Usando GeoGebra en teoría de grafos

Falcón Ganfornina, Raúl Manuel[1] rafalgan@us.es
Ríos Collantes de Terán, Ricardo[2] profesofricardo@yahoo.es

Resumen

En las últimas versiones de *GeoGebra* se observa la aparición de un mayor número de herramientas avanzadas relacionadas con el campo de la Matemática Discreta, donde dicho programa se postula como una futura promesa tanto en la investigación como en la docencia de dicha materia. Sin embargo, aún queda mucho camino por recorrer en este sentido. En el presente trabajo se indican algunas posibles aplicaciones de *GeoGebra* en la teoría de grafos. En concreto, se muestra paso a paso la manera de construir grafos aleatorios con un determinado número de vértices, analizando de forma dinámica algunas de sus propiedades básicas. El uso en el aula de Matemáticas de la plantilla generada posibilita una mejora tanto en la enseñanza como en el aprendizaje de los conceptos relacionados con la teoría de grafos.

1. Introducción

GeoGebra dispone de varias herramientas asociadas a diversos campos de estudio en Matemática Discreta: árbol recubridor, camino mínimo, problema del viajante, cierre convexo, triangulación de Delaunay y diagramas de Voronoi. Si bien los tres primeros conceptos se engloban dentro de la teoría de grafos, aún queda mucho potencial que puede ser explotado en *GeoGebra* a la hora de tratar problemas relativos a dicha teoría, más aún cuando el desarrollo de herramientas informáticas y algoritmos que faciliten el uso y aplicación de grafos en ciencia e ingeniería es un hecho vigente, como así consta en las celebraciones anuales de congresos explícitamente dedicados al diseño de grafos (el simposio internacional *Graph Drawing* celebra por ejemplo este año su vigésimo-primera edición). Existe una gran variedad de software dedicado al estudio de grafos (Jünger y Mutzel, 2004), alguno de ellos desarrollado en España, como es el caso del programa *Grafos* (Rodríguez Villalobos, 2006). Si bien existe una gran variedad en cuanto a los algoritmos y funcionalidades implementadas en cada una de estas herramientas

[1] ETS de Ingeniería de Edificación. Departamento de Matemática Aplicada I. Universidad de Sevilla.
[2] IES Sofía – Jerez de la Frontera.

informáticas, pensamos que *GeoGebra* puede llegar a aportar novedades en diversos aspectos como son la visualización, la animación y la interacción dinámica y paramétrica, junto con un futuro posible desarrollo en el estudio de grafos en tres dimensiones con la nueva versión 5.0 de *GeoGebra*. El abanico de posibilidades es pues amplio y prometedor. El pequeño granito de arena que aportamos en el presente trabajo se basa en la experiencia de desarrollar una plantilla que genera aleatoriamente un grafo con un número de vértices a determinar por el usuario, junto con el análisis de algunas de sus propiedades básicas. El uso de dicha plantilla en el aula de Matemáticas a la hora de comenzar a tratar conceptos de la teoría de grafos posibilita un mejor entendimiento por parte del alumnado, quien puede interactuar de una forma dinámica sobre cada grafo generado, bien en la misma plantilla generada, bien en una hoja de trabajo aparte donde pueda ejercitarse por ejemplo en conceptos como la planaridad del grafo, el coloreado de aristas y vértices o la generación de las matrices de incidencia.

