

ALGORITMO DE BÚSQUEDA DE RUTAS CON PREFERENCIAS ¹

Jorge E. SAGULA*, René J. TESEYRA⁺

Laboratorio LADSI - Universidad Católica de Santiago del Estero, Argentina

*jsagula@infovia.com.ar

⁺ladsi@ucse.edu.ar

Resumen. En este artículo se introduce un algoritmo de búsqueda que combina características de Métodos de Búsqueda en Haz y Búsqueda Voraz, juntamente con métodos de Decisión Multicriterio con el propósito de resolver problemas de búsqueda de rutas con restricciones de preferencias, donde los puntos se organizan en un multigrafo masivamente conectado, tal que esta representación resulta adecuada para modelar situaciones reales (o en su defecto, aproximaciones a la realidad). Además, en el enfoque propuesto se intenta dar una respuesta a situaciones en las cuales no es posible hallar una forma de conexión entre el nodo origen y el nodo destino respetando las preferencias introducidas por un usuario; al efecto, se emplean técnicas de determinación de proximidad denominadas K-ésimo Vecino Más Cercano para hallar una locación lo más cercana posible que satisfaga los requerimientos impuestos por las preferencias de usuario.

Palabras Clave: Multigrafo – Heurística – Vecino más Cercano – Proceso Analítico Jerárquico.

Abstract. This article introduces a search algorithm that combines characteristics of Beam-Search and Voracious-Search methods, together with Multicriteria Decision methods with the purpose of solving search problems constrained routing preferences, where the points are arranged in a massively connected multigraph, such that this representation is suitable for modeling real situations (or failing that, approximations of reality). Furthermore, the proposed approach tries to respond to situations in which it is possible to find a form of connection between the source node and the destination node respecting the preferences introduced by a user, the purpose, techniques used for determining proximity called K-th nearest Neighbor to find a location as close as possible to meet the requirements imposed by the user preferences.

Keywords: Multigraph – Heuristic – Nearest Neighbors – Analytic Hierarchy Process.

¹Este artículo fue realizado como corolario de la investigación desarrollada en el marco del Proyecto MOTAR, adjudicado en la convocatoria UCSE 2010, en la Universidad Católica de Santiago del Estero, Argentina, durante el año 2011.

1 Introducción

Se han realizado numerosos estudios, análisis, experimentos, desarrollos y aplicaciones de modelos de resolución en problemas de determinación de la ruta óptima, a fin de conectar dos puntos cualesquiera dentro de un espacio de búsqueda; dentro de los modelos de mayor relevancia se citan: A* [7]; SetA* [8]; Algoritmos Meméticos [9]; Búsqueda Tabú [2], [4]; Búsqueda Amplia Adaptativa de Vecindad [1]; Algoritmos Voraces [3]; sin dejar de lado el modelo de Colonia de Hormigas (Swarm Intelligence) [10] en la determinación de rutas.

Otro aspecto destacable de las heurísticas de búsqueda es que, casi todas buscan descubrir “la Solución Óptima”, por ende, muchas de las pruebas de bondad de ajuste de tales heurísticas, se orientan a demostrar que pueden converger al mejor resultado sin caer en óptimos locales.

Si bien consideramos que se han realizado numerosos trabajos de investigación en lo atinente a la inclusión de preferencias de usuario como guía del proceso de búsqueda de rutas, nuestro enfoque propone aplicar las heurísticas de búsqueda de rutas como una primera fase algorítmica, para luego mejorarlas mediante un segundo algoritmo que aplique técnicas de Toma de Decisión Multicriterio [11], reduciendo de esta forma el conjunto solución y ordenándolo de acuerdo a las preferencias de los usuarios. Las heurísticas [8], [2], [1], [3], pueden lidiar perfectamente con restricciones incluidas en el proceso mencionado previamente; sin embargo, es necesario consignar que no se debe confundir una restricción con una preferencia, pues en tanto la primera impone cotas o limitantes al proceso de búsqueda, la segunda busca establecer prioridades para la conformación del conjunto de soluciones candidatas.

