Reporte:
Cadenas de Markov y modelo PageRank

Autor: ***Mario E. Ferreyra***

Docentes:

- ***Dra. Ana Georgina Flesia***
- ***Dr. Jorge A. Sánchez***
- ***Dr. Marcelo Lares***

Asignatura: ***Modelos y Simulación***

14 de Junio del 2016

Índice

1. Introducción

Lo que buscamos con este reporte es poder entender y sacar con respecto a características especificas sobre grafos, para lo cual usaremos como herramienta de estudio para dichos grafos las Cadenas de Markov.

Las características que sobre las que trabajaremos son las siguientes:

- Distribución Estacionaria
- Tiempos de Cruce
- Tiempo de Cubrimiento

En la siguiente sección definiremos los conceptos de dichas características mencionadas.

También trabajaremos sobre el modelo PageRank, el cual es una marca registrada y patentada por Google que ampara una familia de algoritmos utilizados para asignar de forma numérica la relevancia de las páginas web indexados por un motor de búsqueda. Dicho modelo es utilizado por el motor de búsqueda Google para ayudarle a determinar la importancia o relevancia de una página.

Por ultimo analizaremos dos estrategias para ayudar a que un Spammer, que agrego su sitio a la web, para que mejore el ranking de su pagina.

Para la obtención de la conclusiones y resultados, utilizaremos dos grafos particulares para poder analizar en ellos las características previamente mencionadas.

2. Definiciones, Métodos, Estrategias y Grafos

En esta sección definiremos los conceptos nombrados en la introducción, métodos a usar, estrategias propuestas por el Spammer y los grafos que utilizaremos para concluir resultados.

2.1. Definiciones de conceptos

Tiempo de Cruce: El tiempo de cruce para un nodo es la cantidad de pasos que necesita un caminante aleatorio para volver a pasar por dicho nodo.

Tiempo de Cubrimiento: El tiempo de cubrimiento para un grafo es la cantidad de pasos que necesita un caminante aleatorio para visitar cada nodo del grafo al menos una vez, comenzando en un nodo elegido de forma aleatoria.

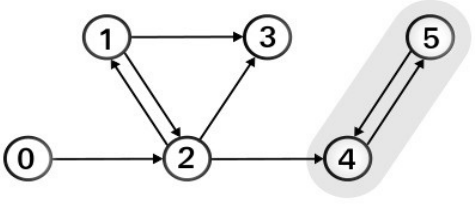
Matriz de Transición: Es una matriz cuadrada de tamaño n (cantidad de nodos del grafo) la cual nos indica los vecinos y la probabilidad de movernos a ellos.

Tendremos dos matrices de transición para cada grafo. Las nombraremos de la siguiente forma:

- **Matriz de Transición Original:** Es una matriz de transición, en la cual la probabilidad de pasar a un vecino es equis-probable, es decir, que la probabilidad es la misma para todos los vecinos, como por ejemplo la Matriz de Transición P de la Figura 1.

- **Matriz de Transición Modificada:** Es una matriz de transición generada por el modelo PageRank que tiene un factor de dumping α (alfa) entre 0 y 1. La intuición de dicho modelo es la de capturar el comportamiento de un "navegador aleatorio" que con probabilidad (1-α) se aburre y efectúa un salto a un sitio arbitrario.

  Este modelo genera una matriz de transición que soluciona el problema que tendría la Matriz de Transición Original, si hubiera "nodos colgados" (por ejemplo, nodo 3 en la Figura 1), nodos que no puedan ser alcanzados en un número finito de pasos (por ejemplo, nodo 0 en la Figura 1) o la existencia de bucles absorbentes (por ejemplo, nodos 4-5 en la Figura 1).



$$P = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 1/3 & 0 & 1/3 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Figura 1

2.2. Métodos para la obtención de la Distribución Estacionaria

Para la obtención de la distribución estacionaria se utilizaron dos métodos:

- **Método 1:** Se empleo un caminante aleatorio para que se moviera sobre el grafo según las probabilidades que figuren en la Matriz de Transición. Una vez que se realizaron X pasos, el caminante se detendrá, de ahí tomaremos el nodo en el que se detuvo. Realizando esto N veces, es decir, correr hacer N caminatas sobre el grafo podremos determinar la Distribución Estacionaria de dicho grafo como la cantidad de caminantes que terminaron en un nodo sobre el total de N caminatas hechas.

