como científico es el de los problemas de localización.[1, 2] Un problema de localización típico tiene que ver con dónde ubicar/construir instalaciones para brindar un determinado tipo de servicio. El problema se modela normalmente como una red, los nodos corresponden a clientes y/o puntos potenciales de ubicación de las instalaciones, y los arcos que unen a estos nodos corresponden a las relaciones entre dichos puntos. Típicamente son relaciones de costo el cual se puede referir sin perdida de generalidad como distancia. En el problema de localización, resulta de marcada trascendencia saber encontrar en forma efectiva el centro de una red. Por centro nos referimos a un punto de la red en el cual la distancia a su nodo más alejado sea lo menor posible. El significado físico es que una instalación ubicada en este nodo centro tiene la característica que, al compararlo con cualquier otro nodo, la distancia recorrida al nodo más alejado siempre será la menor posible. Técnicamente, se dice que cumple con un criterio min-max, es decir, minimizar la peor de las distancias.

Entre las aplicaciones prácticas de este problema destacamos por ejemplo: localización de instalaciones de servicios de emergencia,[3] en donde se busca un punto para colocar un hospital de tal manera que los pacientes más lejanos recorran la menor distancia posible; localización de centros en problemas de diseño de territorios,[4, 5] en los cuales se desea ubicar centros para garantizar la compacidad de los territorios generados. Para un panorama más amplio sobre diversas aplicaciones de problemas de localización de centros de instalaciones como plantas industriales, bodegas o centros de servicios públicos en redes de transporte o de telecomunicaciones, véase, por ejemplo, Brandeau y Chiu,[6] Eiselt y Sandlom[1] o Mirchandani y Francis.[2]

El problema de ubicación del centro absoluto de una red fue introducido por Hakimi.[7] Uno de los métodos más recientes para encontrar el centro absoluto en una red es el de Dvir y Handler.[8] En este trabajo se lleva a cabo una implementación computacional del algoritmo de Dvir-Handler y se ilustra su utilidad en la solución de algunas instancias de red de tamaño mayor.

NOMENCLATURA Y CONCEPTOS BÁSICOS

En esta sección se presentan los conceptos básicos que serán utilizados a lo largo del presente trabajo. Como se expuso anteriormente, este trabajo se basa en el algoritmo desarrollado por Dvir y Handler.[8] La implementación de este algoritmo, que permite encontrar el centro absoluto de una red, necesita como entrada la matriz de distancias más cortas, la cual es una representación de las distancias más cortas entre cada uno de los nodos para un grafo dado. En este trabajo se utiliza el algoritmo Floyd-Warshall [9] para obtener esta matriz, y posteriormente se utiliza el algoritmo de Dvir-Handler para encontrar el centro absoluto de la red.

A continuación se reproducen algunos conceptos introducidos por Dvir y Handler[8] y algunos conceptos básicos de redes[10] con el fin de darle claridad al material presentado. Se denota al arco como la conexión del nodo v_i al nodo v_j y se representa como (i, j), el cual tiene asociada una distancia d_{ij} entre los nodos v_i, v_j respectivamente. Se define como un arco dirigido aquel arco en el cual el orden de los nodos es importante, es decir, el arco (i, j) es distinto al arco (j, i). Un arco no dirigido es aquel para el cual el orden de los nodos no importa, es decir, el arco (i, j) es igual que el arco (j, i). Se define una ruta como una secuencia de arcos ordenada, por ejemplo, (i, j), (j, k). Un ciclo se define como una ruta en la cual el primer nodo visitado es el último nodo de la ruta.