En éste artículo proponemos un algoritmo para búsqueda de rutas posibles para conectar un nodo origen y un nodo destino cualesquiera dentro de un espacio de búsqueda, de modo que el conjunto de soluciones candidatas resultante estará ordenado, en función de su bondad, sobre un conjunto de preferencias impuestas por un usuario. Además, es importante destacar que el algoritmo se aplicará a grandes espacios de búsqueda masivamente conectados, consecuentemente, la forma de representación más adecuada es mediante un multigrafo (un multigrafo es un grafo en el cual dos nodos pueden estar conectados por más de un arco o arista).

2 Representación de un Multigrafo

Un importante número de heurísticas de búsqueda se basan en la organización de los puntos a conectar en un formato de grafo dirigido o no dirigido: $G = (V, A)$, donde:

V: es el conjunto de vértices o nodos del grafo, $\{v_0, v_1, \dots, v_n\}$

A: es el conjunto de aristas o arcos que conectan los diferentes nodos del grafo, tal que:

$$A = \{(v_i, v_j) : v_i \wedge v_j \in V, i \neq j\} \quad (1)$$

Desde el diseño de la búsqueda, es necesario disponer de cierta información a fin de guiar la heurística de búsqueda, sea cual fuere que se trate. En nuestro enfoque, cada nodo o vértice contiene información de las coordenadas posicionales dentro del multigrafo y cada arco almacena información en forma de atributos, por ejemplo: el costo de recorrer el camino de un nodo i hasta otro nodo j , la calidad del servicio que realiza el recorrido, los tiempos de salida del nodo inicial y la llegada al nodo final, etc. Los atributos consignados están en estrecha relación con los criterios que el usuario debe ponderar antes de comenzar el proceso de búsqueda de la solución.

A partir de lo expuesto previamente surgen las particularidades del enfoque propuesto en el presente trabajo, a saber:

1. la información para guiar el recorrido en la búsqueda de la solución no está contenida en los arcos sino en los nodos o vértices.
2. los arcos deben tener dirección, se tiene un Grafo Dirigido, ya que los atributos de los k arcos que unen i con j , no necesariamente son los mismos que los l arcos que unen j con i ²
3. debido al hecho que la representación elegida es un multigrafo dirigido, resulta obvio que en el mismo se pueden presentar ciclos en los diferentes caminos que se pueden formar, corriendo el riesgo de quedar atrapado en un bucle durante el recorrido del mismo. Será un aspecto clave a considerar en el desarrollo del algoritmo de búsqueda el hecho de evitar quedar atrapado en un bucle durante el proceso de solución.

Gráficamente, la representación se muestra, en forma resumida, en la fig.

1.

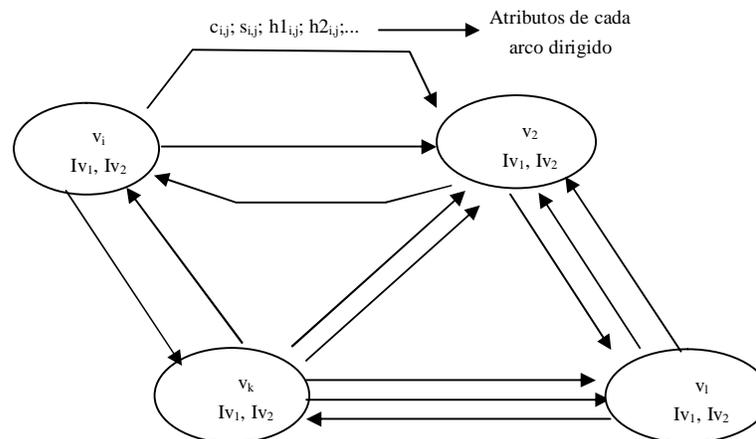


Fig. 1. Representación Resumida de un Multigrafo

² obsérvese que la cantidad de arcos entre i y j no es necesariamente la misma que entre j e i .