- **Método 2:** Un método mas analítico para determinar la Distribución Estacionaria es usar el "Método de las Potencias", un método iterativo pero con una baja velocidad de convergencia.

2.3. Estrategias propuestas por el Spammer

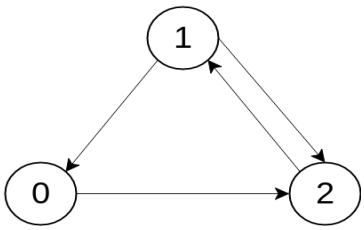
Como dijimos en la introducción, un Spammer desea que su sitio web mejore su ranking, el mismo elabora dos estrategias para así mejorar su rankings. Dichas estrategias son las siguientes:

- **Estrategia A:** Crear K paginas ficticias y hacer que todas ellas apunten a su sitio. Después de esta adición, el numero total de paginas de la web (suponiendo que había n paginas) sera de n + K + 1 paginas.

- **Estrategia B:** Hackear K paginas de las n existentes, agregándoles enlaces a su pagina.

2.4. Grafos a utilizar y su representación

Primero que nada aclararemos como es que serán representados los grafos en este reporte. Por ejemplo, el grafo:



Se representa de la siguiente como:

$$G = ((2,), (0, 2), (1,))$$

Donde el cada índice de la tupla representa un nodo y a su vez cada nodo indica los vecinos a los cual puede desplazarse.

Ahora nombraremos y definiremos los grafos que vamos a utilizar para su posterior análisis.

$$G1 = ((2,), (0, 5, 3), (6, 0, 2, 7, 4), (3, 1),$$
$$(1, 4, 2, 9, 5, 6, 3, 8, 7), (0, 4, 5), (6, 9, 0),$$
$$(3, 9), (7, 2, 6, 5, 3, 1, 0), (2, 0))$$

G1 tiene un total de 10 nodos.

También utilizaremos un grafo llamado **G2** que tiene 100 nodos, por dicha cantidad nodos no especificaremos la adyacencia entre sus nodos.

3. Algoritmos

En esta sección mostraremos los pseudocódigos de los algoritmos que fueron empleados para la obtención de los resultados.

NOTA: En algunos de los pseudocódigos se nombra a alguna de las siguientes funciones:

- g2p_pagerank
- power_iter_one_step

Dichas funciones ya son conocidas, están dadas y definidas; por lo que no daremos su pseudocódigo en este reporte.

3.1. Pseudocódigo Método 1

```
function distribucionEstacionaria(grafo, matriz_trans, pasos, simulaciones):
    length_grafo ← len(grafo) # Tamaño del grafo
    apariciones ←Generar vector de tamaño length_grafo con todos 0's
    dis_est ←Generar vector de tamaño length_grafo con todos 0's

    for 1 to simulaciones:
        nodo ← Elegir un nodo aleatorio del grafo
        for 1 to pasos-1:
            vecino ← Elegir un vecino del nodo elegido de forma aleatoria
            nodo ← vecino
        end for
        apariciones[nodo] ← apariciones[nodo] + 1
    end for
    for i in length_grafo:
        dist_est[i] ← apariciones[i] / simulaciones
    end for

    return dist_est

end function
```

3.2. Pseudocódigo Método 2

```
function metodoPotencias(grafo, matriz_trans, pasos):
    length_grafo ← len(grafo) # Tamaño del grafo
    pi ← Generar vector de tamaño length_grafo con todos 1's
    for 1 to pasos:
       pi ← power_iter_one_step(pi, matriz_trans) # Realiza un paso del método de las potencias
    end for

    return pi

end function
```

