

Selección aleatoria de árboles generadores en gráficas

Sergio Luis Pérez-Pérez¹, Guillermo Benito Morales-Luna¹
y Feliú Davino Sagols-Troncoso²

¹Departamento de Computación, Centro de Investigación y de Estudios Avanzados,
IPN, Distrito Federal,
México

²Departamento de Matemáticas, Centro de Investigación y de Estudios Avanzados,
IPN, Distrito Federal,
México

sperez@computacion.cs.cinvestav.mx, gmorales@cs.cinvestav.mx,
fsagols@math.cinvestav.edu.mx

Resumen. Existen diversos procedimientos para seleccionar aleatoriamente árboles generadores en gráficas conexas no dirigidas, con tiempos esperados de ejecución entre los órdenes $O(n \log n)$ y $O(n^3)$ en los peores casos, donde n es el número de vértices en la gráfica. En este trabajo realizamos la localización efectiva y eficiente de árboles generadores mediante paseos aleatorios sobre dichas gráficas, con la finalidad de obtener un equilibrio entre el diámetro del árbol, la valencia de los vértices internos y el número de hojas de los árboles obtenidos. Para esto, proponemos el uso de diversas matrices de transición en cadenas de Markov, considerando diferentes distribuciones de probabilidad para las vecindades de vértices involucradas en el paseo aleatorio.

Palabras clave. Selección aleatoria de árboles generadores, paseos aleatorios sobre gráficas, matrices de transición en cadenas de Markov, distribuciones de probabilidad en vecindades de vértices.

Random Selection of Spanning Trees on Graphs

Abstract. Random selection of spanning trees on graphs has been treated extensively in technical literature. Popular randomized algorithms have time complexity varying from $O(n \log n)$ to $O(n^3)$, where n is the order of a graph, namely, the number of vertices. In this work, we introduce effective and efficient procedures to select spanning trees using random walks with the purpose to balance the diameter of the selected tree, the valencies of its inner vertices, and the number of leaves at its yield. We describe several ways to form transition matrices of Markov chains in terms of

probability distributions on the neighborhood of any visited vertex along the random walk.

Keywords. Random spanning trees, random walks on graphs, transition matrices in Markov chains, probability distributions on neighborhoods of vertices.

1 Introducción

Existen diversos procedimientos para la *selección aleatoria de árboles generadores* que denotaremos como *Saag* basados en la construcción de paseos aleatorios. Por ejemplo, uno de los más antiguos y conocidos, fue presentado por Broder [2] y Aldous [1] de forma independiente, el cual se basa en la construcción de un paseo aleatorio que recorre una gráfica dada en un tiempo esperado de ejecución $O(n \log n)$ y $O(n^3)$ en el peor caso, donde n es el número de vértices en la gráfica. Por otro lado, D. B. Wilson [5] propone un algoritmo aleatorio de tiempo lineal respecto al número de evaluaciones de una función aleatoria. Tal algoritmo elige aleatoriamente un vértice de la gráfica como vértice inicial para la construcción del árbol y empieza a generar ramas aleatorias agregándolas en cualquier vértice del árbol que va formando en cada paso. Otro ejemplo, es el algoritmo de J. A. Kelner y A. Madry [3], que selecciona árboles generadores “casi uniformemente”, mejorando la cota de $O(\min\{mn, n^{2.376}\})$ para el peor caso, donde m es el número de aristas en la gráfica.

En este trabajo presentamos los resultados de las pruebas realizadas al algoritmo aleatorio propuesto por Broder [2], pero utilizando matrices de transición en cadenas de Markov con distribuciones de probabilidad no uniformes en las vecindades de vértices, con la finalidad de mostrar algunas ideas con las que se pueden obtener aleatoriamente árboles generadores con determinadas propiedades de forma eficiente. Un ejemplo de aplicación de éstos es el desarrollo de protocolos de identificación basados en teoría de gráficas, como el presentado en [4], donde el diámetro del árbol y su número esperado de hojas, son factores determinantes para establecer la seguridad de dichos protocolos.

De forma general, un *paseo aleatorio* sobre una gráfica conexa no dirigida es aquel que parte de un vértice arbitrario y recorre toda la gráfica generando una *cadena de Markov* descrita por una *matriz de transición*. En cada paso del paseo se elige al vértice siguiente en una vecindad del vértice actual.

Dado que el procedimiento propuesto por Broder [2] es utilizado en todas las pruebas realizadas, mencionamos aquí su construcción sólo a modo de recordatorio.

Sea $G = (V, A)$ una gráfica conexa no dirigida, con conjuntos de vértices V y de aristas A . A partir de este momento n denota la cardinalidad de V y m la de A . Una construcción de un árbol generador aleatorio está dada por el siguiente procedimiento:

Generar

1 Se selecciona un vértice arbitrario $v_0 \in V$ y se construye un paseo aleatorio v_0, v_1, \dots, v_k que contenga todos los vértices de V para algún entero $k \geq n - 1$. El paseo aleatorio termina cuando el último vértice no visitado de la gráfica haya sido visitado.

2 El árbol resultante T tiene como conjunto de vértices a V y como aristas a las parejas $v_i v_j$ tales que para $i, j \in \{0, \dots, k\}$, $i < j$ y j es el menor índice tal que $v_i v_j \in A$.

El procedimiento requiere de una matriz de transición P de orden $n \times n$, la entrada i, j de P es la probabilidad de visitar el vértice v_j desde v_i , así $P_{i,j} > 0$ siempre que $v_i v_j \in A$.

Nuestro objetivo principal es presentar las características generales de los árboles obtenidos como resultado de una experimentación sobre algunos tipos de gráficas conexas no dirigidas generadas aleatoriamente y otras obtenidas de la literatura [6].

Presentamos las estadísticas obtenidas de aplicar diferentes matrices de transición en Generar, así como algunas conclusiones derivadas de los resultados obtenidos en tales estadísticas.

Las instancias de pruebas utilizadas, así como los programas elaborados se encuentran disponibles en <http://computacion.cs.cinvestav.mx/~sperez> a fin de que el lector conozca las pruebas realizadas a detalle. El código que ahí se encuentra es presentado bajo una licencia de uso GPL v3.0.

