

UN MÉTODO COMPUTACIONAL PARA LA OBTENCIÓN DE RUTAS ÓPTIMAS EN SISTEMAS VIALES

A COMPUTATIONAL METHOD TO OBTAIN OPTIMAL PATHS IN ROAD NETWORKS

PEDRO LUIS ANGEL RESTREPO

Facultad de Ingeniería, Universidad de Antioquia, sede Medellín, plangel@udea.edu.co

LUIS FERNANDO MARÍN SEPULVEDA

Ingeniero de Sistemas, Facultad de Ingeniería, Universidad de Antioquia, sede Medellín, Luisferot@yahoo.com

Recibido para revisar diciembre 12 de 2010, aceptado abril 4 de 2011, versión final mayo 5 de 2011

RESUMEN: En este artículo se presenta un método basado en técnicas formales que permite determinar las rutas óptimas en un sistema vial. Se muestra como partiendo de la representación de una malla vial utilizando teoría de grafos y posteriormente aplicando una extensión del algoritmo de Floyd-Warshall, es posible obtener los valores de recorrido y la sucesión de segmentos que componen las rutas mínimas entre la totalidad de los puntos de interés del sistema.

PALABRAS CLAVE: Rutas optimas, Sistema vial, Algoritmo de Floyd-Warshall, Inteligencia Computacional.

ABSTRACT: This paper presents a method based on formal techniques for determining optimal routes in a road system. It is shown how a road system can be represented using graph theory; then by applying an extension of the Floyd Warshall algorithm, it is possible to obtain the value of each route and the succession of segments that make up the shortest routes between all crucial points of the system.

KEYWORDS: Optimal paths, Road networks, Floyd-Warshall Algorithm, Computational Intelligence.

1. INTRODUCCIÓN

El diseño y planeación de los trazados viales de las urbes modernas, requiere de la verificación de condiciones tales como determinar que las direcciones de las calles permitan un flujo vehicular máximo, o el garantizar que las distancias a recorrer sean equitativas para todos los usuarios. Igualmente, la utilización eficaz de una malla vial puede llegar a constituirse en un problema para el usuario que pretende determinar las rutas de transporte óptimas, ya sea por la complejidad del sistema o porque ocurran eventualidades tales como accidentes, la presencia de obras y/o de congestiones que dificultan e imposibilitan temporalmente la utilización de algunos de sus sectores.

El manejo de este tipo de situaciones cobra especial importancia cuando, por ejemplo, se presenta la necesidad de transportar de manera urgente un paciente a un centro de atención médica, haciéndose presente la necesidad de determinar la mejor ruta entre diferentes sitios de una ciudad, procurando que los costos asociados a los recorridos sean mínimos.

Esta problemática hace que sea deseable disponer de un método computacional para el cálculo de las rutas de recorrido mínimo en un sistema vial, que permita estudiar las características estructurales y dinámicas de las mallas viales, sin la necesidad de contar con software de costo elevado tanto desde el punto de vista computacional como económico.

En este contexto, este trabajo presenta un método para la determinación de rutas de valor de recorrido mínimo a través de los diferentes componentes de una malla vial, el cual permite obtener a partir de la representación del sistema por un grafo y la posterior aplicación de un algoritmo computacional, el valor de recorrido mínimo y la sucesión de los segmentos que deben ser transitados para que la ruta sea optima.

El artículo está organizado como sigue: en la segunda parte se introducen conceptos de la teoría de grafos, la tercera parte muestra como se representa una malla vial por medio de métodos derivados de la teoría de grafos, en la cuarta parte se introduce el algoritmo de Floyd-Warshall y se presenta una extensión al mismo propuesta por los autores, en la quinta parte se aplica el mencionado algoritmo a un modelo de un sistema

vial, obteniendo tanto las rutas mínimas como su costo de recorrido. Finalmente se presentan las conclusiones.

2. PRINCIPIOS MATEMÁTICOS DE LA TEORÍA DE GRAFOS

Un grafo se define como una estructura matemática compuesta por dos conjuntos, cuyos elementos están relacionados entre sí por una función de asociación; desde un punto de vista formal un grafo es un conjunto $G = \{V, A\}$ tal que los elementos de V se denominan nodos o vértices, mientras que los elementos de A se denominan arcos o aristas. La representación geométrica de un grafo se construye tradicionalmente a partir de círculos para los nodos y líneas para los arcos, cuando la relación definida por los arcos se da de un nodo a otro, de manera no recíproca, el arco se denomina dirigido y se representa por una línea terminada en una flecha que apunta hacia el nodo donde la relación se satisface.

Los arcos tienen una función que les asocia un valor conocido con el nombre de peso, si todos los arcos de un grafo tienen el mismo peso usualmente se omiten en la representación geométrica, también se acostumbra no colocar el peso cuando el valor es unitario. La figura 1 ilustra geoméricamente un grafo.

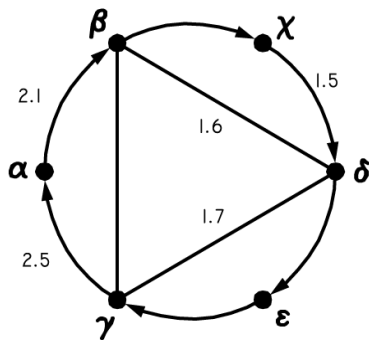


Figura 1. Representación geométrica de un grafo

Figure 1. Pictorial representation of a graph

El uso computacional de la información almacenada en un grafo se realiza usualmente a través de la representación matricial del mismo [1]. Dicha representación matricial se lleva a cabo fundamentalmente por medio de dos matrices: La matriz de Incidencia que relaciona los arcos con los nodos, de tal manera que la entrada de la matriz correspondiente a la intersección de la columna que representa el arco con la fila que representa el nodo,

contiene el valor del peso del arco. Mientras que la matriz de Adyacencia relaciona los nodos entre sí, de tal manera que si existe un arco entre dos nodos la entrada correspondiente a la columna y fila que representan en su orden los nodos inicial y final tiene un valor de uno (1). Si no existe arco entre los nodos el valor es cero (0).