3 Optimización de Rutas en un Multigrafo

Los problemas de determinación de rutas o caminos para el transporte vehicular, típicamente, se orientan a tres tipos de decisiones [6]:

- Decisiones de Asignación: ubicación de los vehículos en lugares clave.
- Decisiones de Secuencia: definición del orden de visita para cada vehículo.
- Decisiones de Planificación: determinación de un horario de las visitas de los lugares seleccionados para cada vehículo.

Luego de fijar las dos primeras, la tercera se deduce en forma sencilla; sin embargo, esto no ocurre con la representación de un multigrafo, pues además se debe determinar qué arco utilizar entre dos nodos consecutivos seleccionados como integrantes de un camino o una ruta solución; este problema, denominado Problema de Selección de Arco de Secuencia Fija (Fixed Secuence Arc Selection Problem (FSASP)), es un problema NP-Hard, que es posible resolverlo en forma aproximada, significativamente eficiente, mediante métodos iterativos.

En el enfoque propuesto, el planteo consiste en tratar de establecer un conjunto de rutas óptimas desde el nodo origen hasta el nodo destino dentro de un multigrafo con un elevado número de conexiones entre sus nodos. Al concluir el proceso previo, se realizará una clasificación de las mismas conforme a la bondad con la que estas se ajustan al ordenamiento de criterios o preferencias introducidas por el usuario. Si no es posible encontrar ruta alguna desde el nodo origen al nodo destino, se construye un conjunto solución de rutas en forma aproximada, llegando a nodos cercanos al nodo destino, siendo éstas, a su vez, clasificadas por su bondad de ajuste.

4 Definición Formal del Problema de Estudio

El problema presentado se puede definir formalmente a partir de la tupla:

$$P = (n_0, n_f, V, Iv, O, C) \quad (2)$$

tal que cada una de las componentes representa:

n_0 : nodo inicial o punto de partida de la búsqueda.

n_f : nodo final o punto de finalización de la búsqueda.

V : conjunto de nodos que conforman el multígrafo (en este trabajo se trata del conjunto de locaciones geográficas que se desean conectar mediante los arcos dirigidos).

Iv : información contenida en cada nodo del multigrafo, para el presente trabajo se tratará de las coordenadas que sirven para posicionar espacialmente cada nodo, es decir latitud y longitud.

O : conjunto de operaciones permitidas o reglas de transición (en este algoritmo se escoge la Relación de Preferencia [3], [5], que servirá para determinar qué es necesario hacer a partir de un nodo determinado, es decir, guiará el proceso de transición entre nodos del multígrafo).

A partir de un nodo determinado, siguiendo las rutas no marcadas anteriormente, se postula el conjunto de nodos alcanzables a partir de dichas rutas, de este conjunto se seleccionan los k arcos más promisorios y los l arcos promisorios determinados a partir de la Relación de Preferencia de Conjunto “ \langle ” que se define a continuación:

Sea un nodo $i \in G$ y sea $A' = \{ai,j1; \dots; ai,jm\}$, con $j = 1 \dots m$, el conjunto de arcos con origen en i , entonces:

$MP = \{ai,jd; \dots; ai,jg\}$, es el conjunto de nodos Más Promisorios alcanzables desde i tal que cumplen:

$$n_f(lv_m) - i(lv_m) > n_f(lv_m) - j(lv_m) \wedge n_f(lv_{m+1}) - i(lv_{m+1}) > n_f(lv_{m+1}) - j(lv_{m+1}) \quad (3)$$

Entonces se puede afirmar que:

$jd \langle jk, \forall j$ que sea destino de algún arco $ai,j \in MP$, en relación con otros jk que no cumplan la relación.

$P = \{ai,jp; \dots; ai,jr\}$, es el conjunto de nodos Promisorios alcanzables desde i tal que cumplen:

$$n_f(lv_m) - i(lv_m) > n_f(lv_m) - j(lv_m) \vee n_f(lv_{m+1}) - i(lv_{m+1}) > n_f(lv_{m+1}) - j(lv_{m+1}) \quad (4)$$

Entonces se puede afirmar que:

$jp \langle jk, \forall j$ que sea destino de algún arco $ai,j \in P$, en relación con otros jk que no cumplan la relación.

