

Ingeniería en Sistema de Información

Trabajo final para analista

El problema del viajante de comercio

Año: 1996
Alumno: Francisco G. Simó
Legajo: 24383-6
Domicilio: Roque Sáenz Peña 157, Villa Carlos Paz (5152), Córdoba Barcelona, España
Teléfono: 0541-22292 +34 655 384 230
e-mail: fsimo@computer.org fransimo@gmail.com
<http://webs.satlink.com/usuarios/f/fsimo/> <http://fransimo.info/>

Para Guillermo M.

Agradezco especialmente la colaboración y paciencia prestadas: a mi mamá, Paquita, a Sergio H. Castro, Jorge E. Torres y Andrea L. Incatasciato. Ellos posibilitaron que este trabajo sea comprensible, supieron soportar mis interrogatorios y exigencias, y se molestaron en comprender lo que intentaba decir.

Índice.

Índice.....	1
Introducción.....	3
Consideraciones metodológicas.....	5
Conceptos.....	6
El problema del viajante de comercio.....	6
Heurísticas.....	6
Modelo de representación de la ciudad.....	7
Red.....	7
Nodos.....	7
Arcos.....	7
Adyacencia.....	8
Camino o cadena.....	8
Anillos.....	8
Árbol abierto.....	8
Longitud de un camino.....	8
Capas.....	8
Islas.....	8
Representación de la red a través de matrices de adyacencia.....	8
Representación de la red a través de ecuaciones lineales.....	9
Nodos proveedores o fuente.....	9
Nodos de transbordo.....	9
Nodos de paso.....	10
Nodos de destino.....	10
Nodos de destino final.....	10
Representación de la red a través de listas.....	10
Representaciones utilizadas en informática.....	10

Sistemas georreferenciados.....	10
Algoritmos.....	11
Programación lineal, problema de transbordo.....	11
Algoritmo de Dijkstra, para encontrar el camino más corto.....	13
Algoritmo heurístico basado en información georreferenciada.....	14
Representación de la ciudad en el algoritmo.....	14
Supuestos.....	15
Descripción del algoritmo.....	15
Primera etapa.....	15
Segunda etapa.....	16
Ejemplo de aplicación.....	16
Primera parte.....	16
Segunda Parte.....	19
Limitaciones.....	19
Redes Neuronales y Algoritmos genéticos.....	20
Redes Neuronales.....	20
Algoritmos genéticos.....	21
Propuestas futuras.....	23
Experimentos con heurísticas sobre sistemas georreferenciados.....	23
Conclusiones.....	24
Fuentes de información.....	25
Nota del asesor.....	28
Indice de referencias cruzadas.....	29

Introducción

A mediados de 1994, en busca de un sistema para la cátedra seminario de sistemas, me encontré con un amigo que me propuso la resolución de un problema que lo tenía preocupado.

En su empresa se realizaban visitas mensuales a todos los clientes de la firma para la entrega de productos. A medida que el tiempo fue pasando, y la empresa creciendo, el reparto dejó de ser un problema trivial para convertirse en un problema fundamental. La planificación y control de los recorridos ya no podía hacerse en forma improvisada.

En ese punto no se podía garantizar la atención al cliente en un cien por cien de las veces con el personal existente, pero ciertamente el problema no era la falta de personal, era la desorganización.

Frente a esta situación la empresa empezó a trabajar en una solución que implementaron con éxito de manera no automatizada y por un método poco científico, y muy subjetivo, de planificación.

El problema propuesto es indudablemente general dentro de las empresas que cuentan con un sistema de reparto de mercadería o de visita a clientes, por ejemplo para brindar servicios de mantenimiento. En el ámbito de la investigación operativa este problema es conocido como el problema del viajante.

En la mayoría de los casos se confía en la experiencia del personal encargado de los repartos para este tipo de planificación. Sin lugar a duda la capacidad intelectual de las personas para resolver problemas complejos y ambivalentes no es despreciable, pero tener una herramienta científicamente comprobada para la toma de decisiones es una gran ayuda. Uno puede confiar en un rendimiento promedio, muy eficiente y acotadamente desviado al utilizar un sistema automatizado para el planeamiento de la distribución. El hecho de contar con un sistema de este tipo le permitirá al responsable del área proponer caminos o visitas alternativas para diferentes días y evaluar con anticipación el impacto, de por ejemplo no encontrar a un cliente específico o de un retraso en un sector de la ciudad.

Los objetivos de este trabajo son:

- dar a conocer el problema,
- explicar algunas de las representaciones que se utilizan comúnmente para resolverlo,
- exponer algunas de las formas de resolución actuales, incluyendo redes neuronales y algoritmos genéticos,
- proponer un algoritmo heurístico de resolución, fácilmente programable, y dejar una implementación de prueba,
- dejar una guía para continuar el trabajo presentado aquí.

No encontramos entonces con un problema sumamente complejo, en el trabajo se estudiará el problema bajo supuestos que permitan disminuir esta complejidad a niveles razonables.

La resolución del camino óptimo, dado un conjunto de puntos a recorrer, no es nada nuevo existe un algoritmo de Dijkstra que permite encontrar los caminos más cortos entre dos puntos,

representados en un grafo. Si realizamos una aplicación recursiva de este algoritmo es posible encontrar un camino perfecto.

Pero ciertamente este algoritmo está lejos de ser técnicamente factible ya que para la resolución del problema evalúa todos los posibles recorridos entre cada par de puntos, y si tenemos en cuenta, que nuestro mapa tiene todas las calles de la ciudad la ‘explosión combinatoria’ de caminos hace que el tiempo de proceso sea exageradamente alto.

El algoritmo de Dijkstra tiene esa falla porque cada nodo del grafo no tiene información particular respecto de su ubicación geográfica, por lo tanto debe evaluar todos los recorridos. Pero si podemos contar con esa información vamos a poder aplicar otro tipo de reglas, como evaluación de recorridos aéreos.

Consideraciones metodológicas.

El objetivo de esta investigación es comprender el problema conocido, dentro del ámbito de la investigación operativa, como el *problema del viajante de comercio* y de alguna de las posibles soluciones, dejando planteado cuáles deberían ser los objetivos de investigaciones posteriores, ya sean del tipo exploratorio o explicativo.

Debido a limitaciones de tiempo, costos, tecnología y recursos humanos se excluirán ciertos temas relacionados con las últimas investigaciones realizadas en el ámbito mundial, ya que sería necesarios viajes y cursos especiales a distintos lugares del país. Sin embargo, se los mencionará, aclarando que no se posee el rigor metodológico necesario como para considerarlos dentro de la investigación, en el sentido de soluciones al problema. Estas limitaciones también impidieron el desarrollo de experimentos, que quedarán planteados al final del trabajo para ser continuados en el futuro.

La fuente de información fueron libros, revistas, seminarios, publicaciones en la web y el consejo de expertos en temas relacionados, que ayudaron a guiar la búsqueda bibliográfica. Cabe mencionar que si bien existe mucho escrito respecto al tema, en el marco de la investigación operativa, no existen diferencias marcadas en sus aspectos teóricos ni prácticos. Frente a esta situación no se mencionarán todas las fuentes utilizadas, sino sólo aquellas que son de fácil acceso, teórico y económico para consultas posteriores.

El análisis se centró en el estudio de los *algoritmos*, contrastados con las teorías en las que se basan. No se presentan, debido a la naturaleza inconexa entre las diferentes teorías y sus algoritmos, ninguna relación matemática de causa consecuencia. En futuros trabajos, experimentales, tal vez se pudiera demostrar qué teorías resuelven el problema más rápidamente, comparando la eficiencia de los algoritmos frente a situaciones reales. Estos resultados serían de un interés fundamental para aplicaciones de ingeniería.

