

Algoritmo de Dijkstra. Un Tutorial Interactivo

Gloria Sánchez Torrubia
Dept. de Matemática Aplicada
Facultad de Informática
Universidad Politécnica de Madrid
e-mail: gsanchez@fi.upm.es

Víctor M. Lozano Terrazas
Facultad de Informática
Universidad Politécnica de Madrid
Madrid

Resumen

En este artículo se presenta un tutorial realizado como un applet en Java2 con el cual los alumnos puedan ejercitarse, vía Web, en el manejo del algoritmo de Dijkstra [3]. Se expone la relevancia de dicho algoritmo en la enseñanza de la informática debido a la importancia de sus aplicaciones y, por último, se analiza la influencia sobre el aprendizaje que supone, por una parte la fácil accesibilidad del entorno Web, y por otra la interactividad de este tutorial.

1. Introducción

Está claro que es necesaria la introducción de las nuevas tecnologías en la enseñanza. En este contexto no se deben olvidar las posibilidades de Internet, tanto por la accesibilidad que supone, como por la posibilidad que da al alumno, si se crea el entorno adecuado, de practicar los conocimientos adquiridos en el aula.

Es con este doble objetivo con el que se ha creado esta herramienta. El desarrollo, utilizando la última tecnología en lenguaje Java (el Java2), ha dado lugar a un tutorial ágil, visual y fácil de usar con el cual los alumnos puedan ejercitarse, vía Web, en el manejo del algoritmo de Dijkstra. Gracias a la interactividad que este tutorial supone, se crea un dinamismo que agiliza y optimiza el estudio además de conseguir que el alumno eleve su interés por el aprendizaje.

La herramienta ha sido diseñada como núcleo de un Proyecto Fin de Carrera [6] en la titulación de Licenciado en Informática e incluida dentro de la creación de una serie de tutoriales, que se van colocando en la página Web del Departamento de

Matemática Aplicada de la Facultad de Informática (U.P.M.) a medida que se van realizando, con el fin de complementar la enseñanza de las asignaturas impartidas por el departamento.

El ejemplo que nos ocupa: el algoritmo de Dijkstra, se estudia, como parte del tema de grafos, en la asignatura de Matemática Discreta (asignatura troncal de 7,5 créditos que se imparte en el primer cuatrimestre de primer curso).

Las razones que han impulsado a la elección del algoritmo de Dijkstra para la creación de este tutorial son: por una parte las numerosas e importantes aplicaciones de este algoritmo, que hacen de él uno de los más relevantes en la teoría de grafos, aumentando así la importancia de su aprendizaje, y por otra, la relativa dificultad de su aplicación, teniendo en cuenta que va dirigido a alumnos de primer curso.

La facilidad de utilización del tutorial, que no requiere más conocimientos previos que ideas básicas de grafos y una explicación teórica del algoritmo, animará y ayudará a los alumnos en la tarea de comprenderlo y manejarlo.

2. Algoritmo de Dijkstra

2.1. Explicación del algoritmo

Dado un grafo a cuyos arcos se han asociado una serie de pesos, se define el camino de coste mínimo de un vértice u a otro v , como el camino donde la suma de los pesos de los arcos que lo forman es la más baja entre las de todos los caminos posibles de u a v .

El algoritmo de Dijkstra es un algoritmo eficiente (de complejidad $O(n^2)$ donde n es el número de vértices) que sirve para encontrar el

camino de coste mínimo desde un nodo origen a todos los demás nodos del grafo. Fue diseñado por el holandés Edsger Wybe Dijkstra en 1959.

El fundamento sobre el que se asienta este algoritmo es el principio de optimalidad: si el camino más corto entre los vértices u y v pasa por el vértice w , entonces la parte del camino que va de w a v debe ser el camino más corto entre todos los caminos que van de w a v (Figura 1). De esta manera, se van construyendo sucesivamente los caminos de coste mínimo desde un vértice inicial hasta cada uno de los vértices del grafo, y se utilizan los caminos conseguidos como parte de los nuevos caminos.

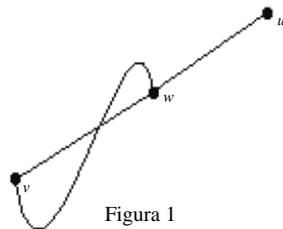


Figura 1

Vamos a explicar el algoritmo con un pseudocódigo sencillo que nos ayudará a comprender como funciona.