Sea $G=(V, A)$ un grafo compuesto por un conjunto de nodos $V=\{v_1, v_2, ..., v_n\}$ y un conjunto de arcos no dirigidos A con cardinalidad m. Sea $d(x,y)$ la distancia más corta entre dos puntos en la red $x, y \in G$, y $p(x,y)$ la trayectoria correspondiente a $d(x,y)$; $l(x)$ denota la distancia de un punto en la red $x \in G$ al nodo más lejano $y \in V$, esto es $l(x)=max\{d(x,y): x \in G, y \in V\}$. Se define ahora el centro nodos (VC) como un nodo que satisface l(VC) es el mínimo valor de $l(x)$ para todo nodo en la red $x \in V$, esto es l(VC)=min$\{l(x): x \in V\}$, además se define AC como un punto en la red $x \in G$ que satisface l(AC) es el mínimo valor de $l(x)$ para todo punto en la red $x \in G$, esto es l(AC)=min$\{l(x): x \in G\}$. En todos estos casos se utiliza la misma unidad de medición y se calculan de la misma manera.

El valor $r(G)=l(\text{AC})$ es conocido como el *radio* de G. El diámetro de un grafo se define como el máximo de las distancias de las rutas más cortas entre un par de nodos, esto es $\max\{l(x): x\in V\}$. Además se define el diámetro $d(G)$ de la red como dos veces su radio, llamados $d(G)=2r(G)$. En general se conoce que $r(G)<\max\ l(x)<d(G)$ y para un de árbol T (un árbol es un grafo en donde los nodos están conectados y no contiene ciclos), se puede ver que $2r(T)=\max\ l(x)=d(T)$. Supongamos ahora que T denota algún árbol de expansión de G (un árbol de expansión es un árbol que contiene todos los nodos del grafo). Luego, se define un mínimo diámetro de árbol (MDT) de G como un árbol de expansión de G, que satisface que $d(\text{MDT})$ es la distancia mínima de todos los árboles de expansión en G, esto es $d(\text{MDT})=\min\{d(T): T\in\Gamma(G)\}$, donde $\Gamma(G)$, es el conjunto de todos los árboles de expansión de G. Se define el árbol de trayectoria mínima enraizado a $x(\text{MDT}(x))$ como el árbol de expansión sobre las rutas más cortas de G desde un punto x en G.

Una alternativa para acelerar la solución del problema consistente en ubicar el centro absoluto de una red, ha sido encontrar cotas inferiores para el diámetro de algún arco. Una mejor cota, denominada $2l(x^*)\geq LB_{rs}$ en donde $\text{LB}_{rs}=d_{rs}+d(s,p)+d(r,q)$, llamada cota de Halpern, donde x^* es un centro local para el arco (r,s) y v_p,v_q son los nodos más lejanos de v_r, v_s respectivamente. El 95% de los arcos pueden ser eliminados comparando esta cota a una cota superior como $\text{UB}=2l(\text{VC})$.

La idea principal del algoritmo es tener un procedimiento iterativo exacto para identificar un MDT local asociado con un arco dado (r,s) de la red. Para el caso de una red cíclica, la cota de Halpern rinde una cota inferior para el diámetro de un árbol de expansión mínimo centrado en un arco particular (r,s) de la red. Comenzando con la cota de Halpern, el método genera una secuencia de cotas inferiores LB_{rs} hasta que el centro local es determinado (o hasta que $\text{LB}_{rs}\geq\text{UB}$.

FUNDAMENTOS TEÓRICOS

Los siguientes conceptos fueron tomados de[8] para ayudar a la comprensión y construcción del algoritmo el cual será presentado más adelante.

Suponga que un centro absoluto (AC) existe sobre un arco. La cota inferior de Halpern es la cota para el diámetro local de un grafo, utilizando esta definición se puede actualizar esta cota hasta que el diámetro del grafo es alcanzado, sin embargo, no conocemos el hecho de que un arco dado (r,s) incluye un AC, entonces hay dos posibilidades para abordar esto:

1) El arco contiene un AC. El diámetro local (diámetro del árbol de expansión) es el diámetro de G y el punto medio de la ruta diametral es un AC.
2) El arco no contiene un AC. Si se ha encontrado el diámetro de algún árbol de expansión de G, el cual es una cota superior del diámetro de G, entonces si se repite este procedimiento para todos los arcos (r,s) en A (conjunto de arcos en el grafo) se garantiza encontrar el diámetro de G como el mínimo de todos los diámetros locales.

A continuación se presenta un lema y tres teoremas que servirán para el procedimiento de diámetro local basado a arco (r,s), los cuales son tomados de[8] y pueden ser visualizados en la figura 1.

