

Visualización de Grafos

Sergio R. Martig Silvia M. Castro
Dpto. de Ciencias de la Computación
Universidad Nacional del Sur – Bahía Blanca
{smartig,smc @cs.uns.edu.ar}

1 Introducción

El dibujado de grafos direcciona el problema de visualizar información estructural o relacional construyendo representaciones visuales geométricas de grafos o redes que son los modelos subyacentes en una gran cantidad de datos abstractos.

La generación automática del dibujo de un grafo tiene importancia en aplicaciones claves tales como la Ingeniería de Software, la Visualización de Información (VI), Data Mining, el diseño de Bases de Datos e Interfases Visuales, la Representación del conocimiento y las Telecomunicaciones entre otros dominios.

Las representaciones pictóricas de grafos han sido una herramienta de modelado tradicional; fueron usadas intensivamente en forma manual y aún hoy siguen utilizándose de este modo. En la década del '60, los científicos en computación comenzaron a utilizar los dibujos de grafos como herramienta para el desarrollo de software. Knuth [Knu63] fue quizás el primero en presentar un algoritmo para dibujar grafos con propósitos de visualización. Actualmente, existen herramientas que permiten que muchas aplicaciones se beneficien, en alguna medida, de la generación automática de dibujos de grafos.

Debido a la naturaleza combinatoria y geométrica de los problemas relacionados y al amplio espectro de los dominios de aplicación, la investigación sobre dibujado y visualización de grafos se ha desarrollado dentro de distintas áreas como la Matemática Discreta (teoría topológica de grafos, teoría geométrica de grafos, teoría de orden), el Diseño de Interfaces (lenguajes visuales, interfaces gráficas, visualización de software) y los Algoritmos (algoritmos de grafos, estructuras de datos, geometría computacional, sistemas de VI).

Este trabajo presenta una descripción de la investigación realizada hasta el momento en visualización de grafos. En la próxima sección se detallan los elementos esenciales a tener en cuenta en el dibujado de los grafos con el objetivo de obtener una visualización efectiva; a continuación se hace una breve reseña y análisis de los distintos métodos para el dibujado de los mismos, concluyendo con una descripción del trabajo realizado hasta el momento.

2 Elementos esenciales en la visualización de grafos

El problema de dibujado de un grafo puede plantearse simplemente del siguiente modo:

Dado un conjunto de nodos y un conjunto de arcos que representan las relaciones entre los nodos, obtener la posición de los nodos y la curva que debe ser dibujada para cada arco.

La utilidad de una representación dada depende de diversos factores y hay varios criterios considerados estándar que se tienen en cuenta en la visualización de grafos. El *tamaño* del mismo juega un rol preponderante cuando se lo quiere visualizar: no pueden usarse los mismos criterios de dibujado cuando el grafo es chico que cuando tiene miles de nodos. Usualmente, los vértices son representados por símbolos como cajas o puntos y las aristas por curvas abiertas conectando los símbolos que representan a los vértices asociados. Sin embargo estos estándares varían, sin duda, según la aplicación. Por otro lado, si la cantidad de elementos es grande, es decir, mayor a miles de nodos, podrá mostrarse todo el grafo para dar una idea de su estructura general pero se hace difícil si no imposible ver el detalle de una parte del mismo. El tamaño del grafo plantea problemas adicionales de navegabilidad e interactividad.

Otro de los factores esenciales es la legibilidad. Los aspectos de la legibilidad pueden ser expresados por medio de criterios estéticos tales como la minimización de cruces entre aristas o la muestra de simetrías. En este marco, algunos problemas de visualización de grafos formalizan estos criterios estéticos como metas de optimización de objetivos múltiples. En general, algunos criterios estéticos no pueden optimizarse simultáneamente. Otra de las tendencias es expresar los criterios estéticos mediante restricciones.

Podemos concluir entonces que los elementos esenciales que deben contemplarse en el dibujado de un grafo son el tamaño, el tipo de conectividad del mismo ([DIB99],[PCJ96],[PUR97]) y su aplicación.

3 Técnicas generales para el dibujado de grafos

Un algoritmo de visualización de grafos recibirá como dato de entrada el propio grafo G que se

quiere visualizar y determinadas restricciones que deben especificarse adecuadamente. Dado que lo percibido por el ser humano varía de una persona a otra, y que aplicaciones diferentes requieren sin duda tipos de representaciones diferentes, el dominio de aplicación surge también como un parámetro esencial en el dibujado de grafos.