El conjunto de vértices j , alcanzables desde i que sean destinos de algún arco pertenecientes a MP o P , se desestiman como no promisorios para alcanzar el nodo final.

La función principal de la Relación de Preferencia de Conjunto, es lograr la poda del espacio de búsqueda de forma de acotar los tiempos de cómputo del algoritmo. Cada nodo desestimado por no resultar promisorio, es una ramificación que no se explorará.

C : es el conjunto de Criterios al que el usuario debe dar prioridad; estos criterios servirán para establecer un orden de bondad de cada solución encontrada al finalizar la operación del algoritmo.

En este artículo sólo se ha seleccionado una lista acotada, privilegiando exclusivamente tres criterios:

- Calidad de servicio: se refiere a las características del medio que se empleará para el recorrido de cada porción de la ruta seleccionada.
- Minimización de tiempos de espera: se refiere a los tiempos ociosos en el caso que para llegar del nodo origen al destino se deba realizar paradas en nodos intermedios.
- Economía de costo: se refiere al costo monetario, o de otros recursos, total necesarios para realizar el recorrido desde el nodo origen al destino.

Resulta necesario consignar, en este punto, que es posible expandir el número de criterios, tema que en este caso formará parte de la dirección futura de investigación.

5 Descripción del Algoritmo

El algoritmo propuesto en este trabajo contempla conceptos de varios métodos de búsqueda heurística, a saber: Métodos Voraces [3]; Búsqueda en Haz [13]; Algoritmos Anytime [14]; además, métodos de Toma de Decisión Multicriterio [11]; y métodos de Determinación de Proximidad o Vecindad, por caso: k-ésimo Vecino Más Cercano [12].

En lo atinente al proceso de construcción del conjunto de soluciones, se puede decir que, desde un multigrafo en el que se presentan una elevada cantidad de conexiones entre sus nodos, se procede a construir un árbol en cuya raíz se ubica el nodo o locación inicial de la búsqueda, cada una de sus ramificaciones serán rutas o caminos aplicables para llegar a un único nodo hoja (en este caso, el nodo o locación destino de la búsqueda) o, en caso de no ser posible cumplir con la premisa anterior, se arribará a un conjunto de hojas correspondientes a los nodos o locaciones lo más cercanas geográficamente al nodo destino.

Seguidamente se postula las entradas, los resultados esperados y la descripción funcional del algoritmo propuesto en pseudocódigo.

Entradas:

n_0 : Nodo inicial
 n_f : Nodo final o destino
 V : Conjunto de nodos
 I_v : Información contenida en cada nodo. Latitud y longitud, por lo tanto 2 valores.
 C : Conjunto de Criterios de Búsqueda Ordenado por Preferencia
 \langle : Relación de Preferencia

Resultados Esperados:

MP: Conjunto de Nodos Más Promisorios
 P : Conjunto de Nodos Promisorios
 NS : Conjunto de Nodos en Rutas o Caminos Solución
 R : Conjunto de Arcos en Rutas o Caminos Solución
 RP : Conjunto de Arcos en Rutas o Caminos Posible Solución

Algoritmo en Pseudocódigo:

```
MP ← {n0}
WHILE MP ≠ ∅ AND P ≠ ∅ do
  IF MP ≠ ∅ THEN
    i ← get_next_node(MP)
  ELSE
    i ← get_next_node(P)
  FOREACH j ai,j DO
    FOREACH k aki,j DO
      IF {j} ∉ NS OR {j} ∉ MP OR {j} ∉ P THEN
        IF j = nf THEN
```

```

        R ← { aki,j }
        NS ← j
    ELSE
        IF nf(lv1) - i(lv1) > nf(lv1) - j(lv1) AND nf(lv2)
- i(lv2) > nf(lv2) - j(lv2) THEN
            MP ← MP ∪ {j}
            RP ← RP ∪ { aki,j }
            get_relative_solution_distance (j)
        ELSE
            IF nf(lv1) - i(lv1) > nf(lv1) - j(lv1) OR nf(lv2) -
i(lv2) > nf(lv2) - j(lv2) THEN
                P ← P ∪ {j}
                RP ← RP ∪ { aki,j }
                get_relative_solution_distance (j)
        NEXT k
    NEXT j

MP ← MP - {i} OR P ← P - {i} //según el conjunto del
//que se tomó i en la
//primera sentencia IF

IF NS ≠ ∅ THEN
    FOREACH l ∈ NS
        get_backward_path_from l //usando backward chaining
//de arcos contenidos en
//RP
ELSE
    get_best_neighborhood_solutions_form RP
    FOREACH m ∈ MP
        get_backward_path_from m //usando backward chaining
//con los arcos
//contenidos en RP
        get_best_neighborhood_solutions_form P
    FOREACH o ∈ P
        get_backward_path_from o //usando backward chaining
//con los arcos
//contenidos en P.
    weigh_solutions //Pondera la bondad de
//cada solución tomando en
//cuenta el ordenamiento
//de criterios realizado
//por el usuario conforme
//a los criterios estable-
//cidos en C