Para un arco dado (r,s) suponga que v_p y v_q son los nodos más lejanos a los nodos v_r y v_s respectivamente.

Lema: Si un AC es localizado sobre el interior de un arco (r,s) entonces la ruta más corta entre AC y v_p no puede ir a través de v_r .

Teorema 1: Si para un arco (r,s) existe v_p y v_q tales que son los mismos entonces el interior de este arco no puede contener un AC.

Teorema 2: (Cota de Halpern) Si un AC está localizado sobre un arco (r,s) entonces $d(G)\geq LB_{rs}$ en donde $LB_{rs}= d_{rs}+d(s,p)+d(r,q)$

donde:

$d(G)$ es el diámetro del grafo.

d_{rs} la longitud de la ruta más corta del nodo r al nodo s.

$d(s,p)$ la longitud de la ruta más corta de s a p.

$d(r,q)$ la longitud de la ruta más corta de r a q.

A continuación se presenta el teorema que nos permite actualizar LB_{rs} hasta que el diámetro actual es alcanzado, suponiendo que el AC existe en el interior del arco.

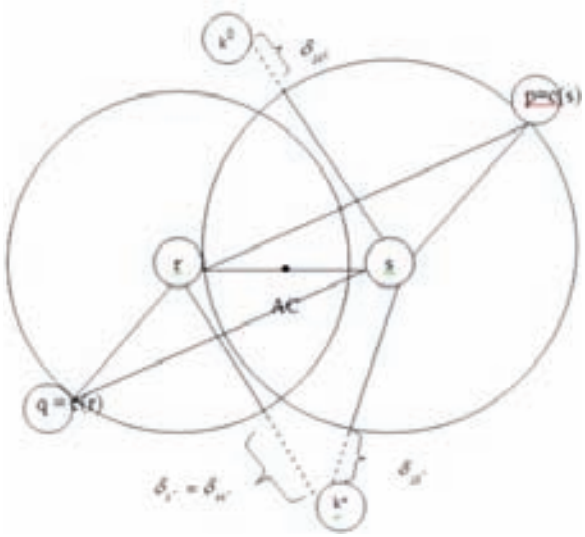


Fig. 1. Procedimiento de diámetro local basado a arco para el lema y los tres teoremas descritos.

En general, la idea básica es encontrar exitosamente un nuevo nodo, el cual es el más lejano que v_p y v_q de AC. Específicamente, se selecciona un nodo más lejano a AC en una dirección, digamos a través de v_r lo cual implica que estará conectado al AC (sobre la ruta mínima del árbol enraizado en AC) en la dirección opuesta v_s.

Ahora, suponiendo que $v_{c(r)}$ y $v_{c(s)}$ son el par de nodos actuales más cercanos conectados a AC vía v_r y v_s respectivamente. Esto significa que si hay un AC en el interior del arco considerado entonces hay una ruta más corta de AC a $v_{c(r)}$ que va a través de v_r, de la misma manera la ruta más corta de AC a $v_{c(s)}$ que va a través de v_s, entonces la cota inferior asociada con este par de nodos es $LB_{rs}= d_{rs}+d(r,c(r))+d(s,c(s))$.

Se supone que para un arco dado (r,s) y un nodo v_k se definen dos diferencias $\delta_{rk}=d(r,k)-d(r,c(r))$, $\delta_{sk}=d(s,k)-d(s,c(s))$. Posteriormente se define $\delta_{k^*}=\max\{\max\{\delta_{rk},\delta_{sk}\}:k\in K\}$ como la máxima diferencia, donde $k=\{k:\delta_{rk}>0,\delta_{sk}>0,k=1,2,...,n\}$ y si $K=\varnothing$ se define $\delta_{k^*=0}$.