Debido a que esta investigación comenzó como un problema que necesitaba solución (y a causa de mi formación en, lo que podríamos llamar, ciencias de la computación) mi tendencia natural de resolución fue tratar el problema como un algo netamente informático.

Esto se convirtió en una barrera temporal para encontrar soluciones alternativas y para permitir explorar campos de resolución inusuales para mí. Sin lugar a duda podríamos considerar a esto como un prejuicio hacia la naturaleza del problema y su solución.

Finalmente encontré respuestas alternativas en las matemáticas y la investigación operativa además de las puramente informáticas. Por supuesto, el planteo matemático del problema es el más completo, pudiendo considerar todas las restricciones, pero tiene serios problemas de implementación.

Conceptos

El problema del viajante de comercio

El viajante tiene como objetivo visitar a un grupo de clientes dentro de la ciudad. Nuestro objetivo es planificar una ruta para él, tal que el recorrido total sea mínimo. El problema puede ser dividido según el grado de complejidad con el que lo abordemos. Esta complejidad está en función de:

- La cantidad de restricciones operativas que consideremos; por ejemplo, si ciertos clientes tienen prioridad sobre otros, si los destinos poseen diferentes horarios dentro de los que se puede arribar, etc.
- La cercanía del modelo de representación de la ciudad con la realidad, si sólo poseemos información sobre avenidas y calles principales o debemos considerar toda la complejidad de la red vial dentro de la ciudad.
- La existencia de restricciones en las formas de recorrer la ciudad, como por ejemplo las manos de las calles.
- La aparición de 'islas', lugares a los que se puede acceder sólo a través de puentes, que pueden hacer imposible la utilización de heurísticas simples para la resolución del problema.
- Si estos puentes son unidireccionales o bidireccionales.
- El vehículo del viajante. No es lo mismo resolver el problema si él viaja en auto, si tiene que tomar líneas de urbanos o si puede caminar toda la ruta.
- Si se debe considerar los problemas de tránsito en avenidas según horarios del día o condiciones particulares del tiempo.

Es fundamental comprender que estos problemas no afectan a la teoría sino a la implementación tecnológica de una solución. Dentro de 20 años, si el avance de la potencia de cálculo de las computadoras continúa como hasta ahora, quien lea esto se morirá de risa.

Heurísticas

Arte de inventar según el diccionario¹. Dentro del ámbito de la informática una heurística es una regla que nos permite, frente a un número grande de posibles soluciones parciales, elegir una que supuestamente tiene una mayor probabilidad de contribuir a la solución final del problema, con el objetivo de ahorrar tiempo de procesamiento. Las heurísticas no son soluciones totales y poseen un margen de error. Sin embargo, los seres humanos las utilizamos permanentemente en la solución de problemas complejos.

Los expertos humanos crean reglas que les ayudan a resolver problemas más rápidamente, éstas son, en cierta forma, heurísticas. Si consideráramos, por ejemplo, el problema de llegar desde la facultad hasta la terminal de ómnibus, la cantidad de recorridos posibles es enorme. Para resolver el problema deberíamos calcular cuanto tiempo tardaríamos en recorrerlos a todos y elegir

¹ Diccionario Enciclopédico Espasa-Calpe

el menor, en cambio, un humano normal, utiliza su experiencia y selecciona un camino casi instantáneamente. Este camino tal vez no sea el óptimo, pero seguramente no será tan malo.

Una posible manera de definir estos métodos es como “procedimientos simples, a menudo basados en sentido común, que se supone ofrecerán una buena solución (aunque no sea necesariamente la óptima) a problemas difíciles, de un modo fácil y rápido” [Zanakis, Evans, 1981]. [OhyRN Pág. 24]

Modelo de representación de la ciudad

La ciudad será representada como una red, esta será el mapa para nuestros algoritmos de solución. Cada algoritmo puede utilizar, a su vez, distintas formas de expresión para las redes como matrices, ecuaciones lineales o listas multidimensionales de datos.

Red

Una red es un modelo de representación de la realidad, está formada por nodos y arcos, que dependiendo del tipo de representación que se intente hacer, poseen distintos significados. La definición de Tanenbaum² para grafo es: “*Un grafo consta de un conjunto de nodos (o vértices) y un conjunto de arcos (o aristas)*”. Para simplificar trataremos los términos red y grafo como sinónimos.

Nodos

Los nodos, para la aplicación que nos interesa, representan lugares físicos, como el local de un cliente, una esquina, etc.

- Información del nodo: Cada nodo tiene un conjunto de datos relacionados, entre ellos el más importante es su ubicación espacial dentro de la ciudad.
- Georreferencia: Existirán nodos que no posean la ubicación espacial, pero que harán referencia a otro nodo que sí la tenga, tal vez en forma recursiva. Esta referencia se llama georreferencia, en el más puro sentido de la red, los nodos que no tengan una referencia espacial directa no forman parte del modelo de la ciudad, sino que nos ayudan a comprenderla y a usarla. Por ejemplo, los clientes del viajante estarán georreferenciados, pero no forman parte de la ciudad, así mismo se define un barrio a través de un conjunto de nodos que referencian a las esquinas que lo delimitan.

Arcos

“*Cada arco de un grafo se especifica por un par de nodos*” [TNB Pág. 531]. Los arcos expresan la conexión física que existe entre los nodos. A través de ellos circula un flujo, que para nuestro caso será el viajante desplazándose. Un arco es, por ejemplo, una calle que va de esquina a esquina, el recorrido de una línea de urbanos entre cada una de sus paradas, etc. Los arcos siempre muestran la relación entre dos nodos vecinos.

- Dirigidos: “*Si los pares de nodos que constituyen los arcos son pares ordenados, se dice que es un grafo dirigido (o dígrafo)*” [TNB Pág. 531]. Entonces, en un arco dirigido el flujo puede moverse en una sola dirección.
- No dirigidos: Es aquel sobre el cual los flujos pueden ir en cualquier sentido.
- Pesos: Es un valor que expresa el costo unitario del flujo. Cada arco tiene un peso. El

²Tanenbaum Estructuras de datos en C. Pág.: 531

objetivo de los problemas de investigación operativa es optimizar la sumatoria total de los pesos involucrados en la solución. Según los apuntes de la cátedra Sistemas de Datos un grafo cuyos arcos tienen asociado un peso es una red.

- Distancias: El peso de los arcos es un costo de flujo, para nuestro problema el cálculo del peso puede encararse desde distintas posiciones. El peso puede calcularse: como la distancia física entre los dos nodos que une; como el tiempo que se tarda en recorrer ese espacio o como una combinación de espacio/tiempo, una velocidad. La elección de una de las opciones depende de cuál es nuestro objetivo, si minimizar las distancias recorridas o los tiempos empleados.

Adyacencia

“Un nodo ‘n’ es adyacente a un nodo ‘m’ si existe un arco de m a n” [TNB Pág. 533]

Camino o cadena

Es una sucesión de nodos y arcos que conectan dos nodos, de manera tal que cierto flujo pueda llegar del primero al último. [DAVIS Pág. 288]

Anillos

Es un camino que conecta a un nodo con sí mismo. [DAVIS Pág. 288]

Árbol abierto

Reúne a un subconjunto de los arcos de la red original que conecta todos los nodos pero que no contiene ningún anillo. [DAVIS Pág. 288]

Longitud de un camino

“Un camino de longitud k del nodo a al nodo b se define como una secuencia de $(k+1)$ nodos n_1, n_2, \dots, n_{k+1} de tal manera que $n_1=a$, $n_{k+1}=b$ y n_i y n_{i+1} sean adyacentes para toda i entre 1 y k . Si para algún entero k , existe un camino de longitud k entre a y b ”. [TNB Pág. 233] De lo que deducimos que la longitud de un camino es, en teoría, la suma de la cantidad de arcos que se necesitan para formarlo.