3.3. Pseudocódigo Tiempo de cruce

```
function visitarNodo(nodo, matriz_trans, length_grafo, simulaciones)
    t ← 0
    for 1 to simulaciones:
       nodo_inicial ← nodo
       while True:
          vecino ← Elegir un vecino del nodo_inicial elegido de forma aleatoria
          t ← t + 1
          # Si el vecino es el nodo con el cual ya empeze corto la búsqueda
          if nodo = vecino:
             break
          end if
          nodo_inicial ← vecino
        end while
    end for
   pasos ← t / simulaciones # Promedio de pasos para cruzarme de nuevo con el nodo

   return pasos

end function



function tiempoCruce(grafo, matriz_trans, simulaciones):
    length_grafo ← len(grafo) # Tamaño del grafo
    tiempos ← []
    media ← 0
    # Calculamos el tiempo de cruce para todos los nodos del grafo
    for nodo in grafo:
        Agregar (visitarNodo(nodo, matriz_trans, length_grafo, simulaciones)) a tiempos
    end for

    media ← (Sumar todos los elementos de tiempos) / simulaciones # Media de tiempos

    return tiempos, media

end function
```

3.4. Pseudocódigo Tiempo de cubrimiento

```
function tiempoCubrimiento(length_grafo, matriz_trans):
    nodo ← Elegir un nodo aleatorio del grafo
    nodos_vistos ← [nodo] # nodos por el caminante paso
    pasos ← 1

    # Corremos el caminante aleatorio hasta pasar por todos los nodos
    while len(nodos_vistos) < length_grafo:
       nodo ← Elegir un vecino del nodo elegido de forma aleatoria
       pasos ← pasos + 1
       if nodo not in nodos_vistos:
          Agregar  (nodo) a nodos_vistos
       end if
    end while

    return pasos

end function

function cubrirGrafo(grafo, matriz_trans, simulaciones):
    length_grafo ← len(grafo) # Tamaño del grafo
    pasos ← 0
    for 1 to simulaciones:
       pasos ←  pasos + tiempoCubrimiento(length_grafo, matriz_trans)
    end for
    tiempo ← pasos / simulaciones

    return tiempo

end function
```

3.5. Pseudocódigo Estrategia A

```
function paginasFictias(grafo, K):
    n ← len(grafo) # Tamaño del grafo
    Agregamos un nodo vacío al grafo # Mi pagina web
    for 1 to K:
        Agregamos un nodo que apunte al nodo n+1 al grafo # Agrego K paginas a la web
    end for

    N ← n + K + 1 # Nuevo tamaño del grafo
    P ← g2p_pagerank(grafo, 0.85) # Matriz de transición con PageRank con α = 0.85
    dist_est ← metodoPotencias(N, P, 100)
    ranking ← dist_est[n]

    return ranking

end function
```

3.6. Pseudocódigo Estrategia B

```
function hackearPaginas(grafo, K):
    n ← len(grafo) # Tamaño del grafo
    Agregamos un nodo vacío al grafo # Mi pagina web
    nodos_hackeados ← []
    for 1 to K:
        nodo ← nodoAleatorio(n)
        while nodo in nodos_hackeados:
            nodo ← Elegir un nodo aleatorio del grafo
        end while
        Agregar (nodo) a  nodos_hackeados
        Agregamos un enlace al nodo hacia el nodo n+1
   end for

   N ← n 1 # Nuevo tamaño del grafo
   P ← g2p_pagerank(grafo, 0.85) # Matriz de transición con PageRank con α = 0.85

   dist_est ← metodoPotencias(N, P, 100)
   ranking ← dist_est[n]

   return ranking

end function
```

4. Resultados

En esta sección se mostrar los resultados obtenidos sobre la Distribución Estacionaria, Tiempos de Cruce y Tiempos de cubrimiento, sobre los grafos G1 y G2 antes mencionados.