A continuación presentamos algunas definiciones y convenciones utilizadas para este trabajo.

En lo sucesivo presentamos *Saag* con diversas matrices de transición, donde en cada caso analizamos las longitudes de los paseos aleatorios, el diámetro y el número de hojas de cada árbol generado, así como las valencias promedio y máxima de cada uno de los vértices internos de cada árbol.

Todas las gráficas utilizadas en este trabajo son gráficas conexas no dirigidas por lo que de aquí en adelante sólo utilizaremos el término de *gráfica* para referirnos a gráficas de este tipo.

Cuando nos referimos al *equilibrio de un árbol* esperamos que éste presente una valencia uniforme v en sus vértices internos de modo que a mayor valor de v tengamos un mayor número de hojas y un menor diámetro.

Una *gráfica completa* sobre n vértices, denotada como K_n , es una gráfica simple con n vértices en la que hay una arista entre cada par de vértices distintos.

Denotaremos como $\delta(v_i)$ a la vecindad de un vértice $v_i \in V$, es decir el conjunto de vértices conectados directamente con v_i mediante aristas. También denotaremos por $[s, t]$ al intervalo con los enteros positivos entre s y t incluyéndolos. En cuanto a la notación utilizada en las figuras presentadas, $G(n, m)$ representa a $G = (V, A)$, una gráfica con $n = |V|$ vértices y $m = |A|$ aristas. En todas ellas presentamos los resultados

ordenados para 100 ejecuciones de *Generar*, describiendo el comportamiento de cada una de las características antes mencionadas y utilizando una matriz de transición asociada a G .

A continuación damos un bosquejo general de la estructura de este trabajo. En la Sección 2, presentamos los resultados obtenidos de pruebas realizadas sobre algunas instancias de gráficas asociándoles matrices de transición con distribuciones de probabilidad uniformes. En la Sección 3, presentamos tres propuestas para generar matrices de transición con distribuciones de probabilidad no uniformes. En la Sección 4, se describe un procedimiento para modificar las matrices de transición mediante la ponderación de los vértices de la gráfica. Finalmente, en la Sección 5 damos las conclusiones del trabajo.

2 Saag usando matrices de transición con distribuciones de probabilidad uniformes

Sea $G = (V, A)$ una gráfica y P_G una matriz de transición asociada a G , cada entrada P_{ij} denota la probabilidad de que el siguiente vértice visitado desde v_i sea v_j , con $v_i v_j \in A$. Decimos que una matriz de transición es *uniforme* si:

$$P_{ij} = \begin{cases} \frac{1}{\delta(v_i)}, & \text{si } v_i v_j \in A \\ 0, & \text{si } v_i v_j \notin A \end{cases} \quad (1)$$

Así, de manera general, se tiene que:

$$\sum_{j=1}^n P_{ij} = 1, \text{ para cada } 1 \leq i \leq n \quad (2)$$

Este tipo de matriz de transición es el que se utiliza en *Generar*.

2.1 Longitud de paseo

En las pruebas que realizamos a diferentes instancias de gráficas con números de vértices entre $n = 50$ y $n = 1000$, la longitud del paseo fue del orden $O(n \log n)$ en la mayoría de los casos y $O(n^2)$ en el peor caso, lo cual sucedió sólo en

gráficas con una baja cantidad de aristas, digamos $|A| = O(n)$.

En la Figura 1 se puede observar que entre mayor es la proporción de aristas por vértice en una gráfica, el tiempo de ejecución de *Generar* tiende a ser $O(n \log n)$ y cuando la proporción de aristas por vértice es menor, el tiempo de ejecución va aumentando aunque no llega a ser $O(n^3)$.

Se construyeron dos gráficas con $n = 1000$ vértices, una de ellas es un anillo y la otra un camino, ambas ejemplifican algunos de los peores casos que encontramos donde se obtiene una longitud de paseo de $O(n^2)$, ver Figura 2. Dicho comportamiento es de esperarse pues el paseo es "unidimensional" por tanto en cada vértice, digamos a distancia k del punto de partida, el paseo podría rehacer lo andado y regresar a ese vértice para continuar el descubrimiento de un nuevo vértice. La longitud esperada, en este caso, es pues $O(\sum_{k=1}^n k) = O(n^2)$.

2.2 Diámetro del árbol

En la Figura 3, presentamos resultados que muestran que el diámetro no varía mucho en los árboles seleccionados aleatoriamente desde gráficas con un mismo número de vértices y una proporción diferente de aristas. Esto indica que el diámetro de dichos árboles depende del número de vértices de la gráfica de donde son extraídos y poco tiene que ver con la cantidad de aristas de dicha gráfica, sin embargo la variación de tales diámetros debería ser mayor en gráficas completas en comparación a gráficas con pocas aristas.

En la Figura 4, se puede observar el comparativo de seis gráficas, tres de ellas son K_{250} , K_{500} y K_{1000} y las otras tres son tomadas de [6], en particular son las instancias `dsjc250.1.col`, `dsjc500.1.col` y `dsjc1000.1.col`, con 250, 500 y 1000 vértices respectivamente. Entre cada par de gráficas con igual número de vértices, la proporción de aristas, es aproximadamente 10:1 y pese a ello se observa un comportamiento similar en términos del diámetro presentado en los árboles resultantes.

2.3 Número de hojas en el árbol

Una *gráfica regular* es una gráfica donde todo vértice posee una misma valencia. El trabajo de Aldous [1] se centra en la Saag en gráficas regulares. Ahí se demuestra lo siguiente: sea T un árbol generador aleatorio escogido de manera uniforme en una gráfica G r -regular, con $r \geq 3$. Entonces:

- (a) $P(v \text{ sea una hoja de } T) \leq \exp\left(-\frac{r-1}{2r}\right), \forall v \in G.$
- (b) $Prom_{v \in G} P(v \text{ sea una hoja de } T) \geq \alpha(r),$

donde:

$$\alpha(r) = \sum_{j=2}^{r-2} r^{-1} \left(1 - \frac{j}{r}\right) \left(1 - \prod_{i=1}^{j-1} \left(1 - r^{-1} \left(1 - \frac{i}{r}\right)\right)\right) \quad (3)$$

En [1] se muestra que $\alpha(r) = 0.106$ cuando $r \rightarrow \infty$. De este modo, si $H(T)$ es el número de hojas del árbol T extraído de G , entonces $E[H(T)] = (r - 1)/2r$.