Capas

Como ya vimos existen nodos que no son puntos espaciales propiamente dichos, sino que son una ayuda para la representación. Estos nodos pueden ser agrupados para expresar algo, esta agrupación es una capa. Las capas pueden ser utilizadas por su importancia en la representación o como una ayuda a la aplicación real. Las capas pueden ser, por ejemplo, todas las calles de la ciudad, una capa muy compleja. O las grandes avenidas y los barrios, como capas menos complejas. También podría ser el cableado de una red de televisión por cable, el cableado de EPEC, la red de distribución de agua o gas, todas las líneas de urbanos, etc. [JGIS]

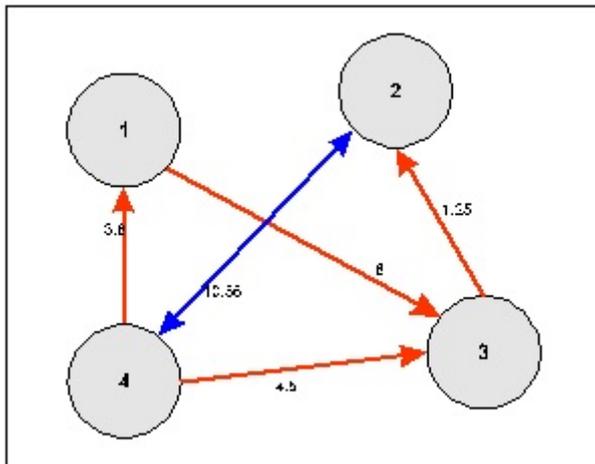
Islas

Una isla es una subred, un subconjunto de arcos y nodos, que están casi aislados de los demás, y a los que solamente se puede llegar a través de un número restringido de caminos, estos caminos serán llamados puentes. Si representáramos Nueva York, Manhattan sería una isla dentro del modelo. Al representar la ciudad de Córdoba existen dos grandes islas divididas por el río Suquía. Si representáramos una provincia cada una de sus ciudades sería una isla.

Representación de la red a través de matrices de adyacencia

Aquí la red está representada por una matriz, que tanto en sus columnas, como en sus filas

posee la lista de nodos de la red. En una red de cuatro nodos tendríamos una matriz de 4x4. Cada elemento de la matriz representa el costo del flujo entre los dos nodos que involucra su posición.



Si no existe arco se convendría un valor excesivamente alto o bajo para el peso. Entonces si existe un arco entre el nodo 4 y el 3 y el peso asociado es 4.5, el valor del elemento (4,3) será igual a 4.5, si el arco fuera bidireccional o no dirigido el elemento (3,4) sería igual al (4,3). [TNB Pág. 536]

“Supongamos que existen 4 ciudades a las que denominamos 1,2,3 y 4, y caminos que comunican algunas de ellas entre sí. El siguiente grafo muestra un ejemplo de implementación, sobre arreglos, en una matriz de valores lógicos que representa una tabla de doble entrada.

Figure 1 - Red básica

Llamaremos a esta ‘matriz de adyacencia’; cada dimensión de este arreglo contiene N elementos, siendo N la cantidad de nodos del grafo. Las filas de la matriz representan los nodos de partida de cada arco y las columnas los de llegada. Un valor verdadero en la intersección de la fila i y j significa que existe un arco que parte del nodo i y llega al nodo j. En el caso de grafos que factor de peso, los elementos que componen la matriz adyacente son de un tipo más elaborado (records, pointers, reales), lo cual le permitirá almacenar la información asociada a cada arco” [DATOS]

Desde/Hacia	1	2	3	4
1	∞	∞	8	∞
2	∞	∞	∞	10.56
3	∞	1.25	∞	∞
4	3.8	10.56	4.5	∞

Representación de la red a través de ecuaciones lineales

Esta representación es utilizada por la programación lineal. Todo modelo de programación lineal está compuesto por tres partes: función de optimización, restricciones tecnológicas y restricciones de no negatividad. Las restricciones tecnológicas, que son un sistema de ecuaciones lineales, representan la red en los llamados problemas de transbordo, ruta más corta y transporte.

En este sistema de ecuaciones cada variable representa el flujo que pasa a través de un arco dirigido, y cada ecuación indica la conectividad de un nodo, su demanda o la necesidad de que la ruta pase por él. Dependiendo del tipo de demanda y la forma de la ecuación los nodos pueden ser proveedores o fuente, de transbordo, de paso, de destino y de destino final. [DAVIS. Capítulo 7, Págs. 281-338]

Nodos proveedores o fuente

Son aquellos que introducen el flujo en la red, para nuestro caso existirá solo un nodo proveedor y será el lugar desde donde parte el viajante, por eso lo llamaremos nodo de

partida.[DAVIS Pág 291]

Nodos de transbordo

A través de estos nodos circula el flujo y, eventualmente, pueden retener parte de él.[DAVIS Pág 291]

Nodos de paso

A través de estos nodos circula el flujo pero no retienen ningún flujo. Todo el flujo que reciben lo entregan.

Nodos de destino

En el problema del viajante son los nodos por los cuales él debe pasar para cumplir su ruta. Estos nodos reciben flujo y entregan pero se caracterizan por su necesidad de recibir y entregar la misma cantidad.[DAVIS Pág 291]

Nodos de destino final

Estos nodos reciben flujo pero nunca entregan. Para resolver el problema del viajante se crea un único nodo destino final ficticio que posee la misma ubicación geográfica que el nodo de partida.

Representación de la red a través de listas

“Si bien la matriz adyacente es sencilla de implementar, tiene desventajas para algunas aplicaciones. Por un lado, debido a la naturaleza estática del tamaño de su almacenamiento debe conocerse de antemano la cantidad de nodos, lo que limita la capacidad de crecimiento o encogimiento que un problema pueda requerir. Por otra parte en un grafo con poca proporción de arcos respecto a la cantidad de nodos, su matriz su vuelve altamente esparcida, lo cual implica que hay gran cantidad de espacio reservado inutilizado, ya que se deberá guardar espacio para cada arco posible, exista o no. La solución a este inconveniente se encuentra en las estructuras encadenadas....(de la comparación con un árbol)...Para representarlo nos servíamos de listas de adyacencia; esta lista adyacente era conformada por todos los hijos del nodo en cuestión haciendo la vez de hermanos. Cada nodo en la estructura contenía dos punteros: uno hacia la lista de descendientes del nodo en cuestión y otro es el que forma la lista de hermanos del mismo. Sin embargo, en un grafo un nodo puede tener más de una arco y de hecho este nodo puede encontrarse en muchas listas adyacentes. Una alternativa de solución es la siguiente:

- Formar una lista vinculada donde cada nodo contiene la información asociada a él. A esta lista la denominaremos lista de cabeceras.
- Formar listas adyacentes a cada uno de los nodos de la lista mencionada que representen los arcos que unen al nodo que de inicio a la lista con los otro nodos del grafo”[DATOS]

Representaciones utilizadas en informática

Por los problemas de espacio antes mencionados, la mayoría de las aplicaciones informáticas utilizan representación en forma de listas.

Sistemas georreferenciados

Debido a que no es el objetivo de este trabajo no se hará una clasificación completa del tema, que puede encontrarse en varios trabajos finales, pero aclararé que cada vez que se mencione el tema se referirá a sistema georreferenciados vectoriales.

Algoritmos

Programación lineal, problema de transbordo

Como ya había aclarado, todos los modelos de programación lineal tienen tres grandes componentes: una función a optimizar, función de decisión; un conjunto de ecuaciones lineales, llamadas restricciones tecnológicas y las restricciones de no negatividad.