Teorema 3: Suponga que un AC esta localizado en el interior de un arco (r,s) y suponga que se tiene un par actual de nodos más lejanos $v_{c(r)}$, $v_{c(s)}$, de un AC asociado a una cota inferior. Construyendo los pares de la diferencia para cada nodo v_k, se distinguen tres casos:

1) Si $\delta_{k^*}=0$ entonces el diámetro $d(G)=LB_{rs}$

2) Si $\delta_{k^*}=\delta_{rk^*}=\delta_{sk^*}>0$ entonces $d(G)=LB_{rs}+\delta_{k^*}$

3) Sin pérdida de generalidad $\delta_{k^*}=\delta_{rk^*}>\delta_{sk^*}>0$ entonces la ruta más corta $p(AC,k^*)$ va a través de v_s, así que $c(s)=k^*$ y $vc_{(r)},v_{k^*}$ es un par actualizado de los nodos más lejanos con respecto a el AC con cota inferior asociada a la cota inferior $LB_{rs}\leftarrow\delta_{sk^*}+LB_{rs}$.

DESCRIPCIÓN DEL ALGORITMO

El algoritmo de Dvir y Handler[8] para encontrar un centro absoluto de una red recibe como entrada la distancia de rutas más cortas D la cual es obtenida mediante el algoritmo Floyd-Warshall, y el conjunto de arcos A. Como salida muestra el diámetro $d(G)$ y un centro absoluto AC. El algoritmo utiliza los siguientes dos procedimientos:

El procedimiento configurarWT( ) (figura 2) se utiliza para guardar la fila r y s, de la matriz de distancias más cortas D. El pseudocódigo de este procedimiento es el siguiente:

```
Procedimiento configurarWT (r,s)
    hacer t(i,c):= d(i,c) para i=r,s y c=1,2,..., n.
```

Fig. 2. Pseudocódigo del procedimiento configurar WT( ).

El procedimiento actualizar (i,j,k) se encarga de modificar los valores en la tabla WT, restando el valor $t(j,k)$ de la tabla WT a la fila j, en caso de que no queden columnas positivas se hace UB=LBrs y CAC se localiza a (.5 UB)-$d(j,k)$ unidades de v_j en (i,j). El pseudocódigo de este procedimiento se encuentra en la figura 3.

```
Procedimiento actualizar (i,j,k)
    Restar t(j,k) de todas las entradas en la fila j
    Si no quedan columnas positivas en WT,
       entonces UB:=LB_rs y CAC se localiza a
       (.5 UB)-d(j,k) unidades de v_j en (i,j).
    Ir a Fin de ciclo.
```

Fig. 3. Pseudocódigo del procedimiento actualizar( ).

El algoritmo de Dvir-Handler (figura 4), consiste en lo siguiente: encontrar el nodo más lejano para cada uno de los nodos del grafo, posteriormente, encontrar el nodo con distancia mínima de los nodos más lejanos obtenidos anteriormente y definirlo como VC. Establecer la cota superior UB como dos veces la distancia de VC y suponer que el centro

```
Inicio
   Dado D, encontrar {k(i)}i=1,...,n, VC
   Hacer UB:=2l(VC), CAC:= VC
   Para todo (r,s) ∈A hacer
   Inicio
     Si k(r)=k(s) entonces ir a fin de ciclo
     Calcular LBrs:=drs+d(r,k(s))+d(s,k(r))
     Si LBrs≥UB entonces ir a fin de ciclo
     configurarWT (r,s)
     actualizar (s,r,k(s))
     actualizar (r,s,k(r))
   cotas:
        (suponga que ocurre una entrada máxima
        de WT en la fila i, i∈{r,s} y en la columna
        k, y sea j la otra fila).
        Buscar una entrada máxima t(i,k) en WT
        Calcular LBrs:=LBrs+t(j,k)
        Si LBrs≥UB entonces ir a fin de ciclo
        actualizar (i,j,k)
        Ir a cotas;
        Fin de ciclo:
Fin
d(G):=UB, AC:= CAC;
```

Fig. 4. Algoritmo de Dvir-Handler.

absoluto es VC, guardándolo en CAC la cual es la variable que contiene el centro absoluto actual.

