
ProRouting: una herramienta para la visualización de grafos de proximidad y estrategias de ruteo *

J. M. Gil¹ y G. Hernández¹

Facultad de Informática, Universidad Politécnica de Madrid
joscargil@gmail.com, gregorio@fi.upm.es

Resumen. Se describe la aplicación informática **ProRouting** que visualiza los grafos de proximidad más conocidos sobre una nube de puntos en el plano y las estrategias locales de ruteo. También calcula la dilación de los grafos construidos y mide la eficiencia de las estrategias de ruteo. La aplicación se ha desarrollado en Java.

Palabras clave: Grafos de proximidad, algoritmos locales, ruteo, dilación.

1 Introducción

Un grafo geométrico es un grafo cuyos vértices son puntos del plano (o de un espacio n -dimensional) y cuyas aristas son segmentos uniendo los pares de vértices adyacentes. Las redes de carreteras, de ferrocarril, de computadores pueden modelarse como grafos o redes geométricas. Entre estas redes son de especial interés los grafos de proximidad, que se definen a partir de diferentes medidas de la cercanía entre sus vértices. Desde el artículo de Toussaint de 1980 sobre el *grafo de vecindad relativa*, [7], se han propuesto muchas nociones de vecindad o proximidad, que permiten capturar la forma y estructura de los conjuntos de puntos siendo así herramientas indispensables de la denominada *morfología computacional*. Así fueron apareciendo los *grafos de vecindad relativa*, *grafos de Gabriel*, *β -esqueletos*, *γ -grafos*, *grafos de Morelia*, etc.

Por otra parte, las redes inalámbricas han tenido un espectacular auge en los últimos años (redes celulares, de sensores para monitorización, redes ad-hoc, etc.) apareciendo numerosos problemas sobre la construcción, organización y manejo de estas redes, en la que cada nodo representa una estación emisora con un rango de transmisión uniforme. Así estas redes toman como modelo básico el denominado grafo unidad (*unit disk graph*, *UDG*) en el que hay una arista entre dos nodos si su distancia euclídea es menor o igual a 1. Pero este grafo puede tener hasta $O(n^2)$ aristas y no es planar por lo que se precisa construir subgrafos que cumplan unas buenas condiciones: conexo

* Trabajo financiado parcialmente por la UPM, proyecto AL08-PAC-16

si lo es *UDG*, cantidad lineal de aristas, grado acotado, planaridad y que sea una buena aproximación en distancia de *UDG*. Esta condición significa que el subgrafo debe ser un *spanner* de *UDG*. Un subgrafo G' es un *spanner* de G si existe $t > 0$ tal que para todo par de nodos u, v la distancia entre ellos en G' (longitud del camino más corto en G') es a lo sumo t veces la distancia en G . Si G es el grafo euclídeo completo al factor t se le denomina **dilación** de G' . La búsqueda de subgrafos de *UDG* con esas buenas propiedades ha dado lugar a nuevos grafos de proximidad como los *grafos de Yao*, *θ -grafos*, *halfspace proximal graphs*, *Theta Lambda graphs*, *grafos Delaunay locales*, etc. En el reciente libro de Narasimhan y Smid ([5]), se encuentra abundante información sobre *spanners* geométricos.

En las redes inalámbricas aparecen nuevas condiciones que imponen nuevas soluciones a problemas clásicos. Las redes de comunicaciones son por su propia naturaleza más dinámicas, los nodos aparecen y desaparecen constantemente. En redes como las de telefonía móvil, los sensores se mueven constantemente cambiando así la topología de la red. En consecuencia, la utilización de algoritmos como el de Dijkstra para calcular caminos mínimos no es válida en muchos problemas de la vida real. Los algoritmos adecuados para trabajar con estas redes son *algoritmos locales* (ver [8]) en los que cada nodo de la red sólo posee información sobre sus vecinos y no se utiliza ningún tipo de conocimiento global de toda la red. Las estrategias locales para resolver el problema de **ruteo** en redes geométricas han sido uno de los retos fundamentales en el área. (Ver por ejemplo, [1], [2], [3] y [4])