4.1. Distribución Estacionaria

Para los resultados que se mostraran a continuación se utilizo el grafo G1, la Matriz de Transición Original y la Modificada (α = 0.85) con ambos métodos mencionados.

Para el método 1 se utilizo 100 pasos para el caminante y para el método 2 se utilizo 100 pasos del método de las portencias.

4.1.1. Distribución Estacionaria con Método 1

Matriz de Transición Original:

[0.162 , 0.077 , 0.262 , 0.133 , 0.078 , 0.055 , 0.094 , 0.062 , 0.009 , 0.068]

Matriz de Transición Modificada (α = 0.85):

[0.155 , 0.081 , 0.233 , 0.127 , 0.085 , 0.067 , 0.088 , 0.067 , 0.025 , 0.072]

4.1.2. Distribución Estacionaria con Método 2

Matriz de Transición Original:

[0.163 , 0.077 , 0.261 , 0.133 , 0.079 , 0.053 , 0.093 , 0.062 , 0.009 , 0.071]

Matriz de Transición Modificada (α = 0.85):

[0.157 , 0.081 , 0.229 , 0.132 , 0.081 , 0.068 , 0.09 , 0.064 , 0.023 , 0.075]

Como podemos notar, las cuatro distribuciones obtenidas con ambos métodos y matrices son muy parecidas. Por lo que todos darán prácticamente la misma figura, como se puede apreciar en la Figura 2.