Para todos los arcos del grafo si sus nodos más lejanos son iguales, se descarta que exista un centro absoluto en el arco, si son distintos se calcula la cota de Halpern como $LB_{rs}= d_{rs}+d(r,k(s))+d(s,k(r))$. Si la cota de Halpern es mayor o igual que la cota UB se descarta el arco, si es menor se utiliza el procedimiento configurarWT( ) para los nodos correspondientes al arco que se está analizando. Se utiliza el procedimiento actualizar( ) para cada uno de los nodos *r,s* con el nodo más lejano para cada uno de ellos respectivamente.

Posteriormente se busca el máximo valor en la tabla WT tomando en cuenta solamente las columnas estrictamente positivas, se asigna *i* a la fila que contiene este valor y *j* a la fila restante; se resta el valor máximo a todos los elementos de la fila *i*; se actualiza $LB_{rs}=LB_{rs}+ t(j,k)$. Si la cota LB_{rs} es mayor o igual que la cota superior UB entonces el ciclo termina, de lo contrario se utiliza el procedimiento actualizar (*i,j,k*) en la tabla WT y se repite este procedimiento. Finalmente hacer el diámetro de la red *d*(*G*)=UB y ubicar el centro absoluto AC en CAC.

Sea *k*(*i*) la columna en *D* que contiene el valor máximo para la fila *i*. Entonces, *k*(*i*) representa un nodo más lejano del nodo *i* y *l*(*i*)=*d*(*i*,*k*(*i*)). De entre todos los arcos se selecciona uno arbitrariamente. Sea UB la cota superior actual en *d*(*G*), y CAC, el candidato actual correspondiente a UB. Una inicialización natural es hacer CAC=VC y UB=2*l*(VC), donde *d*(VC,*k*(VC))=min{*d*(*i*,*k*(*i*)),*i*=1,2,...,*n*}. El algoritmo trabaja sobre un arco (*r*,*s*) dado. Los cálculos se realizan en una matriz WT_{2xn}, la cual inicialmente consiste en el par de filas *r,s* de *D*, esta es actualizada de manera iterativa. Cuando se completa el análisis del arco (*r*,*s*), se forma una nueva tabla para el siguiente arco a ser investigado.

Tal como se documentó en,[8] el algoritmo se justifica por los resultados de los teoremas 1, 2 y 3, reproducidos arriba. Por ejemplo el teorema 1 trata el caso donde v_p,v_q coinciden con *k*(*r*)=*k*(*s*). El teorema 2 establece la cota inferior inicial LB_{rs}, mientras el teorema 3 establece el procedimiento iterativo de actualización para LB_{rs}. En resumen, la inspección del arco (*r*,*s*) termina en una de tres maneras: cuando *k*(*r*)=*k*(*s*), $LB_{rs}\geq UB$, o WT no contiene más columnas positivas, en cuyo caso se mejora UB y CAC.

Complejidad computacional del algoritmo

El orden de complejidad del algoritmo para encontrar el centro absoluto de una red es 0 (mm^2). Sin embargo debido a que el algoritmo recibe como parámetro de entrada una matriz de rutas más cortas, ésta debe ser calculada, para realizar esta operación se utilizó el algoritmo Floyd-Warshall, el cual tiene una complejidad computacional de 0 (n^3), lo que produce un algoritmo de orden 0 (n^3+mm^2).

EJEMPLO ILUSTRATIVO

A continuación se ilustra el uso del algoritmo para encontrar el centro absoluto en una red con 11 nodos y 16 arcos, mostrados en la figura 5. La matriz de rutas más cortas está dada en la tabla I.

El algoritmo inicializa y supone que:

VC=4,*l*(VC)=*d*(4,10)=13,UB=26,CAC=VC.

El algoritmo descarta los arcos $(r,s)\in A$, si el nodo más lejano de *r* es el nodo más lejano de *s*, es decir, si *k*(*r*)=*k*(*s*), al eliminarlos el grafo resultante se muestra en la figura 6, por lo que la búsqueda se limita a los arcos restantes en el grafo.