```

6 Ponderación de la Bondad de Ajuste de las Rutas en el Conjunto Solución

Para completar el proceso de determinación de las mejores soluciones, el algoritmo se orienta al ordenamiento de criterios de preferencia establecidos por el usuario al momento de iniciar la búsqueda desde el nodo origen hasta el nodo destino.

Para determinar el cumplimiento de cada solución encontrada respecto a la ponderación de la importancia relativa asignada a cada criterio se utiliza el método AHP [11].

Es necesario recordar lo expuesto en el Inciso 3., pues cada arco tiene asociada información sobre distintas características del medio usado para conectar los nodos extremos de dicho arco, esta información se incorporará al método AHP, a fin de poder calcular el grado de cumplimiento de cada “trayecto” del camino o ruta solución satisfaciendo cada criterio.

Seguidamente, se describen los pasos a considerar:

1. Establecer la ponderación relativa de cada criterio, considerando el orden establecido por el usuario.
2. Para cada solución ordenarla respecto a la correspondencia de información arcos → criterio i .
3. Con respecto a cada criterio, realizar una división del ordenamiento de las soluciones en cuartiles.
4. A los elementos del primer cuartil les corresponde el valor 5, a los elementos del segundo cuartil el valor 3, a los elementos del tercer cuartil el valor 1/3 y a los elementos del cuarto cuartil el valor 1/5.
5. Incorporar los datos calculados al método AHP.

7 Problema Real: Aplicación

El enfoque desarrollado se ha transferido a la resolución de un problema real, consistente en el hallazgo de rutas a partir del tránsito vehicular entre un nodo origen y un nodo destino.

La aplicación fue codificada en el lenguaje de programación JAVA utilizando el entorno de desarrollo ECLIPSE en virtud de tener compatibilidad con múltiples plataformas; la Base de Datos fue construida en MySQL.

Si bien la aplicación se concluyó en todas sus fases, la Base de Datos no contiene la totalidad de los datos reales al momento de la implementación, consignando la carga de los nodos locaciones juntamente con la información requerida para el funcionamiento del algoritmo. Específicamente, se han cargado cerca de 20.000 nodos, restando fundamentalmente la carga de datos referentes a los arcos para conectar tales nodos, pues la disponibilidad de datos reales ha sido restringida; consecuentemente, se han simulado los datos faltantes a partir de los existentes, con el propósito de poder disponer de un multigrafo masivamente conectado para las pruebas finales del tiempo computacional del algoritmo.

8 Conclusiones

En este artículo se ha propuesto una adaptación de técnicas y métodos de búsqueda heurística, tales como Búsqueda en Haz y Métodos Voraces junto con métodos de Toma de Decisiones y Métodos de Determinación de Proximidad como el K-ésimo Vecino más Cercano, para abordar el problema de determinación de rutas o caminos en amplios espacios de búsqueda masivamente conectados, similarmente a los que se presentan en la realidad. El objeto de esta investigación, teórica y aplicada, no fue desarrollar un algoritmo para encontrar “la Mejor Solución”, sino hallar un conjunto de soluciones factibles y admisibles, que resulten aplicables dadas las condiciones iniciales propuestas por usuarios particulares en instantes dados.