Primero definiremos la red como un conjunto formado por otros dos:

$$(1) \quad R = (A, \mu)$$

Donde A es el conjunto de nodos de la red y se define como:

$$(2) \quad A = (a_1, a_2, a_3, \dots, a_k, \dots, a_n)$$

Y el conjunto μ es un conjunto de pares ordenados, que representan los arcos dirigidos:

$$(3) \quad \mu = ((a_1, a_2), (a_1, a_3), \dots, (a_i, a_j), \dots, (a_{n-1}, a_n))$$

Para resolver el problema del viajante, usaremos una adaptación de los problemas de transbordo y ruta más corta. El problema se resuelve haciendo circular un flujo imaginario de una unidad, teniendo un nodo origen, algunos destinos, un destino final y los demás nodos de la red como nodos de paso. En este tipo de problemas la función de decisión posee la siguiente forma:

$$(4) \quad F(x_{ij}) = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad [\forall (a_i, a_j) \in \mu]$$

representando c_{ij} los pesos de los arcos, las distancias; y x_{ij} el flujo que pasa a través del arco entre los nodos i y j , que debe existir en el conjunto de arcos de la red. Al final del proceso la función objetivo será la distancia total del recorrido.

Dentro del sistema de ecuaciones encontramos cuatro formas de ecuaciones dependiendo del tipo de nodo que intentemos representar y una por cada uno de los nodos de la red. La ecuación de los nodos proveedores u origen es:

$$(5) \quad \sum_{j=1}^n x_{ij} = 1$$

La ecuación de los nodos de paso es:

$$(6) \quad \sum_{k=1}^n x_{ki} - \sum_{j=1}^n x_{ij} = 0$$

Los nodos de destino tienen dos ecuaciones:

$$(7) \quad \sum_{k=1}^n x_{ki} = 1$$

$$(8) \quad \sum_{j=1}^n x_{ij} = 1$$

Y la ecuación de los nodos de destino final es:

$$(9) \quad \sum_{k=1}^n x_{ki} = 1$$

Y para todas estas ecuaciones debe cumplirse:

$$(10) \quad \begin{aligned} &\forall (a_i, a_j) \wedge (a_k, a_i) \in \mu \\ &\forall k \neq i \quad \wedge \quad j \neq i \\ &x_{ij} \geq 0 \quad \text{enteros} \end{aligned}$$

Para el problema del viajante sólo existe un nodo proveedor y un nodo final, que es ficticio. La verdadera esencia del problema se encuentra en los nodos de paso y destino.

Decimos que el problema del viajante es una adaptación de los problemas de transbordo porque se caracteriza por la forma de las demandas y las ofertas de los nodos. La oferta total del nodo proveedor es igual a uno. Y la demanda de cada nodo a visitar es igual a cero pero están obligados a recibir una unidad y a reenviarla hasta que llegue al nodo de destino final.

También es posible plantear este problema a través de programación dinámica, pero para el caso del problema del viajante dentro de una ciudad, que implica una red sumamente compleja, la programación dinámica no resulta adecuada debido a la gran cantidad de etapas que aparecerían. Si realizamos una implementación pura de la programación dinámica nada impediría que tengamos que evaluar la posibilidad de llegar a la manzana vecina dando la vuelta a toda la ciudad.

Este problema en programación lineal se resuelve al utilizar el algoritmo de resolución SIMPLEX que nos garantiza que luego de cada iteración está más cerca de su objetivo. Si tuviéramos una serie de nodos de transbordo todos con demanda, SIMPLEX evaluaría de la forma más rápida posible el camino, cosa que no ocurriría con programación dinámica, la que, aún en una situación ideal, evaluaría todas las posibilidades.

Algoritmo de Dijkstra, para encontrar el camino más corto

Lamentablemente no he encontrado algoritmos informáticos que resuelvan el problema directamente, pero sí son muy comunes algoritmos para encontrar el camino más corto entre dos puntos de un grafo.

Y conociendo el principio de optimidad de Bellman: “*Un conjunto de decisiones (una política) tiene la propiedad de que, si una decisión determinada es óptima, entonces todas las subsecuentes que dependen de esa decisión específica también deben ser óptimas*” [DAVIS Pág. 460]. Lo que implica que el camino óptimo está formado necesariamente por el conjunto de los caminos más cortos que unen a cada nodo.

Luego, se puede aplicar cualquier algoritmo de camino más corto recursivamente sobre la red y encontraremos el recorrido de distancia mínima que une a todos. Como conocemos el punto de partida buscamos el camino más corto entre él y todos los demás nodos, al finalizar los cálculos elegimos el nodo que está más cerca y volvemos a aplicar la regla a los restantes, que es la idea básica del algoritmo de Dijkstra. Indudablemente encontraremos el camino que el viajante debe seguir, pero es obvio que se realizan una gran cantidad de cálculos inútiles.

El algoritmo propuesto por Dijkstra puede trabajar con representaciones enlazadas y matrices de adyacencia, al igual que el de Ford.

La descripción del algoritmo de Dijkstra que encontramos en el libro de Tanenbaum es la siguiente:

“El camino mínimo se define como un camino de s a t tal que la suma de los pesos de los arcos comprometidos en el camino sea mínima. Para representar la red, suponemos que existe una función de pesos, tal que $weight(i,j)$ sea el peso del arco que va de i a j . Si no hay arco de i a j , $weight(i,j)$ se hace igual a un número arbitrariamente grande para indicar el costo infinito (es decir, la imposibilidad) de ir directo de i a j .

Si todos los pesos son positivos, el siguiente algoritmo, atribuible a Dijkstra, determina el camino más corto o mínimo de s a t . Sea $infinity$ una variable que guarda el entero más grande posible. ‘ $distance[i]$ ’ guarda el costo del camino mínimo conocido hasta el momento de s a i . Al

principio, $distance[s]=0$ y $distance[i]=infinity$ para toda $i \neq s$. Un conjunto $perm$ contiene todos los nodos cuya distancia mínima de s es conocida, es decir, aquellos nodos cuyo valor distancia es permanente y no cambiará. Si un nodo i es miembro de $perm$, $distance[i]$ es la distancia mínima de s a i . Inicialmente, el único miembro de $perm$ es s . Una vez que t se convierte en miembro de $perm$, $distance[t]$ es la menor distancia de s a t y el algoritmo termina.

El algoritmo mantiene una variable, $current$, que es el nodo recién agregado a $perm$. Al inicio de $current = s$. Siempre que un nodo se agregue a $perm$, $distance$ se tiene que recalcularse para todos los sucesos de $current$. Para todo sucesor i de $current$, si $distance[current] + weight(current, i)$ es menor que $distance[i]$, la distancia de s a i a través de $current$ es menor que cualquier distancia de s a i encontrada hasta el momento. Así, $distance[i]$ se tiene que hacer igual a este valor más pequeño.

Una vez que $distance$ ha sido recalculada para todo sucesor de $current$, $distance[j]$ (para cualquier j) representa el camino más corto de s a j que incluye sólo miembros de $perm$ (excepto el propio j). Esto significa que para el nodo k , que no está en $perm$, para cualquier $distance[k]$ es menor, no hay camino de s a k cuya longitud sea menor que $distance[k]$. ($distance[k]$ es ya la distancia más corta hasta k que incluye sólo nodos $perm$, y cualquier camino a k que incluya un nodo nd como su primer nodo que no esté en $perm$, tiene que ser más largo, dado que la $distance[nd]$ es mayor que $distance[k]$.) Así, k se puede agregar a $perm$, $current$ se hace luego igual a k y se repite el proceso.”[TNB Pág. 543]

Resumiendo, el algoritmo calcula, para los nodos i que no pertenecen a $perm$, y desde el nodo $current$, la distancia nueva como $distance[current] + weight(current, i)$, si esta es menor a la registrada anteriormente, la reemplaza. Cuando termina con el cálculo de las nuevas distancias selecciona el nodo más cercano que aun no pertenece a $perm$ y lo incorpora, convirtiéndolo en $current$ y volviendo a iniciar el proceso hasta que $current$ sea t , el nodo final.