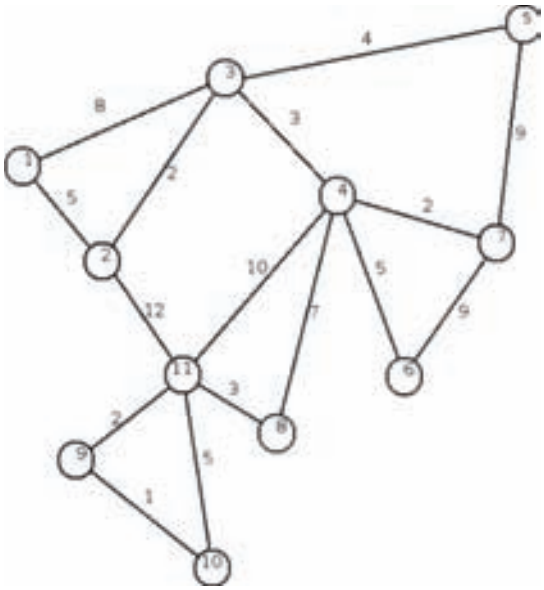


Fig. 5. Grafo original utilizado en el ejemplo.

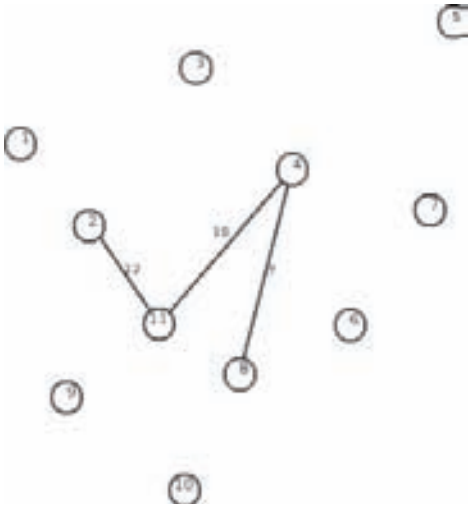


Fig. 6. Grafo resultante al descartar los arcos con $k(r)=k(s)$.

Al procesar el arco (2,11) se calcula $LB_{2,11}=12+5+3=20$, note que es menor que UB, por lo que este arco no se descarta, así que este arco necesita ser investigado. Se guarda WT con las filas 2 y 11 como se muestra en la tabla II.

Se realiza el procedimiento actualizar () dos veces: se resta $t(2,1)=5$ de la fila 2 y $t(11,10)=3$ de la fila 11. Las columnas que permanecen estrictamente positivas son mostradas en la tabla III.

El elemento mayor en la tabla III es 12, ahora se actualiza $LB_{2,11}=20+t(2,6)=25$ y nuevamente

Tabla I. Matriz D que contiene las rutas más cortas para el grafo utilizado en el ejemplo.

		Nodo destino											
		1	2	3	4	5	6	7	8	9	10	11	$k(i)$
Nodo origen	1	0	5	7	10	11	15	12	17	19	20	17	10
	2	5	0	2	5	6	10	7	12	14	15	12	10
	3	7	2	0	3	4	8	5	10	15	16	13	10
	4	10	5	3	0	3	5	2	7	12	13	10	10
	5	11	6	4	3	0	8	1	10	15	16	13	10
	6	15	10	8	5	8	0	7	12	17	18	15	10
	7	12	7	5	2	1	7	0	9	14	15	12	10
	8	17	12	10	7	10	12	9	0	5	6	3	1
	9	19	14	15	12	15	7	14	5	0	1	2	1
	10	20	15	16	13	16	18	15	6	1	0	3	1
	11	17	12	13	10	13	15	12	3	2	3	0	1

Tabla II. Tabla WT para las filas 2 y 11. Los valores de la tabla corresponden a la distancia de la ruta más corta entre los nodos origen y destino.

Nodo origen	Nodo destino										
	1	2	3	4	5	6	7	8	9	10	11
2	5	0	2	5	6	10	7	12	14	15	12
11	17	12	13	10	12	15	12	3	2	3	0

Tabla III. Columnas estrictamente positivas de la tabla WT para las filas 2 y 11.