Es importante tener en cuenta que los índices de la matriz representan en su orden la columna y la fila, esta distinción es importante ya que los arcos pueden ser dirigidos y por lo tanto al existir un arco de este tipo entre los nodos N_1 y N_2 , en la posición correspondiente de la matriz existe un valor diferente de cero (0) mientras que entre N_2 y N_1 no [2]. Las matrices de incidencia y adyacencia del grafo ilustrado en la figura 1 se presentan en la figura 2.

	α	β	χ	δ	ε	γ
α	0	0	0	0	0	1
β	1	0	0	1	0	1
χ	0	1	0	0	0	0
δ	0	1	1	0	0	1
ε	0	0	0	1	0	0
γ	0	1	0	1	1	0

2. a Matriz de adyacencia

2. a Adjacency matrix

	$\alpha\beta$	$\beta\chi$	$\chi\delta$	$\delta\varepsilon$	$\varepsilon\gamma$	$\gamma\alpha$	$\beta\delta$	$\delta\gamma$	$\gamma\beta$
α	0.0	0.0	0.0	0.0	0.0	2.5	0.0	0.0	0.0
β	2.1	0.0	0.0	0.0	0.0	0.0	1.6	0.0	1.0
χ	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
δ	0.0	0.0	1.5	0.0	0.0	0.0	1.6	1.7	0.0
ε	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
γ	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.7	1.0

2. b Matriz de incidencia.

2. b Incidence matrix.

Figura 2. Representación matricial del grafo de la figura 1.

Figure 2. Matrix representation of graph in figure 1.

3. REPRESENTACION DE UN SISTEMA VIAL POR MEDIO DE GRAFOS.

Un sistema vial es susceptible de ser modelado mediante teoría de grafos [3], [4], [5] y [6], de hecho la forma mas adecuada de representar una malla vial es por su representación por grafos [7], esta forma de especificación ha sido incorporada de manera sistemática en las herramientas computacionales utilizadas en el estudio de las mallas viales. Una descripción detallada de esta técnica y su transformación a estructuras de

datos, tan relevantes a nivel mundial como el modelo ARC/Info de la ESRI, es desarrollada en [8].

En términos generales, para construir una representación de una malla vial usando grafos, el trazado de las calles es representado por arcos y sus intersecciones por nodos, la distancia medida en las intersecciones se representa en el peso del arco correspondiente. En caso de que la calle sea en un solo sentido el arco es dirigido, en caso contrario no lo es [9]. Aplicando este método de representación a un sector de la malla vial de la ciudad de Medellín, Colombia, presentado en la figura 3, se obtiene el grafo (sobrepuesto al sector vial), mostrado en la figura 4.

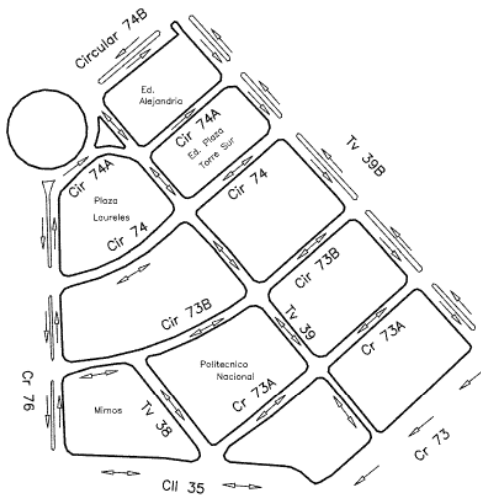


Figura 3. Sector vial objeto de estudio.
Figure 3. Road network sector under study

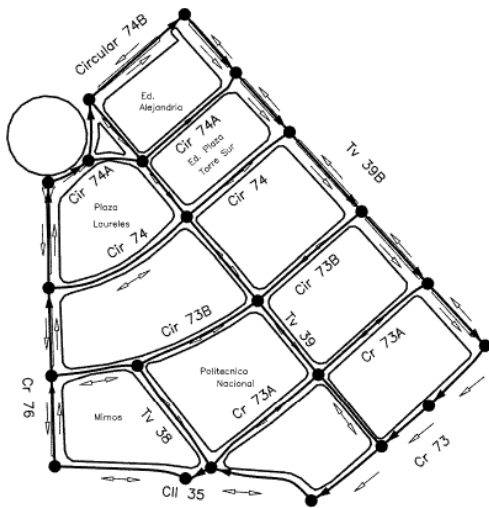


Figura 4. Modelo basado en grafos del sector vial objeto de estudio.
Figure 4. Model based on graphs of the road network sector under study.

A partir del grafo obtenido, y utilizando la nomenclatura ilustrada en la figura 5, se obtienen las matrices de adyacencia e incidencia presentadas en las figuras 6 y 7. En ellas queda registrada la longitud de las distancias entre los puntos de interés (intersecciones) y sus posibles conexiones (calles).

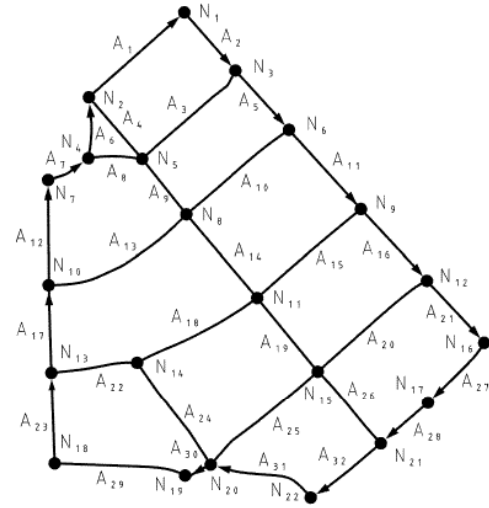


Figura 5. Grafo del sector vial objeto de estudio.
Figure 5. Graph of road network sector under study.