En el caso de nuestras pruebas realizadas las únicas gráficas r -regulares que utilizamos fueron gráficas completas, las cuales presentaron el comportamiento antes mencionado.

En los resultados presentados en la Figura 5 mostramos que para gráficas que no son r -regulares, los árboles resultantes tienen un número de hojas similar al de los extraídos de gráficas r -regulares. También se observa que conforme el número de aristas en una gráfica crece, se decrementa el número de hojas del árbol seleccionado aleatoriamente, es decir el árbol resulta más equilibrado.

2.4 Valencias promedio y máxima de los vértices internos del árbol

En cualquier árbol con $n \geq 1$ vértices, la valencia promedio es igual a $2 - 2/n$ [1], sin embargo esto no dice mucho del tipo de árbol obtenido, ya que en dicho promedio están consideradas las hojas del árbol, por tanto nos centramos en sólo calcular el promedio de los

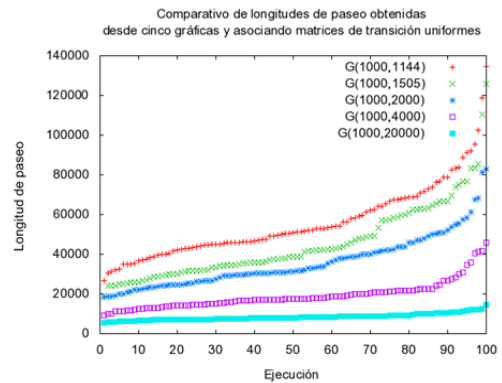


Fig. 1. Longitudes de paseo I

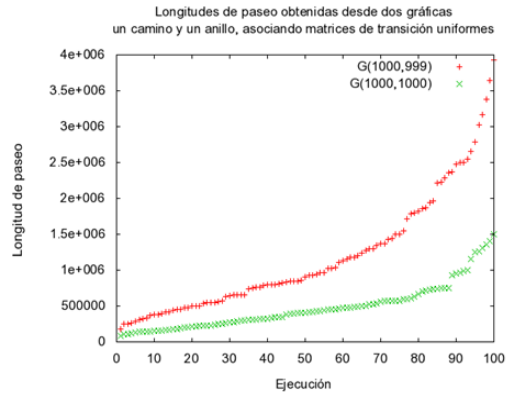


Fig. 2. Longitudes de paseo II

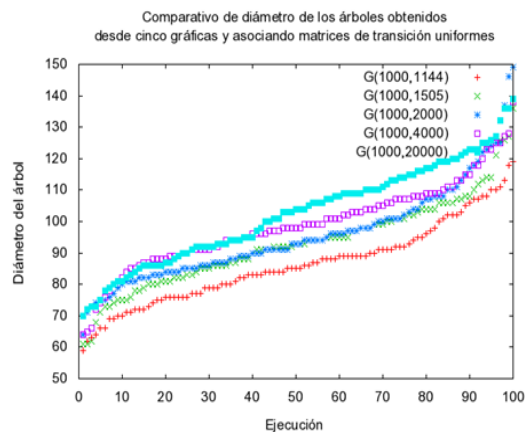


Fig. 3. Diámetro de los árboles I

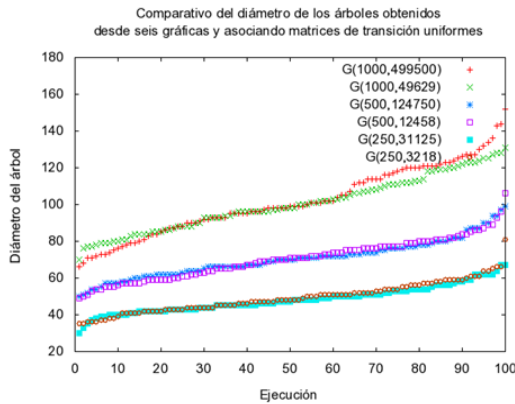


Fig. 4. Diámetro de los árboles II

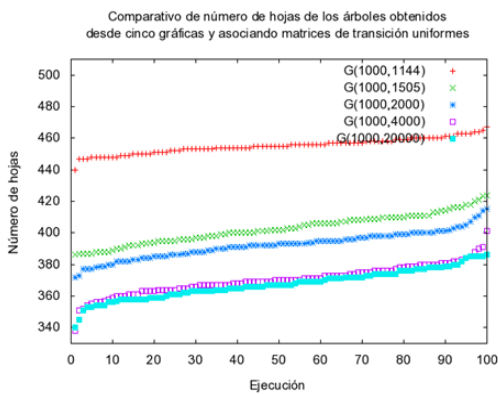


Fig. 5. Número de hojas de los árboles I

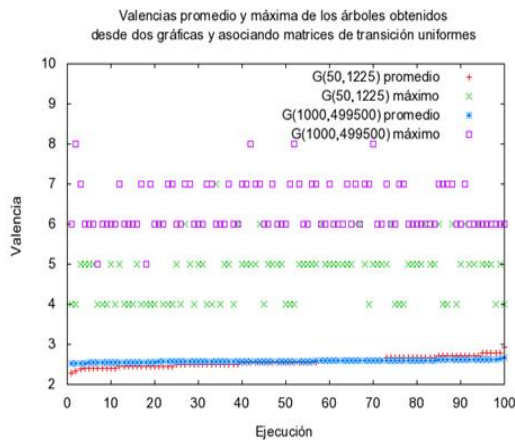


Fig. 6. Valencias promedio y máxima de los vértices internos de los árboles I

vértices internos del árbol, lo cual muestra una idea de la uniformidad en las valencias del árbol generado.