Algoritmo heurístico basado en información georreferenciada.

El problema de la ‘explosión combinatoria’ que surge en los algoritmos anteriores no puede ser sorteado fácilmente. Un método muy utilizado para lograrlo es proveer al proceso de resolución de cierta información adicional. En el presente caso la información que parece más conveniente es la geográfica. Esta idea también es utilizada en las redes neuronales elásticas presentadas por Durbin y Willshaw [OhyRN Pág. 192].

Representación de la ciudad en el algoritmo

El hecho de conocer la ubicación de los puntos en el espacio nos permite intentar crear nuevos métodos heurísticos. El único requisito para desarrollar estos métodos es poseer la información de referencia.

En la figura en la página 15 podemos observar la manera en que se representa la información geográfica de la ciudad. Los puntos rojos son los nodos, en azul y amarillo los arcos. Las flechas indican la dirección del arco, es obvio que los arcos amarillos son unidireccionales y los azules bidireccionales.

Supuestos

El algoritmo que se presenta a continuación se basa en el supuesto que la red represente a una ciudad, y como tal, sus nodos poseen una alta conectividad con los adyacentes. Este supuesto permite asumir que entre dos nodos cercanos existirá un camino que se aproxime bastante a un recorrido aéreo.

Descripción del algoritmo

Primera etapa

La primera etapa del algoritmo consiste en seleccionar el orden en el que serán visitados los destinos. La segunda determinará el recorrido exacto, esquina por esquina, que se debe recorrer para ir entre pares de nodos.

Los pasos de la primera etapa son

1. Dado un conjunto de puntos en el plano y sus respectivas coordenadas.
2. Calculamos el promedio de las coordenadas x .
3. Calculamos el promedio de las coordenadas y .
4. Trazamos una línea recta entre el punto promedio y el punto origen del recorrido.
5. Elegimos una de las mitades del plano.
6. Desde el punto de origen seleccionamos el punto más cercano dentro de la mitad seleccionada, este punto se convierte en el primer paso de la ruta.

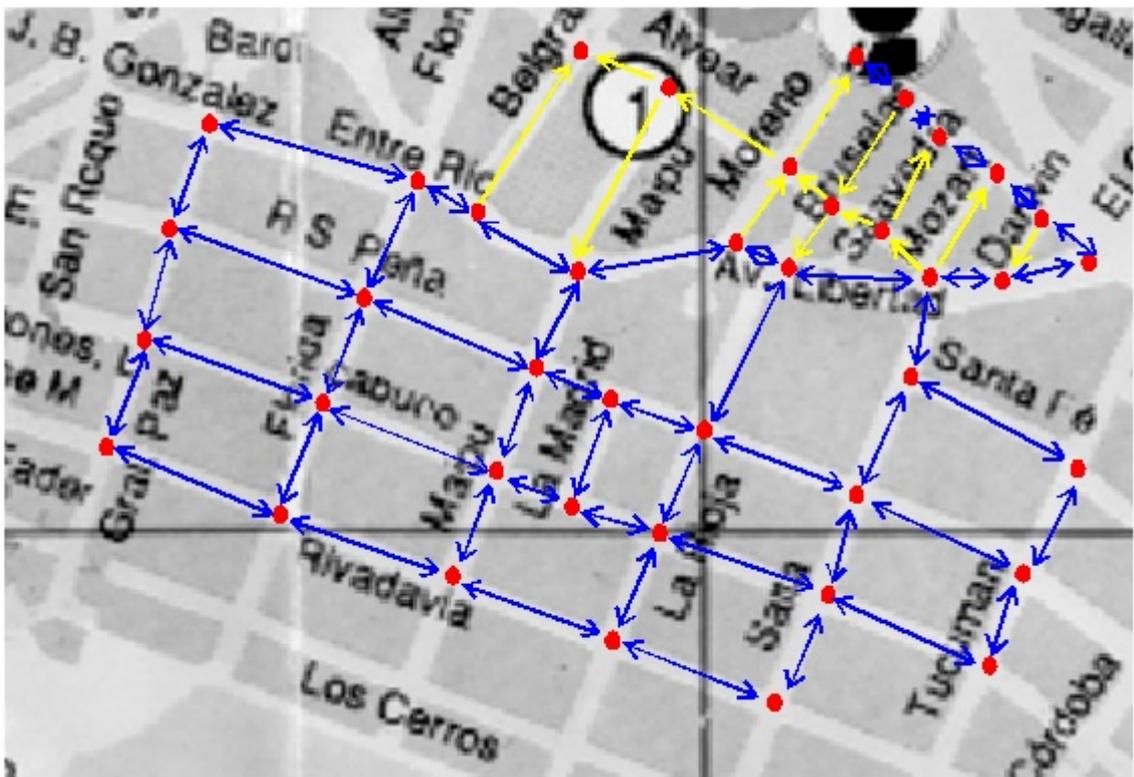


Figure 2 - Representación de la ciudad para el algoritmo propuesto

7. Continuamos seleccionando el punto más cercano al último elegido hasta terminar con los puntos que pertenecen a la primer mitad.
8. Para pasar a la segunda mitad del plano seleccionamos el punto más lejano al origen en la segunda mitad.
9. Continuamos seleccionando el punto más cercano al último elegido hasta llegar al origen.

Segunda etapa

Existen varios algoritmos que permiten determinar el mejor recorrido entre dos puntos. Uno de los más eficientes es el algoritmo de Ford. En el presente trabajo decidimos utilizar otra heurística para este caso.

Se presentaron varias opciones y seleccioné la más sencilla de implementar. Dados dos puntos, origen y destino, los pasos del algoritmo de enrutamiento son los siguientes:

1. Determinar a que otros nodos puedo llegar desde el nodo actual, la primera vez será el origen.
2. Calcular la distancia de estos nodos al destino.
3. Seleccionar aquel que tenga la distancia menor y ponerlo en la ruta.
4. Establecer como punto actual el seleccionado anteriormente.
5. Iterar hasta que el nodo seleccionado sea el nodo de destino.

Ejemplo de aplicación

Primera parte

Para explicar el funcionamiento veamos un ejemplo. Supongamos que nuestra ciudad está representada por la **Ilustración 3**, los nodos marcados con un círculo amarillo, pintado de verde, son nuestros destinos, y sus respectivas coordenadas espaciales son las que se detallan en la tabla siguiente.

Nodo destino	X	Y	Dirección
1	62.00	164.21	Rivadavia, entre General Paz y Florida
2	94.36	184.76	Roque Sáenz Peña, entre Florida y Maipú
3	121.92	193.52	Libertad, entre Maipú y La Rioja
4	163.40	173.79	Santa Fe
5	131.80	154.41	Chacabuco, entre Salta y Maipú
6	103.99	147.96	Rivadavia, entre Maipú y La Rioja
Sumatoria	677.47	1018.65	
Promedio	112.91	169.78	

Los primeros pasos eran calcular los promedios de las coordenadas, tanto x como y. Este promedio nos da un punto en el centro de todos, marcado con naranja. Después se traza una línea recta entre el punto de origen y el centro. El punto de origen no es importante para el algoritmo, pues la ruta final será la misma sin importar qué punto se seleccione como origen, pero a los efectos de que los resultados sean más amistosos para el usuario final, en la implementación el origen debería ser la base desde la cuál sale.