	N ₁	N ₂	N ₃	N ₄	N ₅	N ₆	N ₇	N ₈	N ₉	N ₁₀	N ₁₁	N ₁₂	N ₁₃	N ₁₄	N ₁₅	N ₁₆	N ₁₇	N ₁₈	N ₁₉	N ₂₀	N ₂₁	N ₂₂
N ₁	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N ₂	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N ₃	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N ₄	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N ₅	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N ₆	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N ₇	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N ₈	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N ₉	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
N ₁₀	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
N ₁₁	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
N ₁₂	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
N ₁₃	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
N ₁₄	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
N ₁₅	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
N ₁₆	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
N ₁₇	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
N ₁₈	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
N ₁₉	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
N ₂₀	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
N ₂₁	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
N ₂₂	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Figura 6. Matriz de Adyacencia del grafo modelo.
Figure 6. Adjacency matrix of graph under study.

A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}	A_{18}	A_{19}	A_{20}	A_{21}	A_{22}	A_{23}
N_1	106.72	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
N_2	0.0	64.77	108.36	0.0	51.29	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
N_3	0.0	0.0	68.12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
N_4	0.0	0.0	0.0	0.0	0.0	38.59	45.24	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
N_5	0.0	0.0	0.0	0.0	0.0	0.0	45.24	59.28	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
N_6	0.0	0.0	0.0	0.0	68.83	0.0	0.0	0.0	111.23	0.0	87.72	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
N_7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	59.28	111.23	0.0	0.0	133.36	91.47	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
N_8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	89.48	0.0	0.0	0.0	114.16	0.0	73.31	0.0	0.0	0.0	0.0	0.0	...
N_9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	133.36	91.47	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
N_{10}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
N_{11}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
N_{12}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
N_{13}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
N_{14}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
N_{15}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
N_{16}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
N_{17}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
N_{18}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
N_{19}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
N_{20}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
N_{21}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
N_{22}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...

7. a Matriz de incidencia arcos A_1 a A_{23} .7. a Incidence matrix arcs A_1 to A_{23} .

A_{24}	A_{25}	A_{26}	A_{27}	A_{28}	A_{29}	A_{30}	A_{31}	A_{32}
N_1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
N_2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
N_3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
N_4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
N_5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
N_6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
N_7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
N_8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
N_9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
N_{10}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
N_{11}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
N_{12}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
N_{13}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
N_{14}	105.63	0.0	0.0	0.0	0.0	0.0	0.0	0.0
N_{15}	0.0	119.14	80.26	0.0	0.0	0.0	0.0	0.0
N_{16}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
N_{17}	0.0	0.0	0.0	69.07	0.0	0.0	0.0	0.0
N_{18}	0.0	0.0	0.0	0.0	110.22	23.45	0.0	0.0
N_{19}	0.0	0.0	0.0	0.0	0.0	0.0	91.23	0.0
N_{20}	105.63	119.14	0.0	0.0	0.0	0.0	0.0	0.0
N_{21}	0.0	0.0	80.26	0.0	52.24	0.0	0.0	0.0
N_{22}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	74.27

7. b Matriz de incidencia arcos A_{24} a A_{32} 7. b Incidence matrix arcs A_{24} to A_{32} .

Figura 7. Matriz de incidencia del sector vial objeto de estudio.

Figure 7. Incidence matrix of road network sector under study

4. PROPUESTA DE EXTENSION AL ALGORITMO DE FLOYD-WARSHALL

La combinación de los métodos que involucran la representación de sistemas viales y algoritmos computacionales para su diseño y el estudio de su desempeño, han sido aplicados en tareas tan diversas como el flujo máximo de tráfico o el diseño de rutas de logística [3], [10], pero es aun necesario el desarrollo de herramientas que permitan evaluar mallas viales [11].

En este sentido, el problema de procesar la información almacenada en grafos ha sido ampliamente tratado en la literatura técnica, por ejemplo enfoques jerárquicos se presentan en [12], [13] y [6], algoritmos adicionales que buscan procesamiento eficiente se presentan en [14], un método para utilizar programación dinámica se plantea en [15], igualmente en [16] se presenta la utilización de búsqueda distribuida y en paralelo al problema de encontrar rutas mínimas.

Si bien existen diversos algoritmos para encontrar los valores de recorrido mínimo en un grafo,

tales como el Dijkstra, el Bellman–Ford, el Johnson [17] o el Suurballe [18]; estos se limitan a grafos dirigidos, o a determinar la distancia entre algunos vértices, mientras que el algoritmo de Floyd-Warshall encuentra los valores de las rutas mínimas entre la totalidad de los nodos de un grafo conexo que este compuesto por arcos dirigidos, no dirigidos o mixtos, con pesos positivos [2], [19].

El algoritmo de Floyd-Warshall, o simplemente algoritmo de Warshall, se basa en iteraciones sobre una matriz inicial W^0 , que se construye a partir de las matrices de adyacencia e incidencia del grafo utilizando la ecuación 1. Sobre dicha matriz se realiza un número de iteraciones igual al número de nodos del grafo $(|V|)$ aplicando la ecuación 2. La figura 8 presenta el algoritmo de Warshall.

$$W^0 = \begin{cases} w_{ij} & \text{si } Adj_{ij} > 0 \\ \infty & \text{si } Adj_{ij} = 0 \end{cases} \quad (1)$$

$$W^m = \min(w_{ij}^{m-1}, w_{im}^{m-1} + w_{mj}^{m-1}) \quad (2)$$

Algoritmo de Warshall

Entrada: Matriz $W^0 = [M, M]$

donde: $M = |V|$ cardinalidad del conjunto de vertices del grafo que representa el sistema

Retorno: Matriz de Valores de Rutas Mínimas : W^{M+1}

Inicio

// Procesamiento de las matrices

Para $h = 1$ *hasta* M **hacer**

Para $i = 1$ *hasta* M **hacer**

Para $j = 1$ *hasta* M **hacer**

Si $w_{[i,j]}^{h-1} < w_{[i,h]}^{h-1} + w_{[h,j]}^{h-1}$ **Entonces:**

$w_{[i,j]}^h = w_{[i,h]}^{h-1} + w_{[h,j]}^{h-1}$

Si no:

$w_{[i,j]}^h = w_{[i,j]}^{h-1}$

// Postratamiento de la matriz de valores de rutas

Para $i = 1$ *hasta* M **hacer**

$w_{[i,i]}^{M+1} = 0$

Fin

El algoritmo arroja como resultado una matriz de valores de rutas (W^{m+1}) , la cual almacena en la intersección correspondiente entre el nodo origen y el nodo destino, el valor de la ruta mínima entre ellos. Determinando de esta manera los valores de las rutas mínimas absolutas entre la totalidad de los nodos del grafo, sin embargo es insuficiente por si mismo para establecer las rutas a recorrer que se corresponden con dichos valores.