2. Generación de grafos aleatorios

Nuestro primer objetivo es hacer que *GeoGebra* genere de forma aleatoria un grafo de un determinado número de vértices. Para ello creamos por ejemplo un deslizador ***n*** con valores en el intervalo ***[1,30]*** e incremento ***1***. Una manera de generar los *n* vértices sería utilizar la hoja de cálculo y utilizar la primera columna ***A*** para fijar de antemano el conjunto posible de puntos que utilizaríamos como vértices. Esto permitiría una posterior libertad a la hora de desplazar dichos puntos por la pantalla y estudiar por ejemplo la planaridad del grafo correspondiente. En nuestro caso sin embargo vamos a optar por fijar las *n* raíces de la unidad distribuidas de forma homogénea en la circunferencia de centro el origen de coordenadas y radio unidad. Para ello definimos la secuencia V=Secuencia[cos(2k π / n) + í sen(2k π / n), k, 0, n-1]. A continuación procedemos a generar de forma aleatoria las aristas que compondrán nuestro grafo. En concreto las determinaremos definiendo previamente una matriz de adyacencia, cuyas entradas binarias de ceros y unos serán generadas de forma aleatoria siguiendo una distribución binomial de probabilidad **p** prefijada de antemano en una casilla de entrada incluida en nuestra plantilla, de tal forma que, en particular, si **p**=1, entonces obtendremos un grafo completo (ver Figura 1). Habrá que distinguir además si el grafo es dirigido o no dirigido, lo cual podemos determinar haciendo uso de una casilla de control ***dir***, que marcaremos cuando queramos optar por un grafo dirigido y desmarcaremos en caso de tener interés en uno no dirigido. La matriz de adyacencia y las aristas del grafo dirigido aleatorio se definen respectivamente como:

M=Secuencia[Encadena[Encadena[Secuencia[BinomialAleatoria[1,p],j,1,i-1],{0}],
Secuencia[BinomialAleatoria[1,p],j,i+1,n]],i,1,n]

A = Secuencia[Secuencia[Si[Elemento[M,i,j]≠0,Vector[Elemento[V,i], Elemento[V,j]]],j,1,n],i,1,n]

Hay que indicar como condición para exponer la lista de aristas dirigidas **A** la condición booleana **dir==true**. Por su parte, en el caso del grafo no dirigido debemos tener en cuenta que su matriz de adyacencia **M'** es simétrica, por lo que debemos proceder de una manera diferente. En concreto, definimos una lista aleatoria con las entradas que están por encima de la diagonal de **M'**: L = Secuencia[Secuencia[BinomialAleatoria[1,p],j,i+1,n],i,1,n-1]. La matriz de adyacencia y las aristas del grafo no dirigido aleatorio se definen entonces respectivamente como:

M' = Secuencia[Encadena[Encadena[Secuencia[Elemento[Elemento[L,j],i-j],j,1,i-1],{0}], Secuencia[Elemento[Elemento[L,i],j],j,1,n-i]],i,1,n]

A' = Secuencia[Secuencia[Si[Elemento[M',i,j]≠0,Segmento[Elemento[V, i], Elemento[V, j]]], j,i+1,n],i,1,n]

Hay que indicar como condición para exponer la lista de aristas dirigidas **A'** la condición booleana **dir==false**.

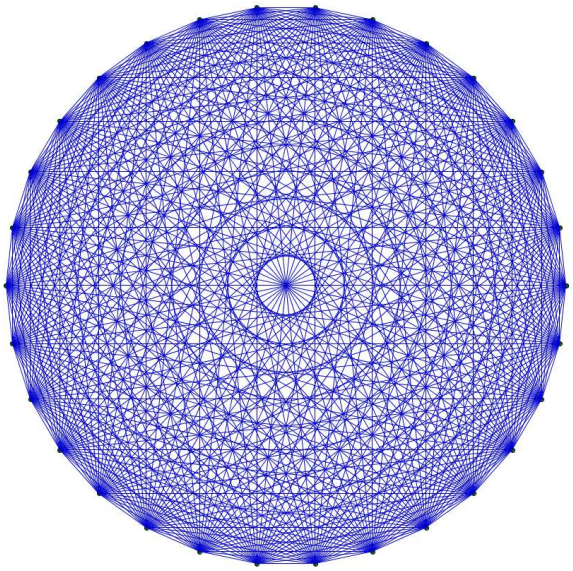

Figura 1: Grafo completo de 30 vértices en *GeoGebra*.

3. Propiedades básicas de grafos

Nos centramos a continuación en la definición en *GeoGebra* de algunos de los invariantes de los grafos que hemos generado aleatoriamente en la sección previa. Los siguientes invariantes pueden ser fácilmente implementados en *GeoGebra*:

- Orden = Orden' = n
- Tamaño = Suma[Secuencia[Suma[Secuencia[Elemento[M,i,j],j,1,n]],i,1,n]]
- Tamaño' = Suma[Secuencia[Suma[Elemento[L,i]],i,1,n-1]]
- detM = Determinante[M]
- detM' = Determinante[M']