De todas las pruebas realizadas en K_n con $n \in \{50,100,250,500,1000\}$, se pudo observar que la valencia promedio de los vértices internos de cada árbol es poco variable, de hecho la tendencia fue de entre dos o tres vecinos por vértice, lo cual nos hace concluir que en la mayoría de los casos el árbol obtenido en cada ejecución de Generar presenta valencias bajas, lo cual resulta del hecho de que la valencia máxima encontrada en todas las pruebas realizadas sobre tales gráficas no fue mayor a 10, así pues, el comportamiento estadístico fue siempre el mismo que el presentado en la Figura 6.

3 Saag usando matrices de transición con distribuciones de probabilidad no uniformes

Sea $G = (V, A)$ una gráfica y P_G una matriz de transición asociada a G , decimos que una matriz de transición tiene una distribución de probabilidad para las vecindades de vértices *no uniforme*, si cumple:

Condiciones-No-Uniforme

1. $0 \leq P_{ij} \leq 1$, para cada $v_i \in V$ y $v_i v_j \in A$.
2. Para cada $v_i \in V$, $\sum_{v_i v_j \in A} P_{ij} = 1$.
3. Existe al menos un índice i_0 y dos índices distintos j_0, j_1 tales que $v_{i_0} v_{j_0}, v_{i_0} v_{j_1} \in A$ y $P_{i_0 j_0} \neq P_{i_0 j_1}$.

A continuación presentamos la construcción de tres tipos de matrices de transición con distribuciones de probabilidad no uniformes y algunos resultados derivados de utilizarlas en Generar.

3.1 Matriz de transición afín decreciente

La construcción de una *matriz de transición afín decreciente* a la cual denotaremos como P_{D_G} , asociada a una gráfica G , se realiza mediante el siguiente procedimiento:

Construir-Afín

1. Sean $v_i \in V$ y $v_{j_1}, \dots, v_{j_{|\delta(v_i)|}} \in A$ los vértices en la vecindad de v_i ordenados arbitrariamente.
2. Sea $g_i = (|\delta(v_i)| \cdot (|\delta(v_i)| + 1))/2$.
3. Asignar $P_{ij_k} = \frac{k}{g_i}$, para cada $v_i \in V$ y $k = 1, \dots, |\delta(v_i)|$.

3.2 Matriz de transición aleatoria

A una matriz de este tipo la nombramos *aleatoria*, porque asocia una matriz P_{AG} de probabilidad no uniforme a una gráfica G . Su construcción se presenta en el Algoritmo 1 (más abajo). Si la primera vez que se ejecuta la línea 9 del Algoritmo 1, se resta un valor grande a la variable `suma`, entonces las demás probabilidades serán pequeñas. Sin embargo, como se observa en la línea 10, al barajar la lista L , las probabilidades de cada vecindad de vértices quedan distribuidas aleatoriamente, evitando que probabilidades grandes sean cargadas hacia los mismos vecinos de cada vértice.

3.3 Matriz de transición aleatorizada sobre un intervalo

A esta matriz de transición la nombramos *aleatorizada sobre un intervalo* porque la matriz de transición P_{IG} asociada a una gráfica G generalmente evita la asignación de valores de probabilidad demasiado pequeños, a diferencia de la matriz anterior. Sin embargo, ello depende del intervalo que sea especificado en la construcción, descrita en el Algoritmo 2.

El Algoritmo 2 toma como entrada una gráfica $G = (V, A)$. Para cada $v_i \in V$ sea r_i un entero mayor o igual que $|\delta(v_i)|$. Este procedimiento utiliza una función recursiva llamada *Divide* que toma como parámetros de entrada un intervalo $I = [s, t]$ y un conjunto $C \subseteq \delta(v_i)$. En la primera invocación a *Divide* para el vértice v_i , tenemos que $s = 1$, $t = r_i$ y $C = \delta(v_i)$ y en cada invocación recursiva se garantiza que $|I| \geq |C|$. La función *Divide* calcula $l_1 = \lceil (|C|/2) \rceil$ y $l_2 = \lfloor (|C|/2) \rfloor$; determina una partición aleatoria C_1, C_2 de C con

$|C_1| = l_1$ y $|C_2| = l_2$; selecciona aleatoriamente un valor entero k , con $s + l_1 - 1 \leq k \leq t - l_2$; construye los subintervalos $[s, k]$ y $[k + 1, t]$ y finalmente invoca recursivamente a *Divide* ($[s, k], C_1$) y *Divide* ($[k + 1, t], C_2$). El caso base de la recursión ocurre cuando $C = \{v_j\}$ para algún $v_j \in \delta(v_i)$ y entonces se asigna la probabilidad $P_{ij} = |I|/r_i$.

Para las pruebas realizadas se consideraron intervalos $r_i = |\delta(v_i)|/|V|$ para cada $v_i \in V$. Una vez construidas las matrices de transición con distribuciones de probabilidad no uniformes se utilizaron en *Generar*.

Los resultados obtenidos se muestran en las Figuras 7 a 12. Se realizaron ejecuciones con K_{1000} y una gráfica con 1000 vértices y 4000 aristas generada aleatoriamente. Para estas dos gráficas se construyeron cuatro matrices de transición: uniforme, afín decreciente, aleatoria y aleatorizada sobre un intervalo denotadas en las figuras como P_U, P_D, P_A y P_I respectivamente. Aunque sólo presentamos una muestra del total de las pruebas realizadas, los resultados para otras gráficas tuvieron un comportamiento estadístico similar.

3.4 Longitud de paseo

En la Figura 7 se puede observar que las mayores longitudes de paseo se obtuvieron con la distribución aleatoria, en los otros tres casos las longitudes de paseo fueron prácticamente idénticas. Cuando el número de aristas es pequeño, las longitudes de paseo generadas con la matriz de transición P_{AG} tienden a ser muy largas, alcanzando longitudes de paseo de orden $O(n^3)$. Ver Figuras 7 y 8.