Consideramos que la principal fortaleza del algoritmo propuesto radica en que algunas veces resultará imposible llegar al destino requerido por el usuario respetando las restricciones establecidas mediante los criterios de preferencia; en tal situación, el algoritmo será capaz de mostrar las soluciones más próximas encontradas, respetando tales condiciones y señalando qué tan alejado se está de alcanzar el destino. Además, si existe alguna otra forma de alcanzar la meta final, sin respetar las restricciones impuestas, estas rutas también formarán parte del conjunto de soluciones, informando al usuario que no se respetan la totalidad de las restricciones impuestas.

9 REFERENCIAS BIBLIOGRÁFICAS

1. BJÖRNSSON, Yngvi, HALLDÓRSSON, Kári; (2006); "Improved Heuristics for Optimal Pathfinding on Game Maps" en Second Artificial Intelligence and Interactive Digital Entertainment Conference, Marina del Rey, California; pp 9-14.
2. BONET, Blai, GEFNER, Hector; (2000); "Planning as Heuristic Search"; Artificial Intelligence Volume 129, Issues 1–2, June 2001; pp 5–33.
3. CANON, LouisClaude, EMMANUEL; (2011); "MO-Geedy: an Extended Beam-Search Approach for Solving a Multi-Criteria Scheduling Problem on Heterogeneous Machines"; en IEEE International Parallel & Distributed Processing Workshop; pp 57-69.
4. CORDEAU, Jean-François, MAISCHBERGER, Mirko; (2011); "A Parallel Iterated Tabu Search Heuristic for Vehicle Routing Problems" en COMPUTERS & OPERATIONS RESEARCH, N° 39 (9); pp 2033-2050.
5. EPPE, Stefan, LOPEZ-IBAÑEZ, Manuel, STÜTZLE, Thomas, De SMET, Yves; (2011); "An Experimental Study of Preference Model Integration into Multi-Objective Optimization Heuristics" en 2011 IEEE Congress on Evolutionary Computation (CEC); pp 2751-2758.
6. GARAIX, Thierry, ARTIGUES, Christian, FEILLET, Dominique, JOSSELIN, Didier; (2010); "Vehicle Routing Problems with Alternativ Path: an Application to Demand Responsive Transports" en The European Journal of Operational Research 204(1); pp 62-75.
7. HART, Peter, NILSSON, Nils, RAPHAEL, Bertram; (1968); "A Formal Basis for Heuristic Determination of Minimum Path Cost" en IEEE Transaction of Systems Science and Cybernetics vol 2; pp 100-104.
8. JENSEN, Rune, BRYANT, Randal, VELOSO, Manuela; (2002); "SetA*: An Efficient BDD-Based Heuristic Search Algorithm" en The 18th National Conference on Artificial Intelligence; pp 668-673.

9. MOSCATO, Pablo, COTTA, Carlos; (2003); "Una Introducción a los Algoritmos Meméticos" en Revista Iberoamericana de Inteligencia Artificial. No.19 (2003),pp. 131-148.
10. MOU, Lian-Ming; (2011); "A Novel Ant Colony System with Double Pheromones for the Generalized TSP" en Seventh International Conference on Natural Computation (ICNC); pp 1923-1928.
11. SAATY, Thomas; (1980); The Analytic Hierarchy Process; McGraw-Hill International.
12. SILVERMAN, B; (1986); Density estimation for statistics and data analysis; Chapman and Hall.
13. SHAFIA, Mohammad, AGHAEI, Mohsen, SADJADI, Seyed, JAMILI, Amin; (2011); "Robust Train Timetabling Problem: Mathematical Model and Branch and Bound Algorithm" en IEEE Transactions on Intelligent Transportation Systems, vol 13 N° 1; pp 307-317.
14. VLASIS, N, ELHORST, R, KOK, J; (2004); "Anytime Algorithms for Multiagent Decision Making Using Coordination Graphs" en IEEE International Conference on Systems, Man and Cybernetics; pp 953-957.