Podemos decir entonces que los requerimientos que contribuirán a caracterizar el dibujo a obtener pueden expresarse en términos de los siguientes aspectos:

- a) *Convenciones del dibujo*: Son las reglas básicas que el dibujo debe respetar para ser admisible..
- b) *Estética*: Los criterios estéticos especifican un conjunto de propiedades del gráfico que deberemos aplicar, tanto como sea posible, para mejorar su legibilidad.
- c) *Restricciones*: Mientras que tanto las convenciones como los criterios estéticos se aplican a todo el grafo, las restricciones comúnmente se refieren a subgrafos específicos.

La investigación en el dibujado de grafos se ha enfocado casi exclusivamente en el aspecto algorítmico, en el que el dibujo del grafo es generado de acuerdo a un conjunto de criterios estéticos y convenciones prefijadas.

El enfoque de formalizar los criterios estéticos para la visualización de grafos como metas de optimización es eficiente desde el punto de vista computacional, pero no es muy flexible a la adopción de restricciones impuestas por el usuario ya que las restricciones definidas por éste no son naturalmente soportadas.

El enfoque declarativo formaliza los criterios estéticos especificando la conectividad del grafo mediante un conjunto de restricciones especificadas por el usuario. La conectividad deseada se obtiene entonces como la solución del sistema de restricciones especificadas por el usuario. La ventaja de este enfoque es su poder expresivo. Como desventaja, debemos notar que algunos criterios estéticos, como por ejemplo la planaridad, deben especificarse con restricciones muy complicadas; además, los sistemas generales para resolver restricciones son ineficientes computacionalmente. Por otro lado, se destaca la falta de un lenguaje poderoso en cuanto a su expresividad para la especificación de restricciones. Esto determina que los algoritmos existentes no tengan una gran capacidad para satisfacer restricciones y que la modificación de los mismos para lograrlo sea muy difícil ([TAM88],[DIB92])

En los últimos años se han realizado varios intentos en el desarrollo de lenguajes para la especificación de restricciones y en el diseño de técnicas para dibujar grafos basadas en sistemas de restricciones ([KAM89],[MAR91],[DEN93]). Edades y Lin ([LIN95]) combinan métodos algorítmicos y declarativos en el dibujado de árboles y Brandenburg ([BRA95]) presenta un intento basado en una gramática para grafos.

3.1 Paradigmas en los Algoritmos para Dibujar Grafos

En esta subsección veremos los paradigmas que rigen el dibujado de grafos. Daremos una breve descripción de los resultados disponibles y discutiremos brevemente su relevancia para la visualización de grafos.

Notaremos con C una clase de Grafos y con C' una subclase de C . Un algoritmo que pueda aplicarse para dibujar cualquier grafo perteneciente a C , puede también ser aplicado para hacer lo propio con cualquier grafo perteneciente a C' . Dado que la metodología para dibujar grafos puede verse como un pipeline de pasos funcionales relativamente independientes, y que

- cada paso ejecuta una única tarea que servirá de entrada para el próximo paso y
- la entrada y la salida de cada uno de los pasos del pipeline gráfico son grafos, no necesariamente pertenecientes a la misma clase, pero enriquecidos con características que hacen a su posterior representación,

se puede concluir que es posible sintetizar nuevas metodologías concatenando pasos funcionales pertenecientes a otras metodologías y optando por una estrategia para abordar el problema de dibujar un grafo. A continuación se describen brevemente los métodos más representativos para el problema de dibujar un grafo:

3.1.1 Topología-Forma-Métrica (Topology-Shape-Metrics)

Este método originalmente propuesto en [BAT86] [TAM87] y [TAM88] fue pensado para construir dibujos en grillas ortogonales, y permite el tratamiento homogéneo de un amplio rango de criterios estéticos y restricciones.

La idea básica del método es la caracterización de un dibujo ortogonal mediante tres propiedades fundamentales *Topología, Forma y Métrica*, definidas en término de las clases de equivalencia que los autores establecen entre los dibujos ortogonales de un mismo grafo. Los conceptos dados no sólo pueden utilizarse para caracterizar dibujos ortogonales, sino también dibujos de líneas poligonales en general.