3.5 Diámetro del árbol

El diámetro de los árboles obtenidos no presenta gran variación respecto al tipo de matriz de transición utilizada. Esto se puede observar comparando las Figuras 3, 4 y 9, donde para gráficas con $n = 1000$ vértices y un número arbitrario de aristas los árboles seleccionados siempre presentan un diámetro aproximado entre 60 y 160.

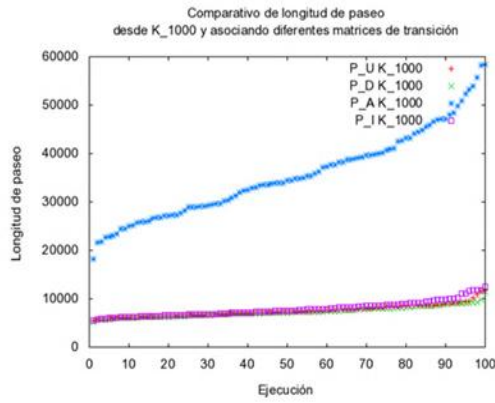


Fig. 7. Longitudes de paseo III

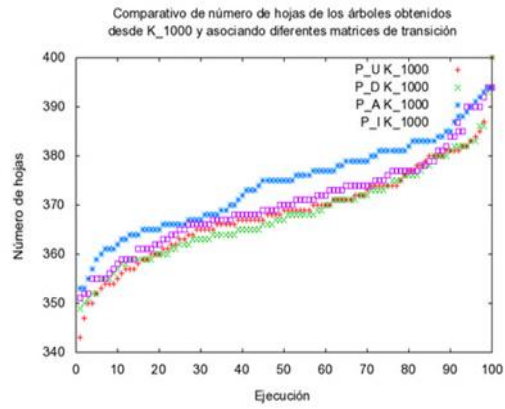


Fig. 10. Número de hojas de los árboles II

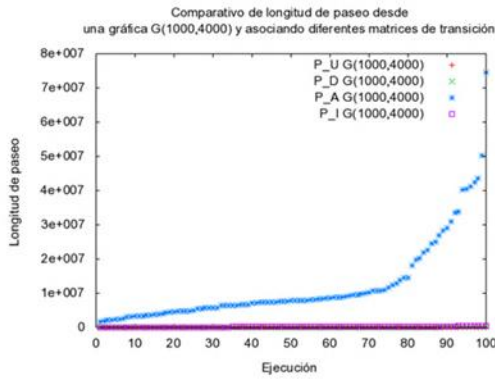


Fig. 8. Longitudes de paseo IV

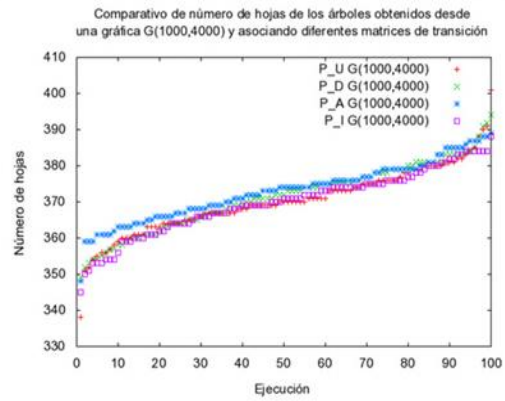


Fig. 11. Número de hojas de los árboles III

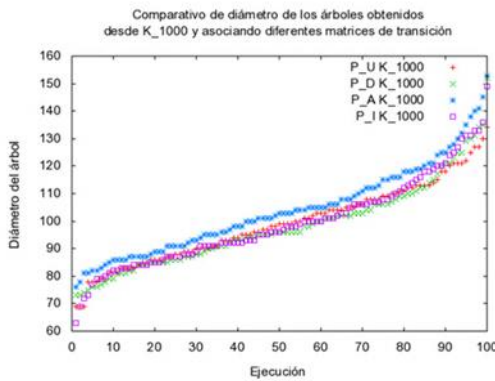


Fig. 9. Diámetro de los árboles III

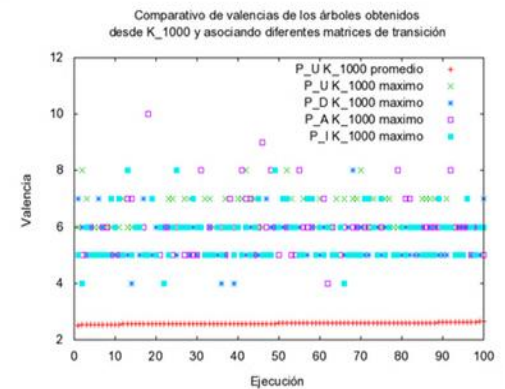


Fig. 12. Valencias promedio y máxima de los vértices internos de los árboles II

Algoritmo 1: Generación de una matriz de transición aleatoria.**Entrada:** Una gráfica conexa no dirigida $G = (V, A)$.**Salida:** Una matriz de transición P_{A_G} asociada a G , con distribución aleatoria.

```

1 Crear una matriz  $P_{n \times n} \leftarrow 0$ ;
2 Para cada  $v_i \in V$  con  $1 \leq i \leq n$  hacer
3   Crear una lista  $L \leftarrow \{\}$ ;
4   Hacer  $u \leftarrow \delta(v_i)$ ,  $suma \leftarrow 1.0$ ;
5   Para  $j \leftarrow 1$  hasta  $|u| - 1$  hacer
6     Repite
7        $s \leftarrow Rnd(eps, 1.0)$ ; //  $eps \leftarrow 10^{-5}$ 
8       hasta  $(suma - s) > 0$ ;
9        $L \leftarrow L || s$ ,  $suma \leftarrow suma - s$ ;
10     $L \leftarrow L || suma$ ;
11    //  $L || x$  denota agregar  $x$  a  $L$ 
12     $L \leftarrow Permutar(L)$ ;
13    // Escoger una permutación aleatoria de  $L$ 
14    Para  $j \leftarrow 1$  hasta  $|u|$  hacer
15       $P_{i,u_j} \leftarrow Frente(L)$ ;
16      //  $u_j$  es elemento  $j$ -ésimo de  $u$ 
17       $L \leftarrow Eliminar(Frente(L))$ ;
18 regresar  $P$ ;