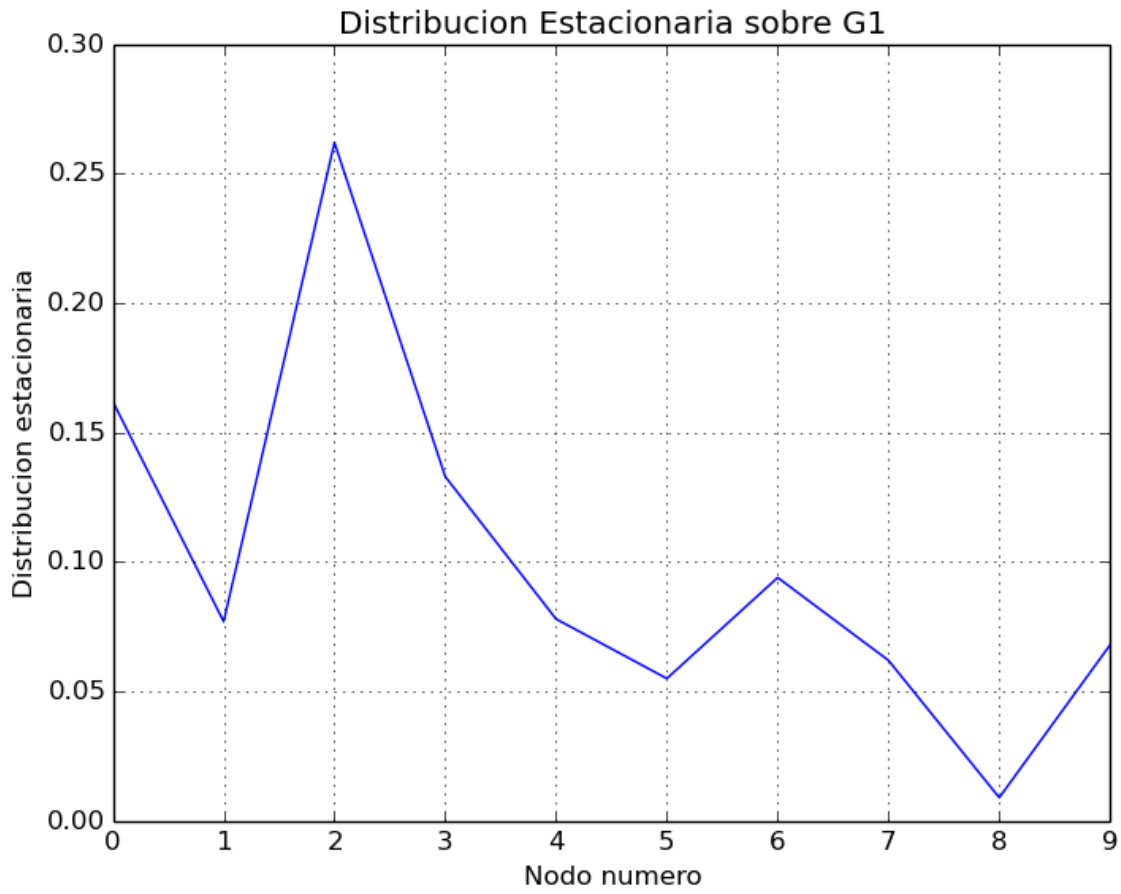


Figura 2

4.2. Tiempos de Cruce

Vamos a evaluar el tiempo de cruce sobre todos los nodos de los grafos G1 y G2, mostrando una lista con el tiempo de cruce de cada nodo para G1 y su media. Para G2 solo mostraremos la media ya que como dijimos anteriormente a la hora de mostrar su estructura, posee una gran cantidad de nodos y seria ilegible poner el tiempo de todos ellos.

4.2.1. Tiempos de Cruce para G1

Con Matriz de Transición Original:

[6.14 , 13.39 , 3.48 , 5.07 , 12.83 , 19.2 , 10.71 , 18.02 , 112.73 , 13.27]

Media de Tiempos = 21.484

Con Matriz de Transición Modificada (α = 0.85):

[7.07 , 12.38 , 4.71 , 6.6 , 10.94 , 13.62 , 8.95 , 15.94 , 46.09 , 12.14]