Para obtener la especificación de los caminos óptimos entre los puntos de interés del sistema, en este trabajo los autores presentan una ampliación al algoritmo de Warshall que arroja como resultado una matriz de rutas en la cual se almacenan como una sucesión de vértices los recorridos correspondientes a las rutas de costos mínimos identificados en la matriz de valores. El funcionamiento del algoritmo extendido se basa en una matriz Θ^0 , construida a partir de la ecuación 3, que es utilizada como punto de partida para registrar los cambios en cada iteración del algoritmo de Floyd-Warshall, hasta lograr la construcción de la matriz Θ^m , necesaria para obtener la matriz de rutas.

$$\Theta_0 = \begin{cases} I & \text{si } \overrightarrow{IJ} > 0 \\ " & \text{si } \overrightarrow{IJ} = 0 \end{cases} \quad (3)$$

La figura 9 presenta del algoritmo de Floyd-Warshall extendido, mientras que la figura 10 presenta el algoritmo para la obtención de la matriz de rutas a partir de la matriz Θ^m .

5. DETERMINACIÓN DE RUTAS ÓPTIMAS EN UN SISTEMA VIAL.

Para encontrar las rutas de valor de recorrido mínimo para el sector vial presentado en la figura 3, cuyo modelo por grafos se muestra esquemáticamente en la figura 5 y matricialmente en las figuras 6 y 7; se procede a utilizar la extensión al algoritmo de Warshall. Las matrices iniciales: W^0 y Θ^0 se muestran en las figuras 11 y 12 respectivamente, estas matrices constituyen las estructuras de datos de entrada del algoritmo extendido.

Figura 8. Algoritmo de Floyd-Warshall.

Figure 8. Floyd-Warshall algorithm.

```

Algoritmo Extendido de Warshall

Entrada: Matriz  $W^0 = [M, M]$ 
          Matriz  $\Theta^0 = [M, M]$ 

donde:  $M = |V|$  cardinalidad del conjunto de vertices
          del grafo que representa el sistema

Retorno: Matriz de Valores de Rutas Míminas :  $W^{M+1}$ 
          Matriz para obtención de Rutas Míminas :  $\Theta^M$ 

Inicio

    // Procesamiento de las matrices
    Para  $h = 1$  hasta  $M$  hacer
        Para  $i = 1$  hasta  $M$  hacer
            Para  $j = 1$  hasta  $M$  hacer
                Si  $w_{[i,j]}^{h-1} < w_{[i,h]}^{h-1} + w_{[h,j]}^{h-1}$  Entonces:
                     $w_{[i,j]}^h = w_{[i,h]}^{h-1} + w_{[h,j]}^{h-1}$ 
                    // Cambio en la matriz  $\Theta_m$ 
                     $\Theta_{i,j} = h$  -ésimo Nodo
                Si no:
                     $w_{[i,j]}^h = w_{[i,j]}^{h-1}$ 
            [
                [
                    [
                        // Postratamiento de la matriz de valores de rutas
                        Para  $i = 1$  hasta  $M$  hacer
                             $w_{[i,i]}^{M+1} = 0$ 
                    ]
                ]
            ]

Fin

```

Figura 9. Algoritmo extendido de Floyd-Warshall.

Figure 9. Floyd-Warshall extended algorithm.

Entrada: Matriz $\Theta^M = [M, M]$

donde: $M = |V|$ cardinalidad del conjunto de vertices del grafo que representa el sistema

Retorno: Matriz de Rutas M nimas : Θ^{M+1}

Inicio

```
// Procesar todas las entradas de la Matriz
Para i = 1 hasta M hacer
    Para j = 1 hasta M hacer
        Si  $\theta_{[i,j]}^M \neq "$  Entonces:
            VariableAuxiliar = i-esimo Nodo + ? +  $\theta_{[i,j]}^M$  +
                                ? + j-esimo Nodo
            Mientras VariableAuxiliar contenga una
                interrogante ( ? ) haga
                    P = Posici n primer interrogante (?)
                    Si  $\theta_{[Nodo[P-1],Nodo[P+1]]}^M = Nodo[P-1]$  O
                         $\theta_{[Nodo[P-1],Nodo[P+1]]}^M = Nodo[P+1]$  Entonces:
                            Eliminar el interrogante (?) en la posici n P
                    Si no:
                        Reemplazar el interrogante (?) por:
                            ? +  $\theta_{[Nodo[P-1],Nodo[P+1]]}^M$  + ?
            Si no:
                 $\theta_{[i,j]}^{M+1} = "$ 
                 $\theta_{[i,j]}^{M+1} = VariableAuxiliar$ 
```

Fin

Figura 10. Algoritmo para la obtención de la matriz de rutas mínimas Θ^{M+1} en un grafo.

Figure 10. Algorithm to obtain the shortest routes matrix Θ^{M+1} in a graph.

[illegible]

Figura 11. Matriz W^0 del grafo objeto de estudio.

Figure 11. W^0 Matrix of graph under study.