```

Algoritmo 2: Generación de una matriz de transición aleatorizada sobre un intervalo.**Entrada:** Una gráfica conexa no dirigida $G = (V, A)$.**Salida:** Una matriz de transición P_{I_G} asociada a G , con distribución aleatorizada sobre un intervalo.

```

1 Crear una matriz  $P_{n \times n} \leftarrow 0$ ;
2 Para cada  $v_i \in V$  con  $1 \leq i \leq n$  hacer
3    $r_i \leftarrow |\delta(v_i)| |V|$ ;
4   Divide( $[1, r_i]$ ,  $\delta(v_i)$ );
5 regresar  $P$ ;
6 // Función divide
7 Función Divide( $I \leftarrow [s, t]$ ,  $C$ )
8 Si  $|C| = 1$  entonces
9   Sea  $C \leftarrow \{u_j\}$  para  $u_j \in \delta(v_i)$ ,  $P_{iu_j} \leftarrow |I|/r_i$ ;
10  regresar;
11  $l_1 \leftarrow \lceil (|C|/2) \rceil$ ,  $l_2 \leftarrow \lfloor (|C|/2) \rfloor$ ;
12  $[C_1, C_2] \leftarrow Elige(l_1, l_2, C)$ ;
13 // La función Elige agrega aleatoriamente  $l_1$  elementos
14 // distintos de  $C$  en  $C_1$  y en  $C_2$  coloca el resto
15  $k \leftarrow Rnd(s + l_1 - 1, t - l_2)$ ;
16 Divide( $[s, k]$ ,  $C_1$ );
17 Divide( $[k+1, t]$ ,  $C_2$ );
18 regresar;

```


3.6 Número de hojas en el árbol

El comportamiento general del número de hojas resultantes en el árbol obtenido fue similar en todos los casos, esto se puede observar en las Figuras 5, 10 y 11.

En una gráfica aleatoria con $n = 1000$ vértices y $m = 4000$ aristas, es razonable que el número de hojas presentadas en el árbol no varíe dentro de un rango muy grande, por otro lado desde K_{1000} deberían poder obtenerse árboles generadores con una cantidad más variable de hojas, pero no fue así. Esto posiblemente se deba a que la proporción existente de árboles generadores que pueden ser extraídos desde K_{1000} , en su mayoría poseen tales características. Así, para obtener otros tipos de árboles como los que tienen 999 hojas, de los cuales hay 1000, es necesario que Generar se ejecute muchas veces, pues el conjunto de árboles distintos en K_{1000} es 1000^{998} (Formula de Cayley 1889).

3.7 Valencias promedio y máxima de los vértices internos del árbol

Tampoco existió variación en estos resultados, lo cual se puede observar en la Figura 12. La máxima valencia fue de diez en todas las pruebas realizadas y la tendencia de la valencia de los vértices internos del árbol se mantuvo nuevamente entre dos y tres.

A continuación presentamos un mecanismo de modificación de las matrices de transición con el cual es posible obtener de forma más eficiente, árboles generadores con características diferentes a los anteriormente presentados.

4 Saag modificando las matrices de transición mediante la ponderación de vértices

Realizamos una modificación del orden en que son asignadas las distribuciones de probabilidad en las vecindades de vértices asociadas a cada gráfica G , para que dentro del conjunto de vértices en V existan ciertos vértices atractores durante el paseo aleatorio, de modo que algunos

vértices siempre tendrán mayor probabilidad de ser visitados desde sus vecinos. Esto se realizó mediante el siguiente procedimiento:

Ordenar-Matriz

1. Sea $G = (V, A)$ una gráfica.
2. Sea $B: V \rightarrow \{1, \dots, |V|\}$ una función biyectiva.
3. Sea P una matriz de transición asociada a G .
4. Para cada $v_i \in V$ si $P_{i, v_{j_1}}, \dots, P_{i, v_{j_{|\delta(v_i)|}}}$ son las probabilidades para transitar de v_i a sus vecinos ordenadas descendientemente y $v_{k_1}, \dots, v_{k_{|\delta(v_i)|}}$ son los vecinos de v_i ordenados descendientemente de acuerdo con la función B , entonces:

$$PB_{i, k_1} = PI_{i, j_1}, \dots, PB_{i, k_{|\delta(v_i)|}} = PI_{i, j_{|\delta(v_i)|}} \quad (4)$$

Con el procedimiento Ordenar-Matriz, se obtiene una matriz de transición no uniforme que además mantiene la propiedad de que el vértice con la mayor prioridad asignada según B , será siempre el más probable de visitar desde cada uno de sus vecinos. De este modo se crean ciertos vértices atractores durante el paseo aleatorio en G .

Esto produce que sean más fáciles de generar árboles generadores desde las mismas instancias de gráficas, pero con distintas características a las mencionadas en las secciones anteriores.

Para cada descripción presentada, las matrices de transición asociadas a cada gráfica G son las mismas que las utilizadas en la sección anterior, pero realizando una modificación a cada matriz de transición mediante el procedimiento Ordenar-Matriz.

4.1 Longitud de paseo

En la Figura 13, se puede observar que con el uso de matrices de transición modificadas en Generar, son generados paseos aleatorios con longitudes aún por debajo de $O(n^3)$ en el caso de K_{1000} , pero en gráficas con una menor cantidad de aristas dicha cota es alcanzada.