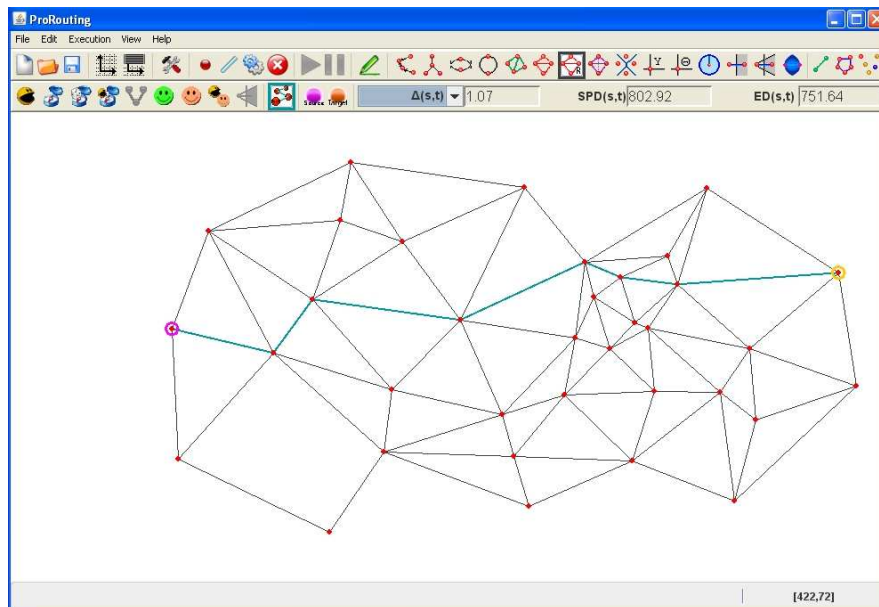


Figura 1. Una vista de la aplicación.

En este trabajo presentamos una herramienta informática, **ProRouting**, que tiene una doble vertiente: en primer lugar sirve para visualizar grafos de proximidad correspondientes a un conjunto de puntos del plano y en segundo lugar visualiza unas cuantas estrategias locales de ruteo en esos grafos geométricos. Los objetivos también son dos, por una parte que sirva como herramienta didáctica para visualizar paso a paso la ejecución de los algoritmos de construcción de grafos y estrategias de ruteo y, por otra parte, que sirva como ayuda a la investigación en propiedades de los grafos geométricos y la búsqueda de estrategias locales eficientes en redes geométricas.

ProRouting se ha desarrollado sobre Java en varias etapas a lo largo de las cuales se han añadido funcionalidades y componentes. Los primeros pasos en la aplicación fueron dados por los alumnos de la Facultad de Informática Rubn Naranjo, David Ramos y Damián Serrano. ProRouting es un software de libre disposición, [6].

2 Descripción de la aplicación

Esta aplicación visualiza las diferentes estructuras de proximidad definidas sobre un conjunto de puntos en el plano. Además permite el trazado de rutas en dichos grafos siguiendo criterios estrictamente locales, es decir, que no presuponen un conocimiento completo del grafo sobre el que se trazan. Se ha implementado en el lenguaje de programación Java y puede utilizarse en cualquier sistema que tenga instalada una Máquina Virtual de Java (JVM).