La línea, de color naranja claro, divide el plano en dos. En la implementación que se realizó del algoritmo los puntos 5, 6 y 7 fueron logrados realizando un giro y traslación en el espacio, de forma que la línea que divide al plano se convierta en la axisa (X), dejando el nodo de origen en las coordenadas (0,0) y el centro en algún lugar sobre la recta y=0. Las ecuaciones se describen a continuación.

$$(11) \quad x_p = \frac{\sum x_{ij} \cdot X}{n} \quad \forall x_{ij} \in \mu$$

$$(12) \quad y_p = \frac{\sum x_{ij} \cdot Y}{n} \quad \forall x_{ij} \in \mu$$

Donde x_{ij} son los nodos, que poseen dos características X e Y, referenciadas como atributos

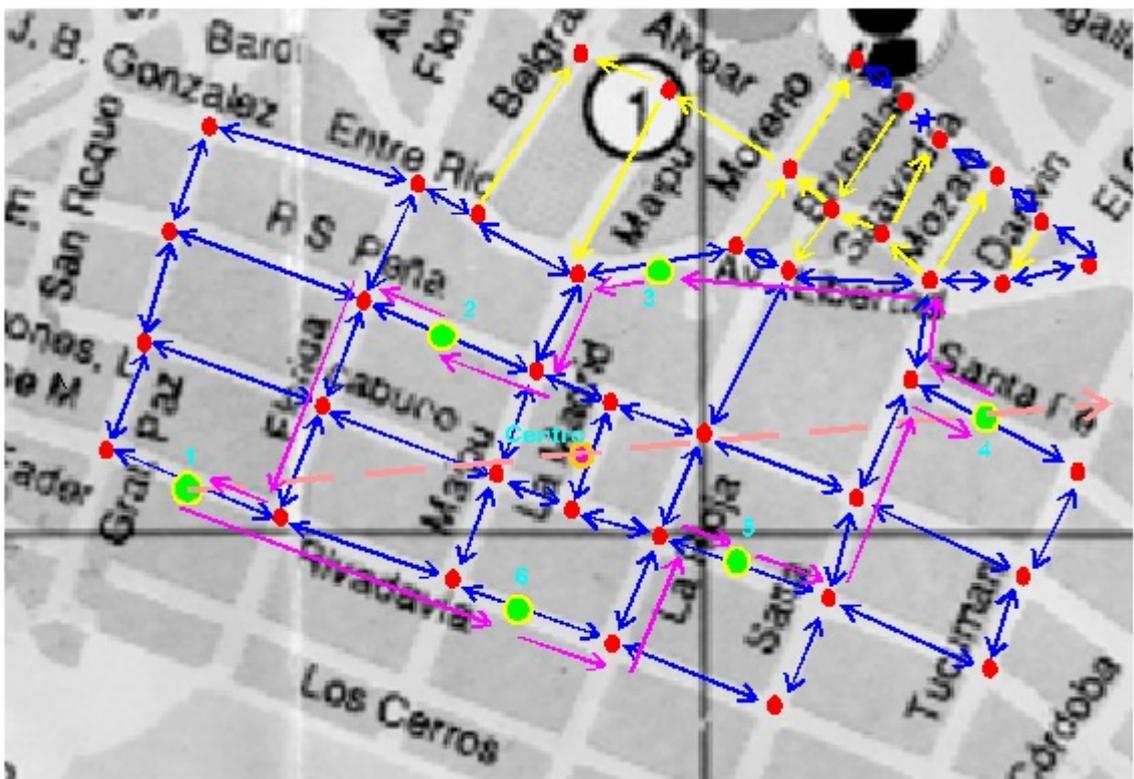


Figure 3 - Ejemplo

de objetos, mu es el conjunto de nodos a recorrer y n el número de nodos que lo conforman.

Luego para cada nodo se calculan sus nuevas coordenadas como:

$$(13) \quad x_{nueva} = (x_{ij} \cdot X - x_0) * \cos(\arctan(\frac{y_p}{x_p}))$$

$$(14) \quad y_{nueva} = (x_{ij} \cdot Y - y_0) * \sin(\arctan(\frac{y_p}{x_p}))$$

donde x_0 y y_0 son las coordenadas del nodo de origen.

Este giro y traslación permiten tener las dos mitades del plano claramente diferenciadas, aquellas con coordenadas ‘y’ negativas serán la primera mitad elegida que menciona el punto 5 del algoritmo.

Luego se calculan las distancias, según la ecuación (15), de todos los puntos con respecto al origen, que se muestran en la siguiente tabla.

$$(15) \quad Distancia = \sqrt{x^2 + y^2}$$

Nodo destino	X	Y	Distancia
1	0	0	0
2	32.16	2.24	32.24
3	59.56	3.18	59.64
4	100.80	1.04	100.80
5	69.38	-1.06	69.39
6	41.74	-1.77	41.77

Pero haremos dos salvedades, referidas a la implementación del algoritmo con bases de datos. Primero, vamos a agregar un dato más, un campo que indique si el punto esta por arriba o por debajo de la recta $y=0$. Y segundo las distancias de los puntos que estén por arriba de $y=0$, se multiplicaran por -1. Estas dos alteraciones permiten resolver toda la secuencia de nodos en una sola consulta SQL.

$$(16) \quad \textit{Arriba/Abajo} = \frac{\textit{Abs(Distancia)}}{\textit{Distancia}}$$

$$(17) \quad \textit{DistanciaNueva} = \textit{Distancia} * \textit{Arriba/Abajo} * (-1)$$

Veamos entonces cómo queda la tabla con los dos campos nuevos, pero ordenando, primero por el campo que diferencia arriba/abajo, y luego por la distancia.

Nodo destino	X	Y	Distancia	Arriba/Abajo	Distancia nueva
6	41.74	-1.77	41.77	-1	41.77
5	69.38	-1.06	69.39	-1	69.39
4	100.8	1.04	100.8	1	-100.8
3	59.56	3.18	59.64	1	-59.64
2	32.16	2.24	32.24	1	-32.24

En esta ya se puede ver el orden de los nodos para recorrer.

Segunda Parte

En esta parte el objetivo es encontrar la ruta más corta entre dos puntos, se puede utilizar cualquiera de los algoritmos conocidos. No creo necesario explicar con más detalles la rutina de enrutamiento presentada anteriormente, es muy simple. Pero es posible utilizar, como alternativa, el algoritmo de Ford, pero limitando el tamaño de la red a un espacio cercano al de los dos nodos, por ejemplo el rectángulo formado por los dos puntos, tomando la recta que los une como la diagonal del mismo.

Limitaciones

La principal desventaja de este método es la aparición de islas, concepto que va en contra del supuesto de conectividad. Aunque no se trata en este texto la solución es agrupar los puntos a visitar dentro de grupos, que estarían dentro de la misma isla, y resolverlos por separado.

El algoritmo no es fácilmente adaptable al problema de múltiples viajantes, conocido como MTSP.

Redes Neuronales y Algoritmos genéticos

Es imposible exponer todas las direcciones que han tomado las investigaciones de TSP. Hasta ahora se han expuestos los conceptos básicos que permiten modelizar el problema y sobre los que se basa el algoritmo propuesto.

Sin embargo haré una breve introducción a las dos tendencias más modernas y su relación con el problema del viajante. No es mi intención explicar en detalle estas teorías, pero citaré a distintos autores y comentarios.

Una fuente invaluable para aprender sobre el problema del viajante y su relación con estas dos nuevas tendencias es el libro Optimización heurística y Redes Neuronales. En la página web oficial del problema del viajante [TSPBIB], mantenida por uno de los autores del libro, se pueden encontrar todos los resultados de las últimas investigaciones a nivel mundial.