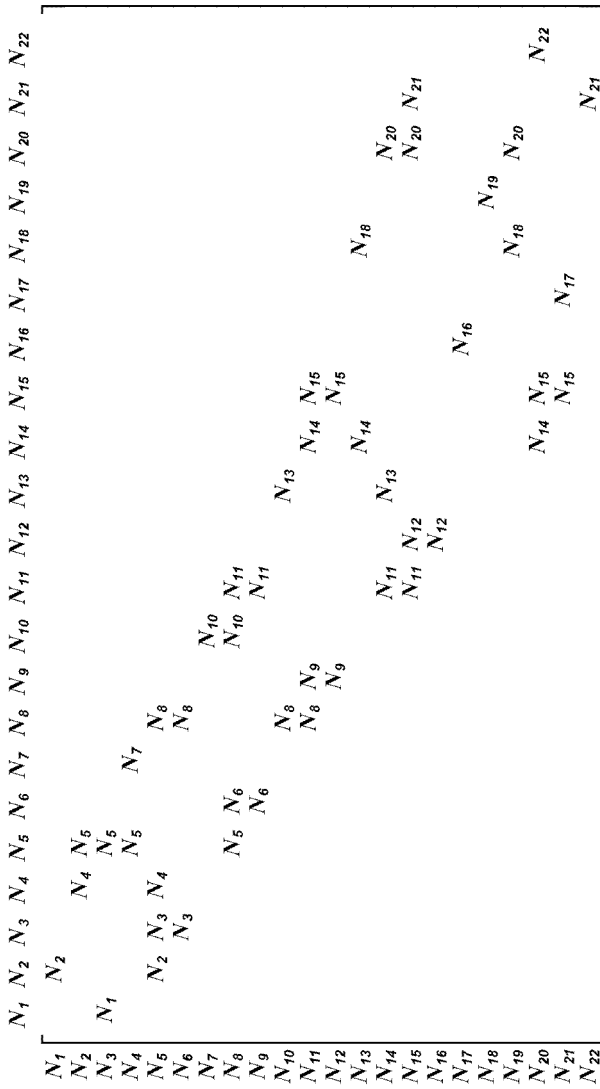


Figura 12. Matriz Θ^0 .

Figure 12. Matrix Θ^0 .

La figura 13, muestra la matriz final de valores de recorrido, correspondiente a la iteración W^{M+1} en tanto que la figura 14 presenta la matriz Θ^M , ambas matrices obtenidas a partir del algoritmo presentado en la figura 9. Finalmente, la figura 15 muestra un fragmento de la matriz de rutas Θ^{M+1} , determinada al aplicar el algoritmo de la figura 10 sobre la matriz Θ^M , mientras que la figura 16 ilustra las rutas mínimas entre los nodos N_1 a N_{22} y N_4 a N_{17} , que se corresponden a las entradas de la matriz Θ^{M+1} entre los vértices N_1 a N_{22} y N_4 a N_{17} , como era de esperar, estas rutas se corresponden exactamente.

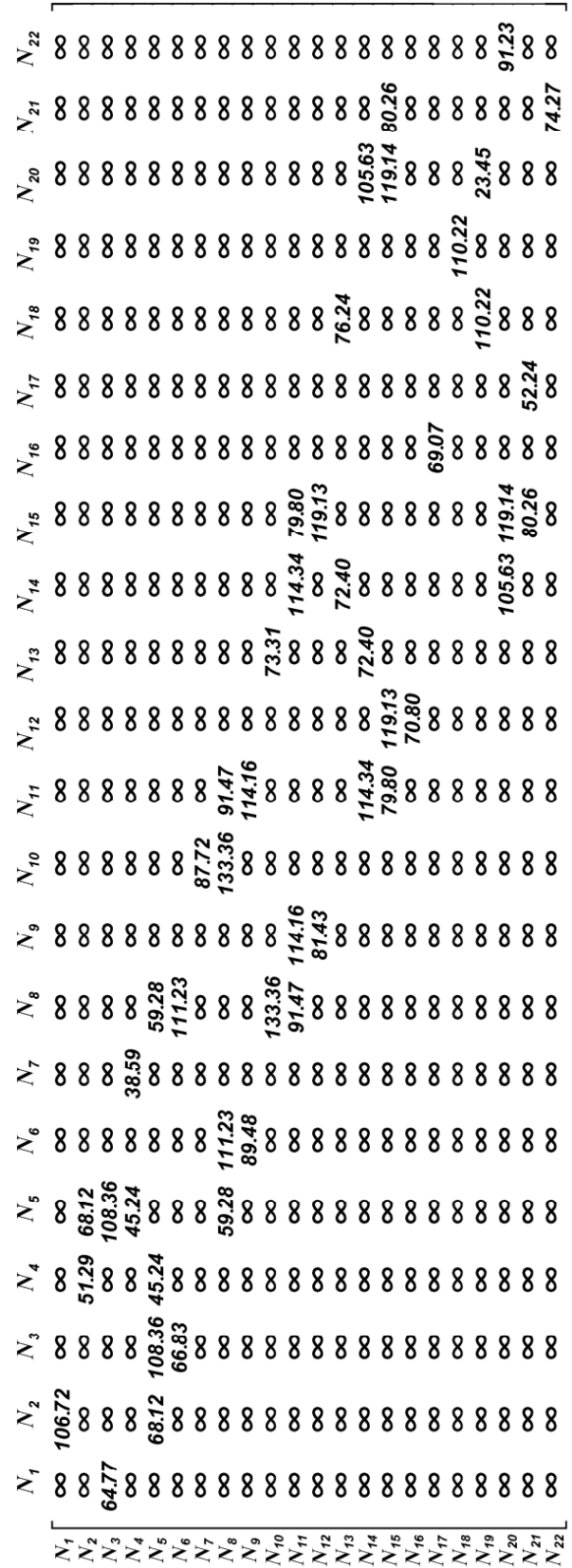


Figura 13. Iteración W^{M+1} .

Figure 13. Iteration W^{M+1} .