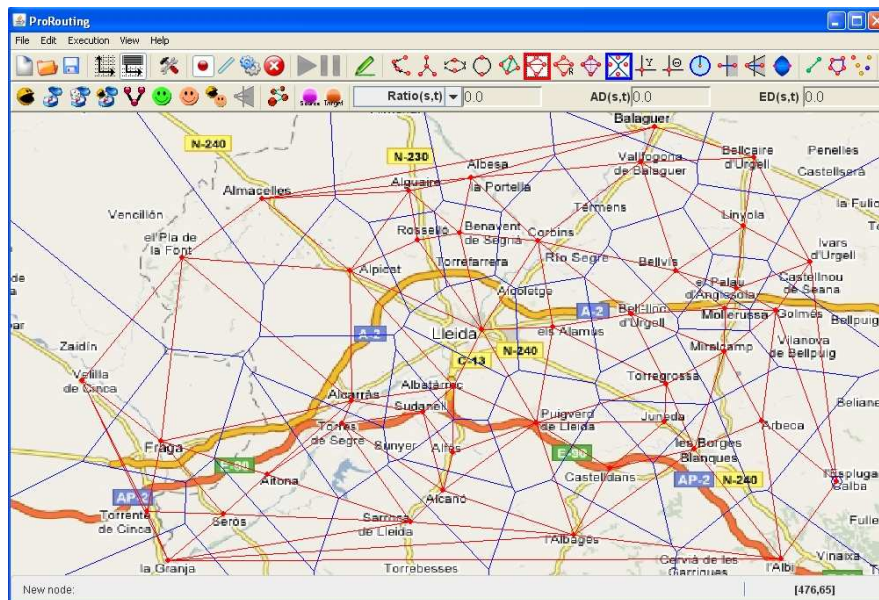


Figura 2. El diagrama de Voronoi en Lleida.

La aplicación consta de los siguientes elementos:

- (1) una barra de menús,
- (2) dos barras de herramientas: una para dibujo de grafos, manejo de ficheros y utilidades de configuración y otra con el trazado de rutas e información numérica sobre la dilación del grafo y la calidad de la ruta conseguida,
- (3) una zona de dibujo y
- (4) una barra de estado para mensajes.

2.1 Introducción y manipulación de datos

Los puntos se pueden introducir manualmente con el ratón o automáticamente con un generador aleatorio. También puede cargarse desde archivo un conjunto de puntos previamente guardado con la extensión `.grp`. También se pueden insertar manualmente aristas. Como fondo de la zona de dibujo podemos cargar una imagen gráfica en formato gif, jpg o png. Tras la introducción de los datos se pueden manipular los puntos, insertando nuevos puntos, moviéndolos o eliminándolos. Todas las operaciones se efectúan en tiempo real, permitiendo al usuario comprobar cómo se modifican los grafos de proximidad y las rutas al cambiar de posición un punto o suprimirlo.

2.2 Estructuras geométricas

Esta aplicación calcula y visualiza los siguientes grafos geométricos asociados a un conjunto de puntos, algunos de ellos con parámetros ajustables desde la ventana de configuración:

- Cierre convexo
- Par más próximo
- Grafo del vecino más cercano
- árbol generador mínimo euclídeo
- Grafo de vecindad relativa
- Grafo de Gabriel (figura 3)
- β -esqueletos
- Triangulación incremental
- Triangulación de Delaunay y Diagrama de Voronoi (figura 2)
- Triangulación voraz
- Grafos de Yao
- θ -grafos
- Grafo Unidad (UDG) (figura 4)
- Half Space Proximal graph
- Familia de grafos $G(\theta, \lambda)$ (figura 7)

En cualquiera de estos grafos geométricos se puede calcular y mostrar la dilación entre un par de vértices, $\Delta(s, t)$ y la dilación del grafo, $\Delta(G)$. Para