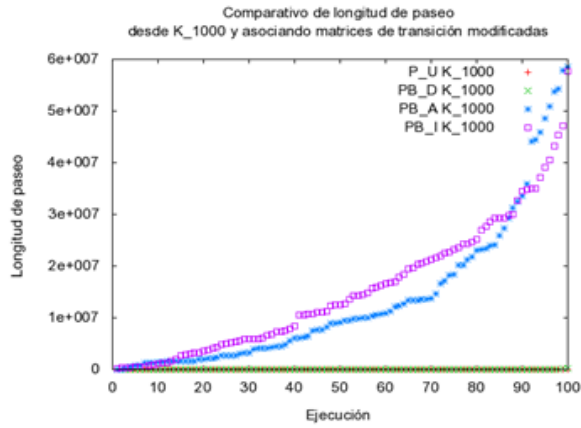


Fig. 13. Longitudes de paseo V

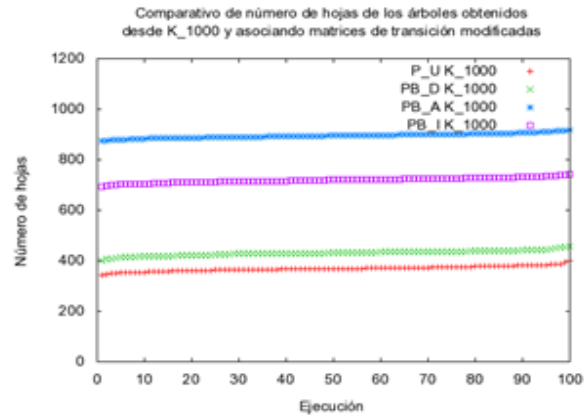


Fig. 16. Número de hojas de los árboles IV

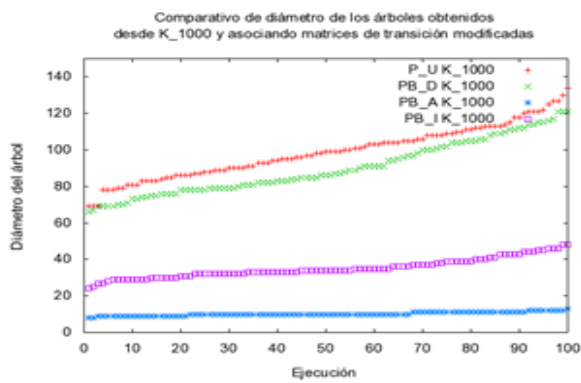


Fig. 14. Diámetro de los árboles IV

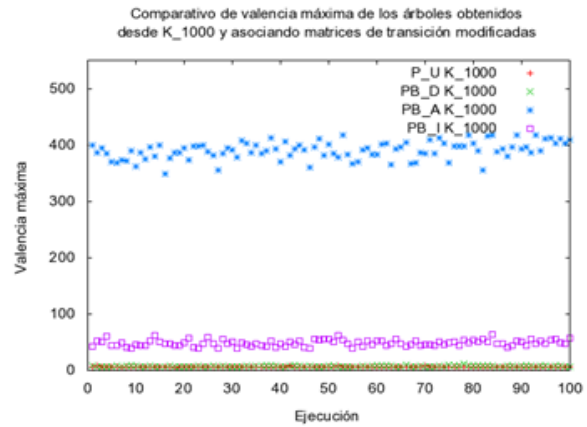


Fig. 17. Valencia máxima de los vértices internos de los árboles

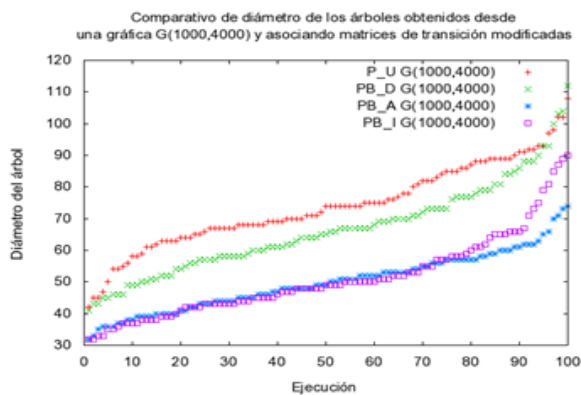


Fig. 15. Diámetro de los árboles V

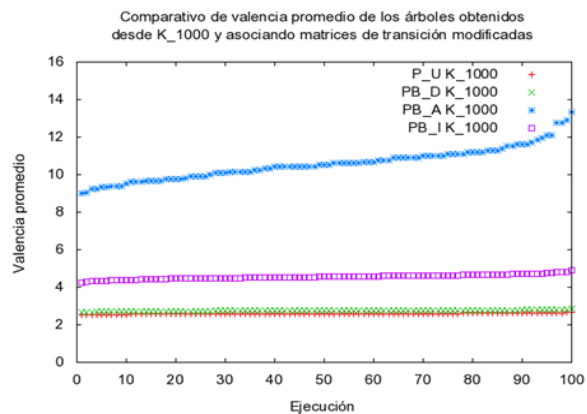


Fig. 18. Valencia promedio de los vértices internos de los árboles

4.2 Diámetro del árbol

En la Figura 14, se observa que existen variaciones más marcadas en cuanto al diámetro de los árboles generadores obtenidos. Por ejemplo, con el uso de la matriz de transición modificada PB_{AG} se obtienen los diámetros más bajos, teniendo árboles con un diámetro no mayor a 20 en el caso de K_{1000} .

Con la asociación de la matriz de transición modificada PB_{IG} a tal gráfica completa, se obtuvieron diámetros ligeramente mayores a los obtenidos mediante la asociación de la matriz de transición modificada PB_{AG} a la misma gráfica. Con la asociación de las matrices de transición modificadas PB_{UG} y PB_{DG} , se obtuvo un comportamiento similar. En esta ocasión los diámetros varían bastante con respecto a los presentados en la Figura 9.

Tales variaciones también ocurren en gráficas con un menor número de aristas, como lo muestra la Figura 15. Por lo tanto si la diversidad de los diámetros de los árboles generadores posibles de una gráfica es grande, pueden ser asociadas diferentes matrices de transición modificadas para obtener diferentes tipos de árboles generadores.

4.3 Número de hojas en el árbol

El número de hojas también presenta grandes diferencias. Con el uso de las matrices de transición modificadas PB_{AG} y PB_{IG} , se obtuvo una cantidad de hojas mayor al 50% del total de vértices en el árbol generador, contrario a lo que sucedía con la matriz de transición uniforme. Ver la Figura 16.