	Nodo destino		
Nodo origen	5	6	7
2	1	5	2
11	10	12	9

se realiza el procedimiento actualizar( ) en WT, restando $t(2,6)=5$ de la fila 2. Ya que no quedan columnas positivas, como se muestra en la tabla IV, se actualiza el $UB=LB_{2,11}=25$ y CAC se localiza a $.5UB-d(2,6)=3$ unidades de v_2 en el arco (11, 2). Se continúa investigando en los dos arcos restantes y se tiene que $LB_{11,14}=10+3+10=23$ por lo que se tiene que revisar este arco ya que es menor que UB, se guarda WT con las filas 4 y 11 como se muestra en la tabla V.

Se realiza el procedimiento actualizar( ) dos veces lo que resta $t(11,10)=3$ de la fila 11 y $t(4,1)=10$ de la fila 4. Al realizar esto ya no hay columnas estrictamente positivas en WT por lo que se descarta este arco.

Tabla IV. Tabla resultante de aplicar el algoritmo en el arco (2,11).

	Nodo destino			
Nodo		5	6	7
	2	-4	0	-3
	11	10	12	9

Tabla V. Tabla WT para el arco (4,11). Los valores de la tabla corresponden a la distancia de la ruta más corta entre los nodos origen y destino.

	Nodo destino										
Nodo origen	1	2	3	4	5	6	7	8	9	10	11
4	10	5	3	0	3	5	2	7	12	13	10
11	17	12	13	14	13	15	12	3	2	3	0

Entonces $UB=LB_{11,14}=23$ y CAC se localiza a .5 UB-$d(4,1)$=1.5 unidades de v_4 en el arco (11,4). Enseguida se investiga el arco restante y se tiene que $LB_{4,8}=7+1065=23$, nótese que $LB \geq UB=23$ por lo que ya no se tiene que revisar este arco. Dado lo anterior se tiene que $d(G)$=UB=23 y AC=CAC.

EXPERIMENTACIÓN COMPUTACIONAL

El propósito de la experimentación reside por un lado en mostrar la aplicabilidad del método y mostrar además su utilidad al intentar resolver instancias de mayor tamaño a las reportadas por los autores en.[8] En dicho trabajo los autores manejan instancias de máximo 250 nodos, sin embargo, en la vida real pueden encontrarse aplicaciones como problemas de localización[5] en donde el tamaño de la red sea de orden mucho mayor.

Para realizar las pruebas del algoritmo se utilizaron instancias en las cuales los nodos representan coordenadas de ubicación de instalaciones y el valor de los arcos representa la distancia entre los nodos. Además se tiene la característica de que un nodo solo se puede comunicar con sus vecinos, es decir, con los nodos que se encuentran cercanos a él. Esta cercanía se definió como una zona en donde se encuentran los nodos vecinos. Los resultados reportados al correr las pruebas se encuentran en la tabla VI.

Nótese en la tabla VI, que en las redes generadas con 250 nodos, se eliminan alrededor de 200 arcos por el criterio de coincidencia, esto ocurre debido a que en las instancias utilizadas las distancias de los arcos corresponden a la distancia en el mapa entre los nodos del arco. Obsérvese que para las redes generadas con 1000 y 5000 nodos ocurre algo similar, los arcos eliminados por el criterio de coincidencia son alrededor de la mitad de los arcos de cada una de las redes utilizadas. De esta manera el algoritmo encuentra los centros absolutos en tiempo de ejecución del orden de los 16 segundos para redes con densidad de alrededor de 0.54 (la densidad se define como 2m / (n (n - 1)), donde *n* es la cantidad de nodos en el grafo y *m* es la cantidad de arcos).

Tabla VI. Resultados obtenidos de la implementación del algoritmo de Dvir-Handler y el algoritmo Floyd-Warshall.

	Nodos	250	1000	5000
	Redes	10	10	1
	Arcos	436-460	1774-1814	9024
Arcos eliminados	Coincidencia	217-229	886-906	4465
	Halpern´s Bound	0	0	46
Tiempo ejecución (seg.)	Floyd-Warshall	0 -1*	13-14	1669
	ACNet	0*	0 -1*	16
* 0 seg. se refiere a que el tiempo de ejecución es menor a un segundo.				