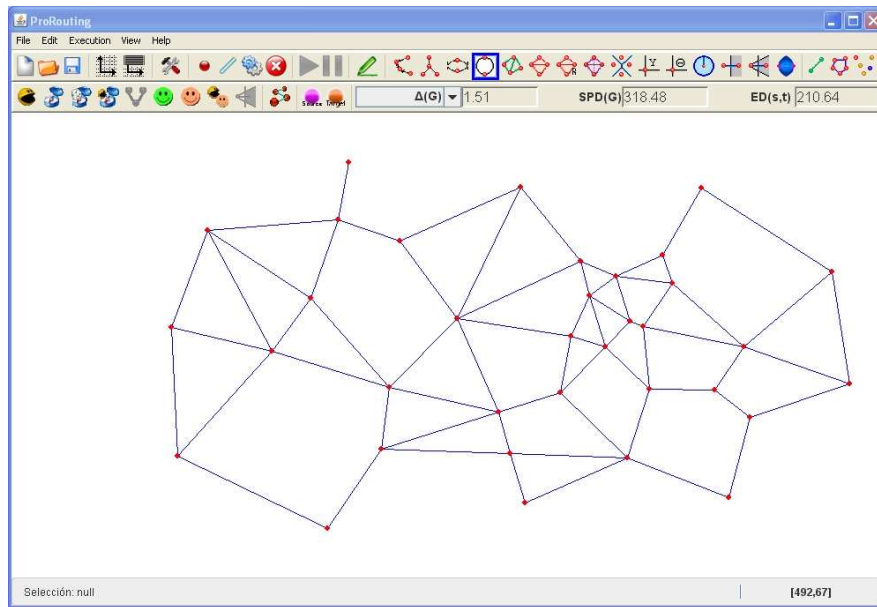


Figura 3. El grafo de Gabriel.

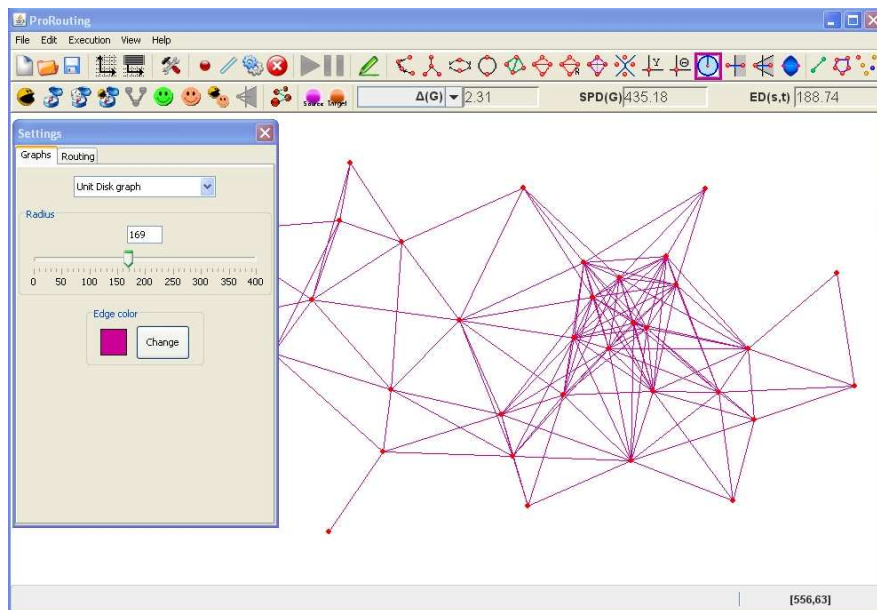


Figura 4. El grafo unidad (UDG) para un disco de radio 169.

ello se ha implementado también el algoritmo de Dijkstra, visualizándose el camino mínimo entre dos vértices cualesquiera, (figura 1).

2.3 Algoritmos de ruteo

Las estrategias locales de ruteo en un grafo geométrico buscan un camino entre dos vértices de forma que la decisión en cada vértice es totalmente local. Hay estrategias en las que el agente móvil carece de memoria, mientras que en otras se le permite memoria constante. Algunas estrategias no garantizan el éxito, es decir, no siempre se alcanza el destino. Los algoritmos implementados son:

- Ruteo voraz (greedy routing)(figura 5)
- Ruteo por brújula (compass routing)
- Aleatorizado por brújula (random compass routing) (figura 6)
- Híbrido voraz-brújula (hybrid compass-greedy)
- Ruteo Voronoi (Voronoi routing) (sólo para triangulaciones de Delaunay)
- Ruteo por caras (face routing)
- Ruteo por caras 2 (face routing 2)
- Ruteo Voraz-Caras-Voraz (greedy-face-greedy) (figura 5)
- Ruteo Theta-Lambda (sólo válido para los grafos $G(\theta, \lambda)$ (figura 7)