N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}	N_{11}	N_{12}	N_{13}	N_{14}	N_{15}	N_{16}	N_{17}	N_{18}	N_{19}	N_{20}	N_{21}	N_{22}
N_5	N_2	N_5	N_2	N_2	N_8	N_4	N_5	N_{11}	N_7	N_8	N_{15}	N_{10}	N_{13}	N_{11}	N_{21}	N_{21}	N_{13}	N_{18}	N_{15}	N_{15}	N_{20}
N_2	N_5	N_6	N_4	N_5	N_8	N_4	N_5	N_{11}	N_7	N_8	N_{15}	N_{10}	N_{13}	N_{11}	N_{21}	N_{21}	N_{13}	N_{18}	N_{15}	N_{15}	N_{20}
N_3	N_1	N_5	N_5	N_5	N_8	N_5	N_5	N_{11}	N_7	N_8	N_{15}	N_{10}	N_{13}	N_{11}	N_{21}	N_{21}	N_{13}	N_{18}	N_{15}	N_{15}	N_{20}
N_4	N_5	N_6	N_5	N_5	N_8	N_7	N_5	N_{11}	N_7	N_8	N_{15}	N_{10}	N_{13}	N_{11}	N_{21}	N_{21}	N_{13}	N_{18}	N_{14}	N_{16}	N_{20}
N_5	N_3	N_3	N_4	N_4	N_8	N_4	N_8	N_{11}	N_7	N_8	N_{15}	N_{10}	N_{13}	N_{11}	N_{21}	N_{21}	N_{13}	N_{18}	N_{15}	N_{15}	N_{20}
N_6	N_3	N_3	N_8	N_8	N_8	N_8	N_8	N_{11}	N_8	N_8	N_{15}	N_{10}	N_{13}	N_{11}	N_{21}	N_{21}	N_{13}	N_{18}	N_{15}	N_{15}	N_{20}
N_7	N_{10}	N_{10}	N_{10}	N_{10}	N_{10}	N_{10}	N_{10}	N_{11}	N_{10}	N_{10}	N_{15}	N_{10}	N_{13}	N_{11}	N_{21}	N_{21}	N_{13}	N_{18}	N_{14}	N_{15}	N_{20}
N_8	N_5	N_5	N_5	N_5	N_5	N_5	N_5	N_{11}	N_{10}	N_{10}	N_{15}	N_{10}	N_{13}	N_{11}	N_{21}	N_{21}	N_{13}	N_{18}	N_{15}	N_{15}	N_{20}
N_9	N_6	N_6	N_6	N_6	N_6	N_6	N_6	N_{11}	N_{10}	N_{10}	N_{15}	N_{10}	N_{13}	N_{11}	N_{21}	N_{21}	N_{13}	N_{18}	N_{15}	N_{15}	N_{20}
N_{10}	N_8	N_8	N_8	N_8	N_8	N_8	N_8	N_{11}	N_{10}	N_{10}	N_{15}	N_{10}	N_{13}	N_{11}	N_{21}	N_{21}	N_{13}	N_{18}	N_{14}	N_{15}	N_{20}
N_{11}	N_8	N_8	N_8	N_8	N_8	N_8	N_8	N_{11}	N_{10}	N_{10}	N_{15}	N_{10}	N_{13}	N_{11}	N_{21}	N_{21}	N_{13}	N_{18}	N_{14}	N_{15}	N_{20}
N_{12}	N_9	N_9	N_9	N_9	N_9	N_9	N_9	N_{11}	N_{10}	N_{10}	N_{15}	N_{10}	N_{13}	N_{11}	N_{21}	N_{21}	N_{13}	N_{18}	N_{15}	N_{15}	N_{20}
N_{13}	N_{14}	N_{14}	N_{14}	N_{14}	N_{14}	N_{14}	N_{14}	N_{11}	N_{10}	N_{10}	N_{15}	N_{10}	N_{13}	N_{11}	N_{21}	N_{21}	N_{13}	N_{18}	N_{14}	N_{15}	N_{20}
N_{14}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{15}	N_{10}	N_{13}	N_{11}	N_{21}	N_{21}	N_{13}	N_{18}	N_{15}	N_{15}	N_{20}
N_{15}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{15}	N_{10}	N_{13}	N_{11}	N_{21}	N_{21}	N_{13}	N_{18}	N_{15}	N_{15}	N_{20}
N_{16}	N_{12}	N_{12}	N_{12}	N_{12}	N_{12}	N_{12}	N_{12}	N_{11}	N_{11}	N_{11}	N_{15}	N_{10}	N_{13}	N_{11}	N_{21}	N_{21}	N_{13}	N_{18}	N_{14}	N_{15}	N_{20}
N_{17}	N_{16}	N_{16}	N_{16}	N_{16}	N_{16}	N_{16}	N_{16}	N_{11}	N_{11}	N_{11}	N_{15}	N_{10}	N_{13}	N_{11}	N_{21}	N_{21}	N_{13}	N_{18}	N_{16}	N_{15}	N_{20}
N_{18}	N_{20}	N_{20}	N_{20}	N_{20}	N_{20}	N_{20}	N_{20}	N_{11}	N_{11}	N_{11}	N_{15}	N_{10}	N_{13}	N_{11}	N_{21}	N_{21}	N_{13}	N_{18}	N_{16}	N_{15}	N_{20}
N_{19}	N_{20}	N_{20}	N_{20}	N_{20}	N_{20}	N_{20}	N_{20}	N_{11}	N_{11}	N_{11}	N_{15}	N_{10}	N_{13}	N_{11}	N_{21}	N_{21}	N_{13}	N_{18}	N_{16}	N_{15}	N_{20}
N_{20}	N_{15}	N_{15}	N_{15}	N_{15}	N_{15}	N_{15}	N_{15}	N_{11}	N_{11}	N_{11}	N_{15}	N_{10}	N_{13}	N_{11}	N_{21}	N_{21}	N_{13}	N_{18}	N_{16}	N_{15}	N_{20}
N_{21}	N_{15}	N_{15}	N_{15}	N_{15}	N_{15}	N_{15}	N_{15}	N_{11}	N_{11}	N_{11}	N_{15}	N_{10}	N_{13}	N_{11}	N_{21}	N_{21}	N_{13}	N_{18}	N_{16}	N_{15}	N_{20}
N_{22}	N_{21}	N_{21}	N_{21}	N_{21}	N_{21}	N_{21}	N_{21}	N_{11}	N_{11}	N_{11}	N_{15}	N_{10}	N_{13}	N_{11}	N_{21}	N_{21}	N_{13}	N_{18}	N_{16}	N_{15}	N_{20}

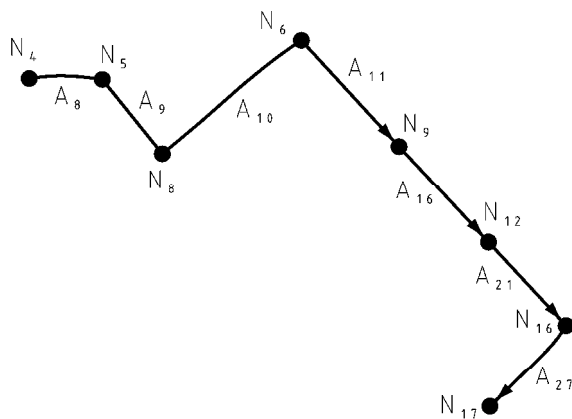
Figura 14. Matriz Θ^M .
Figure 14. Matrix Θ^M .