Las características del equipo en donde se realizaron las ejecuciones del algoritmo son las siguientes: Laptop Dell Inspiron con procesador Intel Centrino Duo de 1.66 GHz, 1 GB RAM, disco duro de 5400 rpm, utilizando como sistema operativo Windows XP. La implementación se realizó en lenguaje C con compilador GNU de Dev C++.

Cabe mencionar que estas pruebas se realizaron en instancias con mayor cantidad de nodos y arcos que las utilizadas por el autor del algoritmo, obteniendo resultados con tiempos de cómputo del mismo orden.

Las instancias fueron generadas de manera aleatoria garantizando que los grafos fueran planares, es decir, que los arcos no se intersectan, con el propósito de asemejar problemas de la vida real, en donde los nodos representan puntos en el mapa y los arcos alguna vía de comunicación. Otros problemas donde aparecen grafos planares en aplicaciones reales relevantes son los problemas de diseño de territorios comerciales,[4,5] diseño de territorios políticos[11] y modelos para la adquisición de terrenos contiguos,[12] por mencionar algunos.

CONCLUSIONES

En este trabajo se ha llevado a cabo una implementación computacional de un algoritmo propuesto en Dvir y Handler para encontrar el centro absoluto de una red, el cual es uno de los problemas importantes que surgen en el área de localización de instalaciones. Además, se realizó una implementación del algoritmo Floyd-Warshall para encontrar rutas más cortas entre parejas de nodos.

Se llevaron a cabo pruebas con redes de una mayor cantidad de nodos y arcos que las utilizadas por Dvir y Handler, obteniendo tiempos de cómputo del orden de los 16´s, con lo cual se ratifica la eficacia del método en instancias relativamente grandes.

AGRADECIMIENTOS

Se agradece al CONACYT, a la UANL y a la FIME el apoyo económico brindado a los becarios del Programa de Maestría en Ciencias en Ingeniería de Sistemas de la FIME-UANL.

REFERENCIAS

1. H. A. Eiselt & C.-L. Sandblom. Decision Analysis, Location Models, and Scheduling Problems. Springer, Berlín, Alemania, 2004.
2. P. B. Mirchandani & R. L. Francis, editores. Discrete Location Theory. Wiley, New York, EUA, 1990.
3. C. Toregas, R. Swain, C. ReVelle & L. Bergman. The location of emergency service facilities. Operations Research, 19(6):1363-1373, 1971.
4. S. I. Caballero-Hernández, R. Z. Ríos-Mercado, F. López y S. E. Schaeffer. Empirical evaluation of a metaheuristic for commercial territory design with joint assignment constraints. En J.E. Fernandez, S. Noriega, A. Mital, S.E. Butt y T.K. Fredericks, editores, Proceedings of the 12th Annual International Conference on Industrial Engineering Theory, Applications, and Practice, pp. 422-427. Cancún, México, Noviembre 2007.
5. R. Z. Ríos-Mercado & E. Fernández. A reactive GRASP for a commercial territory design problem with multiple balancing requirements. Computers & Operations Research. Aceptado, doi: 10.1016/j.cor.2007.10.024.
6. M. L. Brandeau & S. S. Chiu. An overview of representative problems in location research. Management Science, 35(6):645-674, 1989.
7. S. L. Hakimi. Optimum locations of switching centers and the absolute centers and medians of a graph. Operations Research, 12:450-459, 1964.
8. D. Dvir & G. Y. Handler. The absolute center of a network. Networks, 43(2):109-118, 2004.
9. R. W. Floyd. Algorithm 97: Shortest path. Comunications of ACM, 5:345, 1962.
10. R. Ahuja, T. Magnanti & J. Orlin. Network Flows. Prentice-Hall, Englewood Cliffs, EUA, 1992.
11. A. Mehrotra, E. L. Johnson & G. L. Nemhauser. An optimization based heuristic for political districting. Management Science, 44(8):1100-1114, 1998.
12. J. C. Williams. A zero-one programming model for contiguous land acquisition. Geographical Analysis, 34(4):330-349, 2002.