Redes Neuronales

Las redes neuronales son un campo relativamente nuevo, en el que teorías matemáticas permiten la modelación de neuronas humanas; presuntamente, al agruparse en redes intentan imitar al cerebro. Estos modelos, calculados con métodos especialmente desarrollados, están logrando procesar información de una manera completamente nueva dentro de la informática. Las redes neuronales son capaces de trabajar en el reconocimiento de imágenes basándose en semejanzas de forma, en la predicción de series temporales multidimensionales, reconocimiento de lenguaje natural, procesamiento de voz, filtros de ruido especiales, sistemas de guía para robots e innumerables aplicaciones que se descubren a cada momento.

Es interesante mencionar una de las características más sobresalientes de las redes neuronales, no se programan, se entrenan; la persona que realiza la implementación debe enseñarles a resolver el problema. Por lo tanto son capaces de aprender y, además, de responder a situaciones desconocidas, no tenidas en cuenta en el entrenamiento original.

Hassan M. Ghaziri, en **OhyRN** (Pág. 197), hace una comparación de tres arquitecturas de red neuronal para resolver el problema TSP. Estas arquitecturas son: Hopfield, Kohonen y Boltzman o elásticas. Sus conclusiones fueron las siguientes:

“Los dos últimos tienen en común el empleo de una estructura de red especial, en concreto una red deformable, la cual de por sí constituye un procedimiento absolutamente novedoso respecto a lo existente hasta la fecha para resolver problemas de optimización. En este caso no es necesario definir una función de coste o energía”. Esta estructura es muy potente ya que es directamente una representación del concepto de ruta y puede ser considerada como la solución del problema ideal. La interacción del anillo con los datos reales (posición de las ciudades) deformará el anillo, obteniendo finalmente una solución para el problema real.

La única diferencia entre el método de Durbin y Willshaw y el método de Fort radica en el modo en que se presentan las ciudades: en el primero la presentación se realiza en modo batch (todas las neuronas se actualizan a la vez de modo síncrono estando la red en interacción con todas las ciudades) mientras que en el segundo se on-line (sólo una ciudad interacciona en cada instante con la red)”...

“Aunque según se observa los mejores resultados no se obtienen con el empleo de las redes neuronales, sin embargo el empleo de esta metodología es prometedor, principalmente por

dos razones:

a) *La implementación física de redes neuronales artificiales aumentará la velocidad computacional de estos algoritmos permitiendo una reducción en el tiempo de cálculo necesario para ofrecer una buena aproximación de la solución óptima.*

b) *Las redes neuronales poseen un gran potencial para resolver problemas aún más difíciles”*

Algoritmos genéticos

Los algoritmos genéticos, al igual que las redes neuronales, son un intento de reproducir procesos naturales. En este caso se trata de un modelo de resolución de problemas basado en la imitación de la evolución de una población. Esta población está formada por un conjunto de soluciones del problema a optimizar. Esta población se reproduce, es seleccionada y sufre mutaciones, imitando la evolución de las especies naturales.

“La premisa de los AGs, tras la publicación del libro de Holland[1975] ‘Adaptation in Natural and Artificial Systems’ y de los numerosos investigadores que los utilizan como metaheurística para optimización, es que se pueden encontrar soluciones aproximadas a problemas de gran complejidad computacional mediante un proceso de ‘evolución simulada’, en particular como un algoritmo matemático implementado en un ordenador.

Por otro lado, cierto número de investigadores ha reconocido que para algunos problemas de optimización existe una gran literatura de heurísticas rápidas de búsqueda iterativa(búsqueda local) . Han optado entonces por adaptar para esos problemas un método de búsqueda local (a lo mejor basado en metaheurísticas como la búsqueda tabú o recocido simulado) con un método basado en el uso de una población y la utilización de procesos de recombinación como los de los AGs. Ese tipo de metaheurísticas han sido clasificadas bajo la denominación común de ‘Algoritmos Meméticos’ (AM)

Las características de la evolución que hay que tener en cuenta están descritas brevemente por Davis [1991] de la siguiente manera:

1. *La Evolución es un proceso que opera en los cromosomas en lugar de en los seres vivos que ellos codifican.*
2. *Los procesos de selección natural provocan que aquellos cromosomas que codifican estructuras con éxito se reproduzcan más frecuentemente que aquellos que no lo hacen.*
3. *Las mutaciones pueden causar que los cromosomas de los hijos sean diferentes a los de los padres y los procesos de recombinación pueden crear cromosomas bastante diferentes en los hijos por la combinación de material genético de los cromosomas de los padres.*
4. *La evolución biológica no tiene memoria.”[OHyRN]*

Los componentes de una implementación de AG son:

- 1 Cromosomas: representa una solución o configuración del problema.
- 2 Una población inicial.
- 3 Una función de evaluación que permiten ordenar los cromosomas de acuerdo con la función objetivo.
- 4 Operadores genéticos que se utilizan como regla para el cruzamiento de los cromosomas de los padres (crossover).

- 5 Operaciones que se realizan sobre los cromosomas en las mutaciones.
- 6 Valores de los parámetros que el algoritmo genético usa: tamaño de la población, probabilidades asociadas con la aplicación de los crossover, probabilidades de las mutaciones).

Propuestas futuras

Experimentos con heurísticas sobre sistemas georreferenciados

El hecho de poseer información sobre la ubicación espacial de los puntos es una gran ventaja. Sin pensarlo demasiado nos damos cuenta que podríamos eliminar toda la información de las calles y detalles irrelevantes de la modelización de la ciudad y ver los puntos a recorrer como una red abstracta, en la que podemos llegar desde cualquier nodo a otro y calcular la distancia como distancia aéreas. De esta forma eliminaríamos una enorme cantidad de cálculos y su pérdida de tiempo asociado.

Esta simple idea es una heurística, que se basa en una serie de supuestos que pueden tener mayor o menor grado de cercanía con la realidad, como el hecho de que los puntos a recorrer no están dentro de islas, lo que imposibilitaría llegar a ellos desde cualquier otro.

Este obstáculo puede ser superado fácilmente si poseemos un buen sistema de información georreferenciada. De ser así, poseeríamos información en alguna capa (layer) sobre el barrio o el sector de la ciudad a la que pertenece cada nodo. Entonces, si sabemos que hay un grupo aislado lo resolvemos por separado y para el problema general lo vemos como un solo punto.

También, para aprovechar la característica de SIMPLEX de mejorar la solución en cada iteración, podríamos asignar un factor especial a los nodos dentro de ciertas áreas geográficas para lograr que el algoritmo se encamine más rápidamente o que no entre en alguna zona del mapa.

Se puede ver la serie de puntos a recorrer como una serie de datos a los que se desea aproximar alguna función, con características particulares, y en base a esta asignar pesos con factor o guiar al algoritmo de Dijkstra a recorrer ciertos lugares y eliminar otros.

En la vida práctica se podría implementar algún tipo de memoria para el algoritmo, si debe conectar dos puntos que ya calculó simplemente usa los datos anteriores.

Una vez que poseemos un diagrama general del orden para recorrer los puntos podemos aplicar algoritmos de camino más corto dentro de las áreas formadas por cada par de puntos del camino, que son de subredes más simples, y de esa forma lograr la descripción detallada del camino.

La cantidad de reglas para acelerar la solución del problema es suficientemente amplia para confundir, pero lo importante sería poseer alguna información sobre cuál es más adecuada para aplicar sobre redes que modelan ciudades. Dicho de otra forma, es necesario medir el grado de certeza que poseen las suposiciones en las que se basan las heurísticas que se utilizan para una determinada implementación, la necesidad de utilizarlas y la cercanía de éstas con las soluciones óptimas.