Media de Tiempos = 13.844

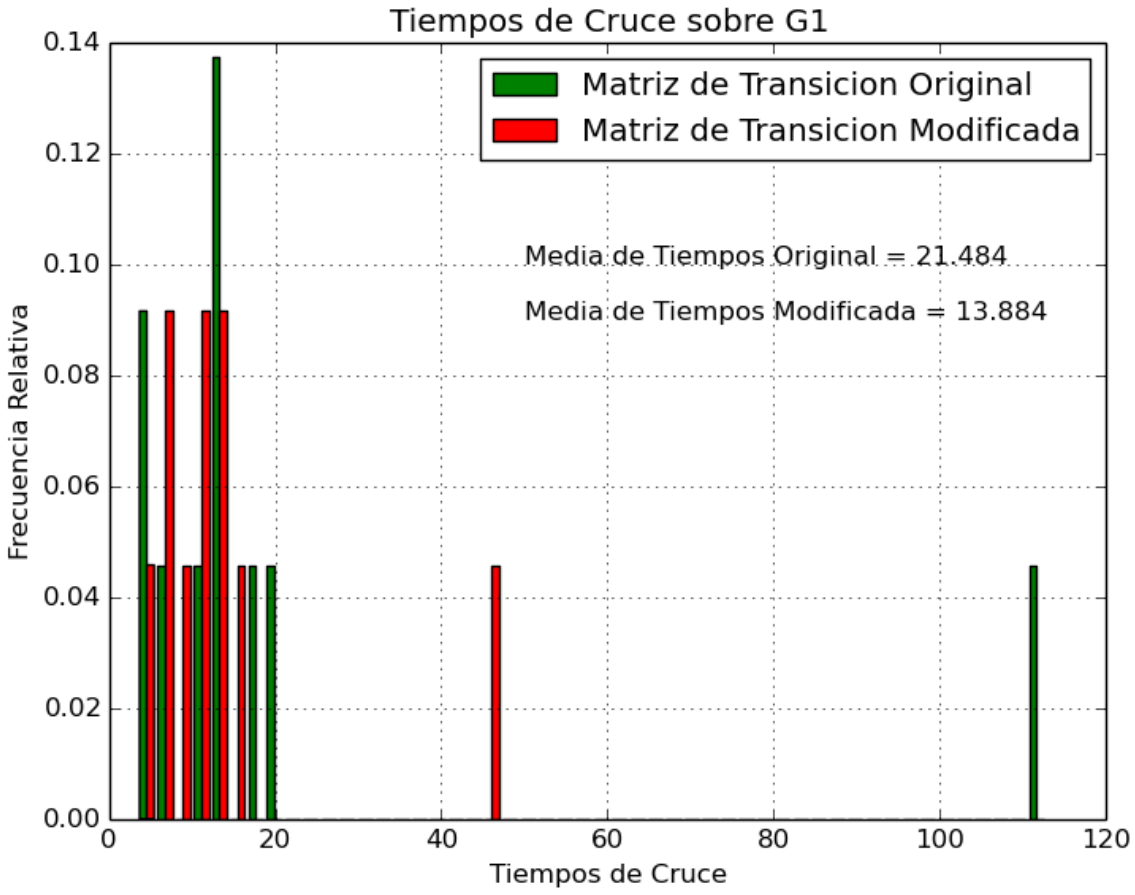


Figura 3

4.2.2. Tiempos de Cruce para G2

Con Matriz de Transición Original:

Media de Tiempos = 103.501

Con Matriz de Transición Modificada (α = 0.85):

Media de Tiempos = 102.522

4.3. Tiempos de Cubrimiento

Vamos a calcular el tiempo de cubrimiento sobre G1 y G2 usando las dos posibles Matrices de Transición, en la modificada vamos a ir variando el α y compararlos mediante el uso de una tabla.

Tiempos de Cubrimiento con Matriz de Transición Original:

G1	G2
107.641	548.714

Tiempos de Cubrimiento con Matriz de Transición Modificada:

α	G1	G2
0.1	30.449	523.399
0.2	29.942	519.633
0.3	31.549	522.228
0.4	32.85	529.887
0.5	35.213	522.984
0.6	36.81	525.198
0.7	42.012	535.352
0.8	49.959	539.09
0.85	54.777	543.288
0.9	63.136	544.125
0.99	100.901	548.871

En la Figura 4 y 5 podemos ver de forma gráfica como van aumentando los tiempos de cubrimiento según va aumentando el alfa para los grafos G1 y G2 correspondientemente.