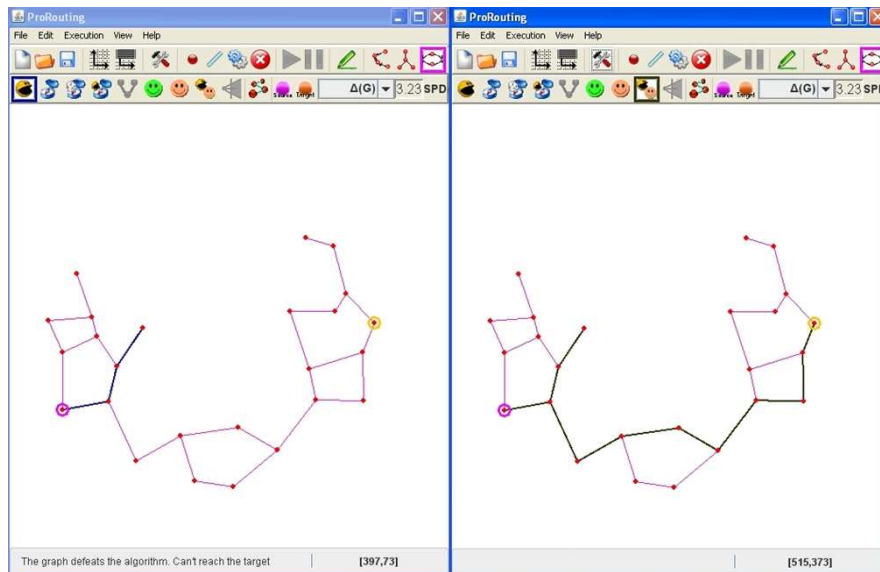


Figura 5. El ruteo voraz puede fallar, pero si se combina con el ruteo por caras siempre alcanza el objetivo.

En todos los algoritmos se calcula para cada par de vértices s y t , la calidad del camino encontrado, $Ratio(s, t)$. Ésta es el cociente entre la longitud del camino determinado por el algoritmo y la distancia euclídea entre s y t . Así se

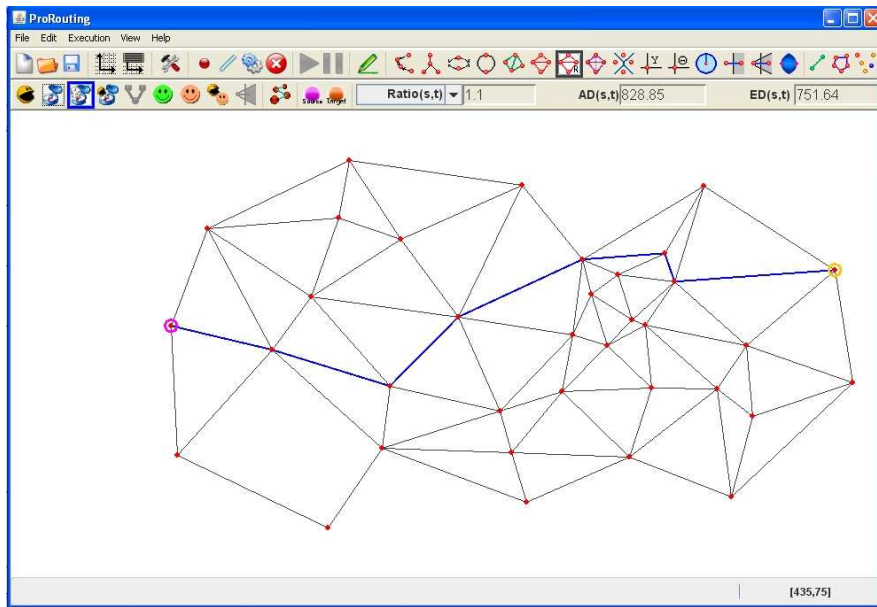


Figura 6. Algoritmo aleatorizado de ruteo por brújula.