Sin lugar a dudas las investigaciones más interesantes son las que podrían realizarse en el campo de los algoritmos meméticos. Creo que podría ser muy prometedor generar la población inicial de un AG con mi algoritmo, luego introducir distintos tipos de mutaciones. Por ejemplo podrían utilizarse al azar el intercambio de ciudades una iteración de SIMPLEX y una iteración de Kohonen.

Conclusiones

Hay que tener siempre en mente que el objetivo del desarrollo de una aplicación de este tipo es proveer elementos de decisión a un experto humano. La velocidad es fundamental, más que una exactitud perfeccionista.

La representación a utilizar en un implementación debe depender de la velocidad y de la facilidad de mantenimiento asociada. No deberían influir en la decisión ningún otro tipo de factor como gustos personales del grupo de trabajo o tendencias tecnológicas del momento.

Una solución ideal sería proveer algoritmos de resolución rápida que puedan convivir con otros de menor velocidad y mayor calidad, de forma que el usuario del sistema obtenga una solución casi instantánea, y mientras el la observa el otro algoritmo la va corrigiendo.

Es de gran importancia realizar experimentos y medir la velocidad de los algoritmos, para poder tomar una decisión fundamentada en el momento de la implementación.

Fuentes de información

<p>[DAVIS] Tipo: Libro Título: Modelos Cuantitativos para administración.</p> <p>Autores: K. Roscoe Davis - Patrick G. McKeown</p> <p>Editorial: Grupo Editorial Iberoamérica Primera edición en inglés: 1984 Primera edición en castellano: 1986 Edición consultada: 1995</p>	<p>Conceptos de redes, arcos y nodos Representación matemática de redes</p>
<p>[DATOS] Tipo: Apunte Título: Apunte de la cátedra Sistemas de Datos</p> <p>Autores: Tymoschuk-Burdisso-Cora</p> <p>Edición: 1994</p>	<p>Representación matricial de grafos Representación enlazada de grafos</p>
<p>[CRN] Tipo: Curso presencial Título: Curso “Redes Neuronales”</p> <p>Disertante: Ing. Mario Mastrini</p> <p>Organizador: Ramas estudiantiles del IEEE de la Universidad Tecnológica Nacional y la Universidad Nacional de Córdoba</p> <p>Realización: 11 y 12 de octubre de 1996</p>	<p>Comentarios del disertante acerca de la posibilidad de desarrollar con Redes Neuronales soluciones a los problemas de Programación lineal y específicamente al problema del viajante.</p>

<p>[JSIG] Tipo: Curso presencial Titulo: II Jornadas sobre Sistemas de Información Georreferenciada</p> <p>Organizador: Facultad de Ciencia Económicas - UNC - Fundación Sol- Municipalidad de la ciudad de Córdoba</p> <p>Auspiciantes: Universidad Tecnológica Nacional- Consejo de Ciencia Informáticas de la Provincia</p> <p>Realización: 22,23 y 24 de mayo de 1995</p>	<p>Conceptos básicos sobre sistemas georreferenciados, su desarrollo, implementación y algunas aplicaciones a redes de distribución.</p>
<p>[TNB] Tipo: Libro Titulo: Estructuras de datos en C</p> <p>Autor: Tanenbaum</p> <p>Edición:</p>	<p>Definición de grafos y componentes.</p>
<p>[KSR] Tipo: Tesis doctoral Título: Polyhedral solution to the pickup and delivery problem</p> <p>Autor: Kevin Scott Ruland.</p> <p>Edición: Agosto 1995</p> <p>Consultar en : http://rodin.wustl.edu/~kevin/dissert/dissert.htm</p> <p>1</p>	<p>Conceptos de redes, arcos y nodos Representación matemática de redes</p>

<p>[OHyRN] Tipo: Libro Título: Optimización heurística y redes Neuronales en Dirección de Operaciones e Ingeniería</p> <p>Autores: Adenso Díaz Fred Glover Hassan M. Ghaziri J.L.Gonzáles. Manuel Laguna Pablo Moscato Fan T. Tseng</p> <p>Editorial: Parafino Edición: 1996</p>	<p>Definición de heurísticas. Redes Neuronales y Algoritmos genéticos.</p>
<p>[TSPBIB] Tipo: Página Web Título: TSP Bible</p> <p>Autor: Pablo Moscato</p> <p>Consultar en: http://www.ing.unlp.edu.ar/cetad/mos/TSPBIB_home.html</p>	<p>Referencias a las últimas investigaciones. Encontré [KSR] Encontré [OhyRN]</p>
<p>[AGyLD] Tipo: Curso presencial Título: Curso “Algoritmos genéticos y lógica difusa”</p> <p>Disertante: Ing. Mario Mastrini</p> <p>Organizador: Ramas estudiantiles del IEEE de la Universidad Tecnológica Nacional</p> <p>Realización: 23, 24 y 25 de octubre de 1997</p>	<p>Concepto básicos sobre Algoritmos genéticos y la resolución de TSP.</p>

<p>[NP] Tipo: Paper Título: External problems concerning N-Path connected graphs</p> <p>Autores: F. Escalante L. Montejano</p> <p>Editorial: Centro de investigaciones en matemáticas aplicadas y en sistemas, Universidad Nacional de México.</p> <p>Publicación: Comunicaciones técnicas, Volúmen 4, Número 60, Serie B, 1973</p>	<p>Trata problemas de la conectividad de nodos en grafos.</p>
<p>[BEA] Tipo: Paper Título: The traveling salesman problem and the bound energy algorithm used in cluster analysis</p> <p>Autores: Raúl Caravajal Guillermo Espinosa Arturo López</p> <p>Editorial: Centro de investigaciones en matemáticas aplicadas y en sistemas, Universidad Nacional de México.</p> <p>Publicación: Comunicaciones técnicas, Volúmen 5, Número 79, Serie B, 1974</p>	<p>Demuestra que el algoritmo BEA es una buena heurística para TSP.</p>

Si bien no todas las fuentes son citadas en el texto, todas las que se mencionan han sido consideradas, durante la investigación. Cuide de mencionar aquellas que son de más fácil acceso para quién desee continuar la investigación y intente implementar el algoritmo. Las citas textuales fueron seleccionadas por su claridad.

Nota del asesor

Córdoba, 4 de marzo de 1998.

Sr. Director
Cátedra de Trabajo Final.

De mi consideración:

En mi carácter de asesor en el presente trabajo realizado por el alumno Francisco G. Simó, Legajo N° 24383-6, para la obtención del título intermedio de Analista Universitario de Sistemas, certifico haberlo asesorado y encaminado en la construcción de dicho trabajo, así como también de contribuir en la consulta de material.

Después de la revisión, la he encontrado satisfactorio en su contenido y tratamiento de la información.

Quedando a su disposición, saluda atentamente.

Ing. en Sistemas de Información
Sergio H. Castro

Índice de referencias cruzadas

algoritmos genéticos (21)

Arcos

 Dirigidos (7, 11)

 No dirigidos (7)

Bellman (13)

Camino (8)

 algoritmo de camino más corto (13)

Dijkstra (13)

Distancias (8)

Experimentos (23)

Ford (13)

Fuente de información (5)

 CRN (25, 26)

Georeferencia (7)

grafo (7)

Holland (21)

limitaciones (5)

MTSP (19)

Nodos

 destino final (12)

 proveedores (11)

 transporte (12)

 un conjunto formado (11)

Objetivo (5)

Pesos (7)

prejuicio (5)

problema del viajante (6)

 complejidad (6)

 objetivo (6)

 programación lineal (11)

problemas de transbordo (11)

Red (7)

 un conjunto formado (11)

redes neuronales (20)

representación de la ciudad (7)

 layer (23)

supuesto (15)