Ciertos invariantes en teoría de grafos se basan en las distancias entre los vértices. Para lograr obtener en *GeoGebra* la matriz de distancias de un determinado grafo será necesario hacer uso del lenguaje *JavaScript*. Como ejemplo indicaremos el código basado en el algoritmo de Dijkstra asociado a los grafos no dirigidos, siendo análogo el referente a los dirigidos. Previamente necesitamos pasar la matriz de adyacencia del grafo a la secuencia simple m' = Aplana[M']. Seguidamente creamos la siguiente función en la correspondiente solapa de *JavaScript Global*:

```
function distancia(){
var n = ggbApplet.getValue("n"); var D = new Array(); var NM = new Array();
var P = new Array(); var i, d, m, v, a, a0; var s=0;
for(i =0; i < n*n; i++){D[i] = 100;}
for(a0 =1; a0 < n+1; a0++){
  for(i =0;i<n;i++){NM[i]=0; P[i]=-1;}
  a=a0;  D[(a0-1)*n+a-1]=0;  NM[a-1]=1;
  while(a!=0){
    for(i =0;i<n;i++){
      m=ggbApplet.getListValue("m'",(a-1)*n +i+1);
      if (NM[i]==0 & m!=0 & D[(a0 - 1)*n + i] > D[(a0 - 1)*n + a - 1] + m){
        D[(a0 - 1)*n + i]=D[(a0 - 1)*n + a - 1] + m; P[i]=a;}}
     d=100;
     for(i=0;i<n;i++){
       if (NM[i]==0){
         if (s==0){s=i; d=D[(a0-1)*n+i]; }
         else{if (D[(a0-1)*n+i]<d){d=D[(a0-1)*n+i]; s=i;}}}}
     if (d==100){v=0;} else{v=s;}
     if (v==0){a=0;} else{ NM[v]=1; a=v+1;}}}
     return(D);}
```

A continuación, basta indicar por ejemplo el código ggbApplet.evalCommand("d'={"+distancia()+"}"); en la solapa **Al actualizar** de la lista **m'**. La matriz de distancias se puede definir entonces como D' = Secuencia[Secuencia[Si[Elemento[d', (i-1) n + j] == 100, ∞, Elemento[d',(i-1) n+j]], j, 1, n], i, 1, n]. Por su parte, el diámetro del grafo no dirigido se define como Diámetro'=Máximo[d']. Por último, el número de componentes conexas puede calcularse incluyendo la siguiente función en *JavaScript Global*:

```
function conexo(){
  var n = ggbApplet.getValue("n");  var i,j,d;  var c=0;  var NM = new Array();
  for(i =0;i<n;i++){NM[i]=0;}
  for(i =0;i<n;i++){
    if (NM[i]==0){
      c=c+1;
      for(j =i;j<n;j++){
        if (NM[j]==0){
          d=ggbApplet.getListValue("d'",i*n +j+1);
          if (d<100){NM[j]=1;}}}}}
 return(c);}
```

Basta entonces añadir el código ggbApplet.evalCommand("c'="+conexo()); en la solapa **Al actualizar** de la lista **m'** debajo de la línea que calculaba la matriz distancia.

4. Conclusiones

En el presente trabajo se ha mostrado cómo puede comenzarse a trabajar en teoría de grafos con *GeoGebra*, desarrollando una plantilla que genera grafos aleatorios dirigidos y no dirigidos con un número determinado de vértices y mostrando algunos de sus invariantes. Esta plantilla dinámica es de gran utilidad en el aula de Matemáticas a la hora de abordar las primeras nociones de grafos. En cualquier caso, el potencial que ofrece *GeoGebra* en este campo es mucho mayor y pueden plantearse una amplia gama de posibilidades como por ejemplo la generación de grafos aleatorios con distintos tipos de invariantes prefijados (no sólo el orden de los mismos), el estudio de grafos bipartitos

Referencias

1. Alejandro Rodríguez Villalobos (2006). *Grafos: herramienta informática para el aprendizaje y resolución de problemas reales de teoría de grafos.* X Congreso de Ingeniería de Organización. Universitat Politècnica de Valencia

2. Michael Jünger, Petra Mutzel, Petra (eds.) (2004). Graph Drawing Software. Springer-Verlag. New York, Inc. Secaucus, NJ, USA.