- $G = (V, E)$ donde V es el conjunto de vértices y E el de arcos.
- S es el conjunto de vértices cuyos caminos más cortos al origen han sido ya determinados.
- $V-S$ es el resto de vértices.
- d : ARRAY de estimaciones de caminos más cortos a dichos vértices.
- pr : ARRAY de predecesores para cada vértice.

```

<<Inicializar  $d$  y  $pr$ >>
<<Poner  $S = \emptyset$ >> // aún no hemos estudiado
ningún vértice
While  $\{V-S\} \neq \emptyset$  // mientras queden
nodos sin determinar su camino mínimo al
origen
  <<Ordenar los vértices en  $V-S$  y
  analizar de acuerdo a la menor
  distancia al origen>>
  <<Añadir  $u$ , el vértice más cercano en
   $V-S$ , a  $S$ >> //  $S = S + \{u\}$ 
  <<Recalcular la distancia a todos los
  vértices todavía en  $V-S$  adyacentes a
   $u$ >>

```

2.2. Aplicaciones del algoritmo

Las aplicaciones del algoritmo de Dijkstra son muy diversas y de gran importancia en distintas áreas del conocimiento. Vamos a presentar algunas de ellas.

- *Encaminamiento de paquetes por los routers* [10]: Consideremos una red telefónica. En un momento dado, un mensaje puede tardar una cierta cantidad de tiempo en atravesar cada línea (debido a efectos de congestión, retrasos en las conexiones etc.). En este caso tenemos una red con costes en los arcos y dos nodos especiales: el nodo de comienzo y el de finalización, el objetivo aquí es encontrar un camino entre estos dos nodos cuyo coste total sea el mínimo.
- *Aplicaciones para Sistemas de información geográficos: extracción de características curvilíneas de imágenes usando técnicas de minimización del camino* [4]: La imagen se representa como una matriz de puntos, cada uno con una especial intensidad. Cada nodo se corresponde con un punto (pixel) de la imagen y tiene hasta ocho nodos adyacentes. El peso de los arcos viene dado en este caso por la diferencia de intensidad. Esta técnica presenta un gran ahorro de costes frente a las herramientas existentes actualmente en el mercado que usan métodos de vectorización automáticos.
- *Caminos mínimos en Grafos usando XML y parsers de Java* [7]: El concepto de camino es una secuencia de operadores y conectores: un **operador** será cualquier unidad de proceso de información realizando un algoritmo específico (i.e. conversores digitales, de formato etc.) y un **conector** será cualquier mecanismo a través del cual los operadores se comunican entre sí. Dado un conjunto de descripciones de operadores y conectores, unos parámetros de **optimización** (que el usuario queda encargado de introducir) y una serie de requisitos, el sistema se encargará de encontrar un camino óptimo de una entrada establecida hasta un tipo de salida especificada aplicando transformaciones específicas en el menor tiempo posible.
- *Reconocimiento de lenguaje hablado* [8]: Un problema que se presenta es el distinguir

entre palabras que suenan de manera similar. Se puede construir un grafo cuyos vértices correspondan a palabras posibles y cuyos arcos unan palabras que puedan ir colocadas una al lado de otra. Si el peso del arco corresponde a la probabilidad de que estén así colocadas, el camino más corto en el grafo será la mejor interpretación de la frase.

- *Otras aplicaciones.* Enrutamiento de aviones y tráfico aéreo. Tratamiento de imágenes médicas. Problemas de optimización de una función de coste para moverse entre diversas posiciones.

3. Uso y explicación del tutorial

Al tutorial se accede dentro de la página Web del Departamento de Matemática Aplicada de la Facultad de Informática (U.P.M.), en la dirección: <http://www.dma.fi.upm.es/dijkstra/>

Ésta es una página de introducción que da acceso, por una parte al manual de uso y por otra a la página que presenta el interfaz de usuario del applet.

El programa está diseñado para permitir al usuario dibujar sus propios grafos haciendo uso del ratón y del teclado. Así mismo, y para facilitar el aprendizaje del algoritmo, será posible hacer uso de un ejemplo predefinido que ilustrará el funcionamiento del mismo sin necesidad de proceder a la introducción de los nodos y las aristas.

3.1. Introducción del grafo

Los nodos se dibujan mediante el botón izquierdo del ratón. El primer nodo será el origen del grafo y se visualizará en color negro. Los demás irán en azul.

Las aristas se introducen posicionando el puntero en el nodo origen de la arista, presionando el botón izquierdo y manteniéndolo pulsado hasta que se alcance el nodo destino. Se introduce en ese momento el peso a través del teclado y se presiona Enter.