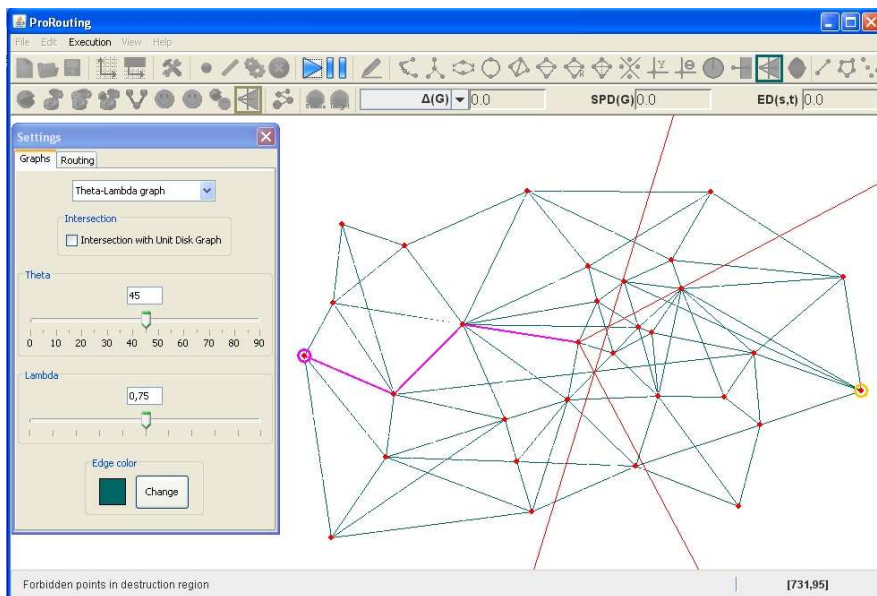


Figura 7. El grafo $G(\theta, \lambda)$ para los valores $\theta = \pi/4$, $\lambda = 3/4$ y la ruta correspondiente en un paso intermedio de su construcción.

pueden realizar estudios comparativos de la calidad de las soluciones obtenidas por los diferentes algoritmos.

3 Conclusiones

ProRouting es una herramienta de visualización de grafos geométricos y de estrategias de ruteo, útil tanto para la didáctica de estos temas como para su investigación. La aplicación está en continuo desarrollo, actualmente se están incorporando algunas variantes de los θ -grafos con las que se consiguen *spanners* de grado y diámetro acotados. También está prevista la mejora del sistema en cuanto a la entrada y salida de datos.

Referencias

- [1] P. Bose and P. Morin. Online routing in triangulations. *SIAM J. Comput.*, 33(4):937–951, 2004.
- [2] P. Bose, P. Morin, I. Stojmenovic and J. Urrutia. Routing with guaranteed delivery in ad-hoc wireless networks. *Wireless Networks*, 7(6):609–616, 2001.
- [3] E. Kranakis, H. Singh and J. Urrutia. Compass routing on geometric networks. *Proc. 11th Canadian Conf. on Comp. Geom.*, 1999.
- [4] X.-Y. Li and Y. Wang. *Wireless Sensors Networks and Computational Geometry, in "Handbook of Sensor Networks"*. ed. M. Ilyas et al. CRC Press, 2003.
- [5] G. Narasimhan and M. Smid. *Geometric Spanner Networks*. Cambridge University Press, 2007.
- [6] ProRouting. 2007. Disponible en <http://www.dma.fi.upm.es/gregorio/ProRouting>.
- [7] G. T. Toussaint. The relative neighbourhood graph of a finite planar set. *Pattern Recognition*, 12:261–268, 1980.
- [8] J. Urrutia. Local solutions for global problems in wireless networks. *J. of Discrete Alg.*, 5:395–407, 2007.