Aunque dichas cantidades presentan comportamientos uniformes en cada caso, es posible observar que con el uso de una cierta matriz de transición no uniforme modificada, se pueden obtener árboles con cantidades de hojas diferentes a los que resultan asociando matrices de transición uniformes.

4.4 Valencias promedio y máxima de los vértices internos del árbol

Con la asociación de las matrices de transición modificadas PB_{AG} y PB_{IG} se pueden obtener cantidades mayores al 50% de hojas en el árbol generador, esto implica que las valencias de los vértices internos del árbol sean grandes. Por ejemplo, hubo casos en los que la valencia máxima de un vértice en el árbol fue de ≈ 400 asociando la matriz de transición PB_{AG} y ≈ 50 asociando la matriz de transición PB_{IG} . Ver Figura 17.

En la Figura 18 se puede observar una tendencia uniforme del promedio de valencia de los vértices internos del árbol en cada caso, al mismo tiempo se observan tendencias diferentes asociando distintos tipos de matrices de transición modificadas.

Aunque presentamos los resultados con sólo algunas de las instancias utilizadas, consideramos que con estas ideas se pueden elegir eficientemente conjuntos de árboles con determinadas características.

5 Conclusiones

Entre mayor variabilidad exista entre las probabilidades de la vecindad de cada vértice y mediante el procedimiento Ordenar-Matriz, se pueden obtener árboles con un buen equilibrio entre su diámetro, la valencia de sus vértices internos y su número de hojas.

La modificación de las matrices de transición con distribuciones de probabilidad no uniformes mediante la ponderación de vértices es esencial para obtener árboles generadores con características diferentes a los mencionados en las secciones 2 y 3. Además permite mayor control sobre las características de los árboles generados mediante *Generar*.

Con el uso de las matrices de transición modificadas es posible seleccionar ciertos conjuntos de árboles de forma aleatoria de entre la enorme cantidad de posibilidades existentes en la mayoría de las gráficas, aún si la proporción de aristas es pequeña.

Los árboles resultantes poseen un cierto balance en cuanto a sus propiedades. Cuando un

árbol presenta un diámetro bajo, los vértices internos del árbol tendrán valencias altas y por ende habrá una mayor cantidad de hojas, a diferencia de cuando el árbol resultante es de mayor diámetro.

Consideramos que la forma más adecuada de obtener árboles con ciertas propiedades, es utilizar las matrices de transición modificadas generadas por el Algoritmo 2 ya que variando el rango de distribución de la probabilidad son obtenidos árboles aleatorios con diferentes propiedades.

Estas mismas ideas se pueden aplicar en otros algoritmos aleatorios que utilicen formas similares de selección de árboles generadores en gráficas, como el caso del algoritmo aleatorio introducido por D. B. Wilson descrito en la introducción de este trabajo.

Referencias

1. **Aldous, D.J. (1990).** The random walk construction of uniform spanning trees and uniform labelled trees. *SIAM Journal on Discrete Mathematics*, 3(4), 450–465.
2. **Broder, A. (1989).** Generating random spanning trees. *30th Annual Symposium on Foundations of Computer Science*, Research Triangle Park, North Carolina, USA, 442–447.
3. Graph coloring and its generalizations (s.f.). Retrieved from <http://mat.gsia.cmu.edu/COLOR03/>
4. **Kelner, J.A. & Madry, A. (2009).** Faster generation of random spanning trees. *50th Annual IEEE Symposium on Foundations of Computer Science*, Atlanta, Georgia, USA, 13–21.
5. **Sagols, F. & Morales-Luna, G. (2010, 8 de septiembre).** Two identification protocols based on Cayley graphs of Coxeter groups. *Cryptology ePrint Archive*. Retrieved from <http://eprint.iacr.org/2010/470>.
6. **Wilson, D.B. (1996).** Generating random spanning trees more quickly than the cover time. *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, Philadelphia, Pennsylvania, USA, 296–303.



Guillermo Benito Morales-Luna es Investigador Titular en el Departamento de Computación del Cinvestav-IPN, es Licenciado en Física y Matemáticas por la ESFM-IPN, Maestro en Ciencias con especialidad en Matemáticas por el Cinvestav-IPN, y Doctor en

Ciencias Matemáticas por el Instituto de Matemáticas de la Academia Polaca de Ciencias, en Varsovia, Polonia. Áreas de interés: los Fundamentos Matemáticos de Computación, Lógica y Deducción Automática, Criptografía y Teoría de la Complejidad. Ha sido profesor en el IPN y en la Benemérita Universidad Autónoma de Puebla. Ha realizado dos estancias sabáticas en el Instituto Mexicano del Petróleo. Es mexicano por nacimiento y le fue concedida la ciudadanía polaca.



Feliú Davino Sagols-Troncoso es Doctor en Ingeniería Eléctrica. Actualmente es Profesor Titular en el Cinvestav-IPN en la ciudad de México, adscrito al Departamento de Matemáticas. Áreas de interés: Teoría de la

Computación, Desarrollo de Sistemas Accesibles Vía Internet, Combinatoria, Teoría de Gráficas y Teoría Topológica de las Gráficas. Ha desarrollado múltiples herramientas para representar mapas combinatorios, que aplicados a la esfera, producen métodos excelentes para construir Sistemas de Información Geográfica.



Sergio Luis Pérez-Pérez es Ingeniero en Computación (2004-2009) por la Universidad Autónoma Metropolitana (UAM). Actualmente se encuentra haciendo sus estudios de Maestría en el Depto. de Computación en el Cinvestav-IPN, en la ciudad de México (desde 2009). Trabajó como asistente de profesor investigador

en la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, de abril de 2007 a abril de 2010. Ha participado en diversos concursos de programación: en el ACM-ICPC zona México y América Central, en el Abierto Nacional de Programación Ants, en el 10 Concurso Nacional de Programación, entre algunos otros. Áreas de interés: Análisis y Diseño de Algoritmos, Optimización Combinatoria, Teoría de Gráficas y Criptografía.

Article received on 06/04/2011; accepted on 25/02/2012.