3.2. Ejecución del algoritmo.

Una vez introducidos los nodos y las aristas con sus respectivos pesos se procede a la

ejecución del algoritmo. En la ejecución se presentan dos posibilidades: *paso a paso* y *completa*

Al pulsar el botón de ejecución completa (Figura 2) la resolución se completa sin pasos intermedios desde el punto en que se encuentre la ejecución, mostrando el camino de coste mínimo del nodo origen a cada uno de los nodos del grafo. Esta opción tiene carácter menos educativo que la ejecución paso a paso, pero permite a un usuario avanzado comprobar el resultado obtenido.

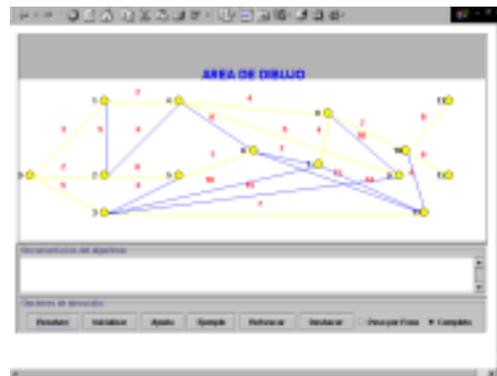


Figura 2

Al pulsar el botón de ejecución paso a paso (Figura 3) se ejecuta un ciclo del algoritmo pudiéndose de esta manera comprobar el efecto que se produce de forma gráfica gracias al

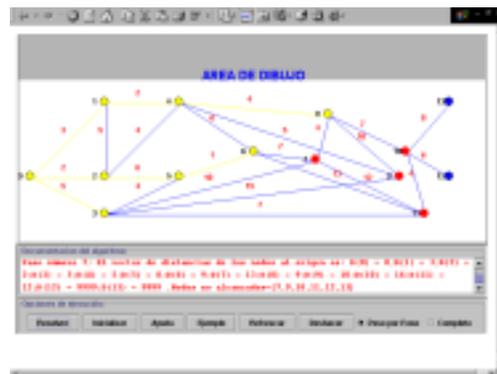


Figura 3

repintado de las aristas y de los nodos, tanto de los que se encuentran en proceso (en rojo), como de los ya alcanzados (en amarillo).

Tras la ejecución de cada paso se visualizará en el área de texto de la consola del applet, el estado de las variables internas utilizadas en el proceso de resolución del algoritmo indicando el paso actual, el vector de distancias al origen y los nodos no alcanzados, facilitando de esta forma el seguimiento de la resolución del algoritmo por parte del alumno.

El applet presenta además en la ventana de documentación del algoritmo una pequeña documentación de ayuda que el alumno podrá consultar durante el proceso de dibujado y resolución del algoritmo, que se limita al uso de la herramienta, no al método de ejecución del algoritmo.

Se recomienda en todo caso que el alumno no se enfrente al uso de este tutorial sin tener un conocimiento previo del algoritmo de Dijkstra, pues la información que presenta sólo es útil después de una sesión teórica de inicialización en la materia.

4. Valor educativo del tutorial.

Internet se está convirtiendo en un importante recurso educativo gracias a la capacidad que nos ofrece para complementar las clases de formato tradicional en el entorno del aula con la posibilidad de permitir el acceso a la información de forma remota. Las limitaciones de tiempo y lugar han desaparecido. Además de esa posibilidad, es importante tener en cuenta el efecto que la interactividad que puede proporcionar una herramienta informática tiene en el proceso de aprendizaje. En este sentido se sabe que la interactividad es un atributo importante y probablemente esencial de cualquier técnica de aprendizaje que pretenda el éxito, que el aprendizaje es más efectivo cuando el estudiante puede controlar el intercambio de información.

Estudios realizados a principios de los noventa en la universidad de Deakin (Australia) [5] mostraron que los estudiantes que habían utilizado técnicas de aprendizaje interactivas y multimedia tuvieron un 55% ganado sobre aquellos estudiantes que recibieron las clases en el entorno tradicional de enseñanza. Otras cifras significativas demostraron que los alumnos aprendieron el material un 60% más rápido y la

retención del conocimiento después de 30 días fue entre un 25 y un 50% superior.