6. CONCLUSIONES

El algoritmo de Warshall extendido propuesto en este trabajo permite obtener la secuencia de segmentos de recorridos mínimos entre la totalidad de los nodos en un grafo. Esta característica al ser utilizada sobre el modelo en grafos de un sistema vial, permite encontrar las rutas óptimas y el valor de su recorrido entre los puntos de referencia especificados en el modelo.

El algoritmo es correcto desde el punto de vista computacional, ya que resuelve el problema planteado para la estructura de datos de entrada definida, y su tiempo de ejecución es finito.

N_1	N_1	N_1	N_1	N_1	N_1	N_1	N_1	N_1	N_1	N_1	N_1	N_1	N_1	N_1	N_1	N_1	N_1	N_1	N_1	N_1	N_1
N_2	N_2	N_2	N_2	N_2	N_2	N_2	N_2	N_2	N_2	N_2	N_2	N_2	N_2	N_2	N_2	N_2	N_2	N_2	N_2	N_2	N_2
N_3	N_3	N_3	N_3	N_3	N_3	N_3	N_3	N_3	N_3	N_3	N_3	N_3	N_3	N_3	N_3	N_3	N_3	N_3	N_3	N_3	N_3
N_4	N_4	N_4	N_4	N_4	N_4	N_4	N_4	N_4	N_4	N_4	N_4	N_4	N_4	N_4	N_4	N_4	N_4	N_4	N_4	N_4	N_4
N_5	N_5	N_5	N_5	N_5	N_5	N_5	N_5	N_5	N_5	N_5	N_5	N_5	N_5	N_5	N_5	N_5	N_5	N_5	N_5	N_5	N_5
N_6	N_6	N_6	N_6	N_6	N_6	N_6	N_6	N_6	N_6	N_6	N_6	N_6	N_6	N_6	N_6	N_6	N_6	N_6	N_6	N_6	N_6
N_7	N_7	N_7	N_7	N_7	N_7	N_7	N_7	N_7	N_7	N_7	N_7	N_7	N_7	N_7	N_7	N_7	N_7	N_7	N_7	N_7	N_7
N_8	N_8	N_8	N_8	N_8	N_8	N_8	N_8	N_8	N_8	N_8	N_8	N_8	N_8	N_8	N_8	N_8	N_8	N_8	N_8	N_8	N_8
N_9	N_9	N_9	N_9	N_9	N_9	N_9	N_9	N_9	N_9	N_9	N_9	N_9	N_9	N_9	N_9	N_9	N_9	N_9	N_9	N_9	N_9
N_{10}	N_{10}	N_{10}	N_{10}	N_{10}	N_{10}	N_{10}	N_{10}	N_{10}	N_{10}	N_{10}	N_{10}	N_{10}	N_{10}	N_{10}	N_{10}	N_{10}	N_{10}	N_{10}	N_{10}	N_{10}	N_{10}
N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}	N_{11}
N_{12}	N_{12}	N_{12}	N_{12}	N_{12}	N_{12}	N_{12}	N_{12}	N_{12}	N_{12}	N_{12}	N_{12}	N_{12}	N_{12}	N_{12}	N_{12}	N_{12}	N_{12}	N_{12}	N_{12}	N_{12}	N_{12}
N_{13}	N_{13}	N_{13}	N_{13}	N_{13}	N_{13}	N_{13}	N_{13}	N_{13}	N_{13}	N_{13}	N_{13}	N_{13}	N_{13}	N_{13}	N_{13}	N_{13}	N_{13}	N_{13}	N_{13}	N_{13}	N_{13}
N_{14}	N_{14}	N_{14}	N_{14}	N_{14}	N_{14}	N_{14}	N_{14}	N_{14}	N_{14}	N_{14}	N_{14}	N_{14}	N_{14}	N_{14}	N_{14}	N_{14}	N_{14}	N_{14}	N_{14}	N_{14}	N_{14}
N_{15}	N_{15}	N_{15}	N_{15}	N_{15}	N_{15}	N_{15}	N_{15}	N_{15}	N_{15}	N_{15}	N_{15}	N_{15}	N_{15}	N_{15}	N_{15}	N_{15}	N_{15}	N_{15}	N_{15}	N_{15}	N_{15}
N_{16}	N_{16}	N_{16}	N_{16}	N_{16}	N_{16}	N_{16}	N_{16}	N_{16}	N_{16}	N_{16}	N_{16}	N_{16}	N_{16}	N_{16}	N_{16}	N_{16}	N_{16}	N_{16}	N_{16}	N_{16}	N_{16}
N_{17}	N_{17}	N_{17}	N_{17}	N_{17}	N_{17}	N_{17}	N_{17}	N_{17}	N_{17}	N_{17}	N_{17}	N_{17}	N_{17}	N_{17}	N_{17}	N_{17}	N_{17}	N_{17}	N_{17}	N_{17}	N_{17}
N_{18}	N_{18}	N_{18}	N_{18}	N_{18}	N_{18}	N_{18}	N_{18}	N_{18}	N_{18}	N_{18}	N_{18}	N_{18}	N_{18}	N_{18}	N_{18}	N_{18}	N_{18}	N_{18}	N_{18}	N_{18}	N_{18}
N_{19}	N_{19}	N_{19}	N_{19}	N_{19}	N_{19}	N_{19}	N_{19}	N_{19}	N_{19}	N_{19}	N_{19}	N_{19}	N_{19}	N_{19}	N_{19}	N_{19}	N_{19}	N_{19}	N_{19}	N_{19}	N_{19}
N_{20}	N_{20}	N_{20}	N_{20}	N_{20}	N_{20}	N_{20}	N_{20}	N_{20}	N_{20}	N_{20}	N_{20}	N_{20}	N_{20}	N_{20}	N_{20}	N_{20}	N_{20}	N_{20}	N_{20}	N_{20}	N_{20}
N_{21}	N_{21}	N_{21}	N_{21}	N_{21}	N_{21}	N_{21}	N_{21}	N_{21}	N_{21}	N_{21}	N_{21}	N_{21}	N_{21}	N_{21}	N_{21}	N_{21}	N_{21}	N_{21}	N_{21}	N_{21}	N_{21}
N_{22}	N_{22}	N_{22}	N_{22}	N_{22}	N_{22}	N_{22}	N_{22}	N_{22}	N_{22}	N_{22}	N_{22}	N_{22}	N_{22}	N_{22}	N_{22}	N_{22}	N_{22}	N_{22}	N_{22}	N_{22}	N_{22}