La relación jerárquica existente entre los conceptos de topología, forma y métrica sugieren una generación paso a paso del dibujo, donde en cada paso se produce una representación intermedia. El método consiste en los siguientes pasos: En una primer etapa se planariza el grafo de entrada, luego se obtiene una representación ortogonal del mismo y por último se determinan las coordenadas de cada vértice y de los puntos de inflexión de las aristas. El objetivo normalmente es mantener mínima el área total del dibujo. También en este paso se eliminan los vértices ficticios insertados en el paso 1.

Las ventajas de los métodos de este tipo son su rapidez, su aplicabilidad en caso de tener que dibujar arcos rectos, ortogonales y aún poligonales, soportados por resultados teóricos sobre dibujado plano y extensible aún a grafos con gran cantidad de nodos. Sin embargo como desventajas cabe destacar que su implementación es realmente compleja, que es muy dificultoso extenderlo a 3D y que las transformaciones topológicas pueden alterar el mapa mental que tiene el usuario del grafo. La limitada capacidad de satisfacción de restricciones puede mirarse como ventaja y desventaja al mismo tiempo.

3.1.2 Fuerza-Dirigida

Los algoritmos pertenecientes a esta familia constituyen métodos intuitivos para crear dibujos de líneas rectas de grafos. Estos son bastante populares porque sus versiones básicas son fáciles de entender y codificar. Estos algoritmos simulan un sistema de fuerzas definido sobre el grafo de entrada y produce una configuración de energía localmente mínima.

También en este caso, la limitada capacidad de satisfacción de restricciones puede mirarse como ventaja y desventaja al mismo tiempo. Las restricciones pueden especificarse tanto por el posicionamiento de los vértices del grafo como a través de las fuerzas.

Este método es muy fácil del implementar, se le pueden adicionar heurísticas de manera sencilla, la evolución lenta del dibujo del grafo a su estado final preserva el mapa mental que tiene el usuario del grafo, puede extenderse fácilmente a 3D y en la práctica trabaja bien para grafos pequeños con estructura regular. Como desventajas caben citar su lentitud en ejecución y su dificultad para ser extendido en el dibujado de grafos que representarán sus arcos con líneas ortogonales y poligonales.

3.1.3 Jerárquico

Esta metodología propuesta en [SUG81],[CAR80] y [WAR77] es intuitiva y fue diseñada para representar dígrafos acíclicos con convenciones de dibujos con líneas poligonales y orientadas hacia abajo. Podemos distinguir los siguientes pasos: Se genera un dígrafo en niveles propios (proper layered digraph) en el que los vértices de G tienen asignadas capas o niveles N_1, N_2, \dots, N_b , tal que si $(u,v) \in A$ con $u \in L_i$ y $v \in L_j$, entonces $i = j + 1$. Luego se procede a minimizar el número de cruces entre las aristas de las distintas capas y por último se procede a la asignación de las coordenadas horizontales a los vértices preservando el ordenamiento asignado en el paso previo. El generar entonces el dibujo final y las aristas se dibujan como líneas rectas entre los vértices, pudiendo quedar algunas aristas representadas por una poligonal.

4 Conclusiones y Trabajo Futuro

Dada la diversidad de los dominios de aplicación, uno de los desafíos en el diseño de algoritmos para dibujar grafos es el de brindarles la capacidad para generar dibujos adecuados teniendo en cuenta las especificaciones del usuario.

Otro problema relevante es el de obtener algo más que la simple representación pictórica del grafo. El objetivo en este caso es enriquecer los dibujos de modo tal que logren transmitir algo más que lo expresado por la topología del grafo. Esto nos mueve al campo de la Visualización.

En el campo de la VI, la visualización de grafos juega un rol preponderante en lo referido a representación de información estructurada. Deben enfrentarse diversos problemas al utilizar grafos para visualizar información. Inicialmente se debe determinar si un grafo es una herramienta adecuada para la aplicación; viendo si dentro del espacio de información hay en los datos relaciones inherentes que queremos visualizar. Si es así, podemos representar los datos como nodos y las relaciones entre los mismos como los arcos.

Otra de las características de los grafos que se utilizan para VI es su gran tamaño, el cual puede llegar a ser inmanejable en al menos dos aspectos: legibilidad de la representación obtenida y costo computacional. Los grafos más pequeños que se utilizan comúnmente en VI tienen miles de nodos. Esto nos lleva a contar con técnicas de simplificación del grafo, basándonos en la estructura de la información a mostrar (*clustering*), debiendo tal vez obtener así distintas representaciones para una misma muestra, cada una de ellas más simple que la anterior. Para que las representaciones así obtenidas sean capaces de transmitir la información original hay que proveer técnicas de navegación y de interacción.