En otro estudio, realizado en la universidad de Acadia (Canadá) [11] sobre alumnos de un curso introductorio de física en primer curso de enseñanza universitaria, se puede comprobar que tanto los resultados como la satisfacción al terminar el curso de los estudiantes que utilizaron herramientas interactivas fueron claramente superiores a los de aquellos que recibieron el curso con técnicas clásicas.

Por otra parte, la experiencia en la enseñanza de las matemáticas nos demuestra que es necesaria la práctica por parte del alumno de los métodos que se enseñan de manera teórica en el aula. También sabemos que esa práctica implica unas sesiones, a menudo tediosas, de estudio para el alumno, mientras que la motivación de éste suele ser mayor al enfrentarse a una herramienta de tipo interactivo, lo cual influye positivamente en el proceso de aprendizaje.

Además de lo anteriormente citado, cabe también destacar la utilidad que puede tener esta herramienta en el propio aula al permitir al profesor la utilización de otros materiales distintos a los habituales y conseguir de ese modo agilizar las clases y atraer la atención de los alumnos. En este sentido he podido comprobar que al utilizar el tutorial dentro del aula, aumentaba la atención y mejoraba la comprensión y por tanto el aprendizaje.

5. Conclusión

Comprender los beneficios e importancia de la interactividad en el proceso de aprendizaje puede mejorar la calidad de los materiales de enseñanza, animar a los profesores a utilizar las herramientas que la Web pone a su disposición y aportar suficientes razones para persuadir a los que aún permanecen escépticos sobre las ventajas que este tipo de aprendizaje aporta.

Las personas recordamos el 20% de lo que vemos, el 40% de los que vemos y escuchamos y el 70% de lo que vemos, escuchamos y realizamos.

Como parte de nuestros esfuerzos en la mejora de la enseñanza, debemos aplicar todos los recursos que nos proporciona la tecnología para desarrollar de la mejor manera posible la capacidad y conocimientos de nuestros

estudiantes. En este sentido, no sólo es necesaria la utilización de los recursos disponibles, sino que exige un cambio sustancial en la manera de enfocar las clases. Éste es, en mi opinión, el mayor reto profesional que se plantea en este momento. Por tanto, debemos redoblar nuestros esfuerzos para integrar estas poderosas herramientas de modo que facilite y mejore el aprendizaje de nuestros alumnos.

En este contexto considero el tutorial como una herramienta más que sin duda ayudará en la tarea de enseñar de manera más agradable y efectiva el algoritmo de Dijkstra.

Referencias

- [1] M. Abellanas y D. Lodaes: *Análisis de Algoritmos y Teoría de Grafos*. Rama 1990.
- [2] G. Chartrand, O. R. Oellermann *Applied and algorithmic graph theory* New York : McGraw-Hill, 1993
- [3] E. W. Dijkstra: *A note on two problems in connection with graphs*. Numerische Mathematik, vol. 1, pp. 269-271, 1959
- [4] M.R. Dobie y P. H. Lewis: *Extracting curvilinear Features from Remotely Sensed Images Using Minimum Cost Path Techniques*. <http://www.mmrg.ecs.soton.ac.uk/publications/archive/dobie1994/html/> Universidad de Leeds. U.K.
- [5] Street S. y Goodman A.: *Some Experimental Evidence on the Educational Value of Interactive Java Applets in Web-based Tutorials*. Third Australasian Conference on Computer Science Education, Association for Computing Machinery, 1998 "Brisbane, Australia", pp. 94-100
- [6] Víctor M. Lozano Terrazas *Algoritmo de Dijkstra: tutorial interactivo en Java 2*. Madrid. Facultad de Informática. U.P.M. 2000
- [7] LindaMay Patterson: *Using the IBM XML Parser Find and replace elements in an XML Document* http://www.as400.ibm.com/developer/java/xml/xml_parser2.html IBM XML Parser 2000.
- [8] Steven S. Skiena: *Shortest Path* <http://www.cs.sunysb.edu/~algorithm/files/shortest-path.shtml>. Stone Brook (State University of New York). Computer Science Department, 2001.
- [9] M.N.S. Swamy, K Thulasiraman: *Graphs, networks, and algorithms*. New York : John Wiley & Sons, 1981
- [10] Michael A. Trick: *Shortest Path* <http://mat.gsia.cmu.edu/classes/networks/node4.html> Carnegie Mellon University. Computer Science Department. 1998
- [11] P. Williams, C. MacLachy, P. Backman y D. Retson . *Studio physics report on the Acadia*: <http://www.acadiau.ca/advantage/physics.htm> Department of Physics, Acadia University 1996