16. b Ruta óptima entre nodos N_4 y N_{17} .

16. b Path between nodes N_4 and N_{17} .

Figura 16 Rutas mínimas obtenidas mediante el algoritmo extendido de Floyd-Warshall.

Figure 16 Shortest paths obtained using Floyd-Warshall extended algorithm.

La sencillez y economía computacional del método presentado permite que sea utilizado en implementaciones en sistemas de baja capacidad, pues los requerimientos de cómputo son excedidos con facilidad por un computador personal de gama baja.

Las características del algoritmo propuesto permiten su utilización en situaciones tales como la selección del centro de atención médica más cercano a la ocurrencia de un accidente, la identificación de las rutas de entrega de mercancías, y en general, en situaciones donde se haga necesario encontrar y detallar las rutas de costo mínimo en el uso de mallas viales, siendo especialmente útil en sistemas urbanos dinámicos que presentan cambios súbitos, por lo cual es una herramienta valiosa desde el punto de vista de la inteligencia computacional.

7. REFERENCIAS

- [1] GIBBONS ALAN., Algorithmic graph theory, Cambridge: Cambridge University Press, 1994.
- [2] ROSS, Kenneth, y WRIGHT Charles Discrete Mathematics, Saddle River, New Jersey, Prentice-Hall, Inc.2002.
- [3] GIRIDHAR A., KUMAR P.R., Scheduling Automated Traffic on a Network of Roads Vehicular Technology, IEEE Transactions on, Volume 55, Issue 5, 1467 – 1474, September 2006.
- [4] LIU S., DONG Y., Characterizing the hierarchy of road network and its landscape effect with graph theory, The 7th International Symposium on Operations Research and Its Applications (ISORA'08) Lijiang, China, 152–159, October 31–November 3, 2008.
- [5] SHIMIZU H., KOBAYASHI M. A., ISHIKAWA H., FUJII H., Traffic flow control system in urban road networks, Circuits and Systems, 2004. MWSCAS '04. The 2004 47th Midwest Symposium on Volume 3 25-28, iii - 149-52, July 2004.
- [6] YUEFENG L, YIQIN X., ZHIMING G., JIANGHUA Z., SHI Q., Feature-based two level structure road network model for navigation, Geoscience and Remote Sensing Symposium, 2005. IGARSS '05, IEEE International Volume 2 25-29, 4 pps. 878 – 881, July 2005.
- [7] PUN-CHENG L., LI Z., Optimal Path Finding Independent of Centerline Topology, Shaping the Change XXIII FIG Congress Munich, Germany, October 8-13, 2006.
- [8] CHEN Y., WU C., YAO M., Dynamic Topology Construction for Road Network, Networked Computing and Advanced Information Management, 2008. NCM '08. Fourth International Conference on Volume 1 2-4, 359 – 364, September 2008.
- [9] TANG Y., ZHANG Y., CHEN H., A Parallel Shortest Path Algorithm Based on Graph-Partitioning and Iterative Correcting, High Performance Computing and Communications, 2008. HPCC '08. 10th IEEE International Conference on, 155 – 162, September 2008.
- [10] LIN S., CAI W., Logistics vehicle routing problem between two objects based on real-time traffic data, Service Operations and Logistics and Informatics 2008, IEEE/SOLI 2008, IEEE International Conference on Volume 2 12-15, 2989 – 2994, October 2008.
- [11] NAGAR A., TAWFIK H., A Multi-CriteriaBased Approach to Prototyping Urban Road Networks, Issues in Informing Science and Information Technology Volume 4, 2007.
- [12] JAGADEESH G.R., SRIKANTHAN T., Route computation in large road networks: a hierarchical approach, Intelligent Transport Systems, IET Volume 2, Issue 3, 219 – 227, September 2008.
- [13] PARK C-K., SUNG K., DOH S., PARK S., Finding a path in the hierarchical road networks, Intelligent Transportation Systems, 2001, IEEE 25-29, 936 – 942, August 2001.

- [14] SHAW K., IOUP E., SAMPLE J., ABDELGUERFI M., TABONE O., Efficient Approximation of Spatial Network Queries using the M-Tree with Road Network Embedding, 19th International Conference on Scientific and Statistical Database Management (SSDBM 2007)
- [15] MAINALI M. K., Shimada K., Mabu S., Hirasawa K., Optimal route of road networks by dynamic programming, Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence), 3416 – 3420, June 2008.
- [16] YUE H., SHAO C., Study on Distributed and Parallel Search Strategy of Shortest Path in Urban Road Network, Natural Computation, 2007. ICNC 2007. Third International Conference on Volume 5 24-27, 457 – 462, August 2007.
- [17] CORMEN, Th. H., LEISERSON, Ch. E., RIVEST, R. L., STEIN, Cl., Introduction to Algorithms, Cambridge, MIT Press and McGraw-Hill, 3rd edition, 2009.
- [18] RAMESH Bh., Survivable Networks: Algorithms for Diverse Routing, Berlin, Springer-Verlag, 1999.
- [19] GOODAIRE E., PARMENTER M., Discrete Mathematics, Upper Saddle River, New Jersey, Pearson, 2006.