El trabajo realizado hasta ahora se cumplió en dos etapas:

- Se estudiaron y analizaron distintas técnicas básicas para el dibujado de grafos.

Se analizaron las características de dichas técnicas en el contexto de la VI que es el área en la que se proyectan aplicar las técnicas del dibujado de grafos.

La meta de nuestro trabajo consiste en encontrar algoritmos eficientes para lograr dibujos de grafos que sean legibles. Para ello se trabajará con los grafos considerados chicos (hasta varios cientos de nodos) en lo que se refiere a una propuesta que considere algoritmos que nos permitan emular automáticamente el modelado manual tradicional ya que la calidad de los trabajos hechos manualmente es una buena opción. También se pretende abordar el dibujado de grafos desde el punto de vista de la Visualización para ser aplicado a la VI. En este caso se trabajará con grafos grandes, y nuestra meta debe ser más amplia y debe extenderse al desarrollo de nuevas propuestas y modelos para la visualización de los mismos.

Bibliografía

- [BAT86] C. Batini, E. Nardelli y R. Tamassia, *A Layout Algorithm for Data Flow Diagrams*, IEEE Trans. Softw. Eng., SE-12, no. 4, 1986.
- [BRA95] F. J. Brandenburg, *Designing Graph Drawings by Layout Graph Grammars*, Graph Drawing (Proc. GD '94), vol. 894 of Lecture Notes Comput. Sci., Springer-Verlag, 1995.
- [CAR80] M. J. Carpano, *Automatic Display of Hierchized Graphs for Computer Aided Decision Analysis*, IEEE Trans. Syst. Man Cybern., SMC-10, no. 11, 1980.
- [DEN93] E. Dengler, M. Friedell, and J. Marks. *Constraint-driven diagram layout*. Proc. IEEE Sympos. On Visual Languages (VL. '93), 1993.
- [DIB92] G. Di Battista, R. Tamassia y I. Tollis, *Constrained visibility representation of Graphs*, Inform. Process. Lett., vol. 41, 1992.
- [DIB99] G. Di Battista, P. Eades, R. Tamassia y I. Tollis, *Graph Drawing, Algorithms for the visualization of graphs*, Prentice Hall, 1999.
- [KAM89] T. Kamada. *Visualizing Abstract Objects and Relations*. World Scientific Series in Computer Science, 1989.
- [KNU63] D. E. Knuth, *Computer Drawn Flowchart*, Commun.. ACM, 6, 1963.
- [LIN95] T. Lin y P. Eades. *Integration of declarative and algorithmic approaches for layout creation*. Graph Drawing (Proc. GD '94), vol. 894, Lectures notes in Computer Science, Springer-Verlag, 1995.
- [MAR91] J. Marks, *A formal specification for network diagrams that facilitates automatic design*. Journal of Visual Languages and Computing, 1991
- [PCJ96] H. Purchase, R. Cohen y M. James, *Validating Graph Drawing Aesthetics*, Graph Drawing (Proc. GD '95), Lecture Notes Comput. Sci. vol. 1027 , Springer-Verlag, 1996.
- [PUR97] H. Purchase, *Wich aesthetics has greatest effects on human understanding*, Graph Drawing '97, vol. 1353 of Lectures Notes in Computer Science, Springer-Verlag, 1997.
- [SUG81] K. Sugiyama, S. Tagawa y M. Toda, *Methods for Visual Understanding of Hierarchical Systems*, IEEE Trans. Syst. Man Cyber., SMC-11, no. 2, 1981
- [TAM87] R. Tamassia, *On Embedding a Graph in the Grid with a Minum number of Bends*, SIAM J. Comput., 16, no. 3, 1987.
- [TAM88] R. Tamassia, G. Di Battista y C. Batín. *Automatic Graph Drawing and Readability of diagrams*, IEEE Trans. Syst. Man Cybern., SMC-18(1), 1988.
- [WAR77] j. Warfield, *Crossing Theory and Hierarchy Mapping*, IEEE Trans. Sys. Man. Cyber., SMC-7, no. 7, 1977