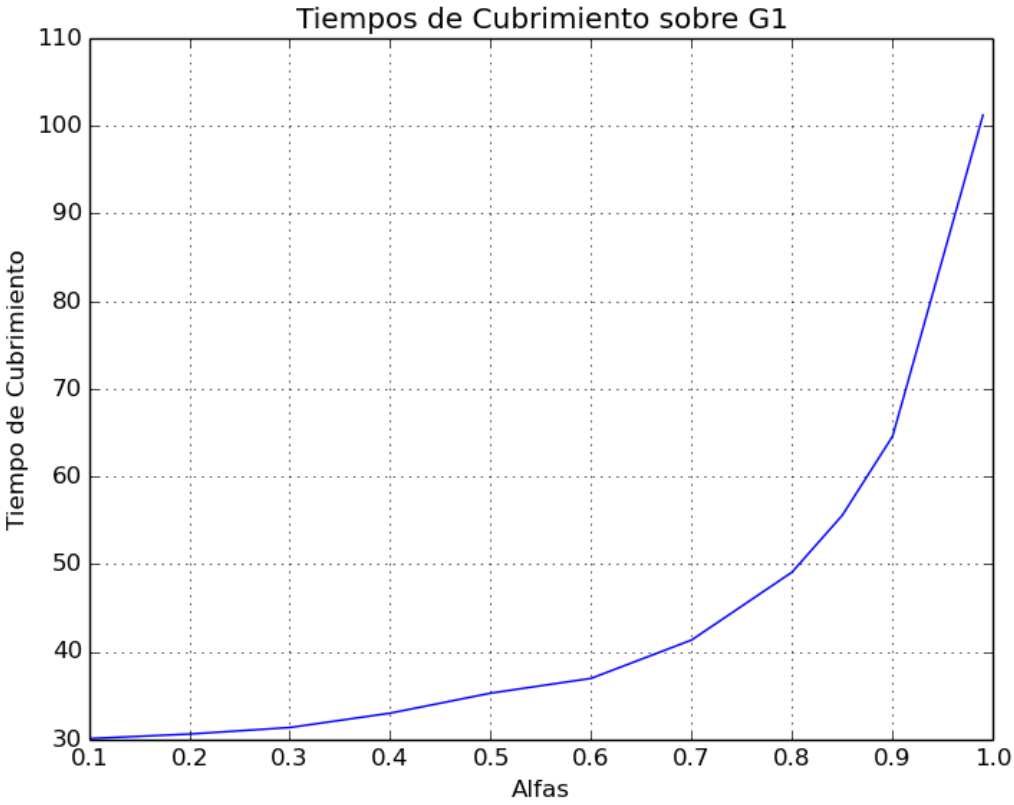


Figura 4

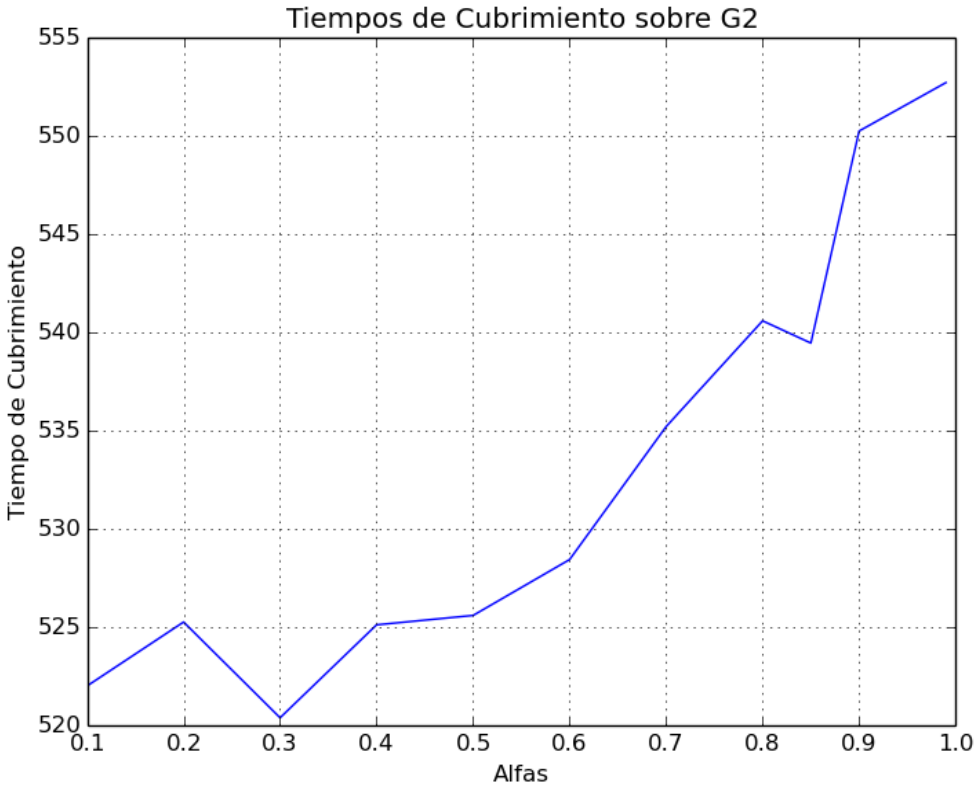


Figura 5

4.3.1. Tiempos de Cubrimiento para grafos de distintos tamaños

Veremos los tiempos de cubrimiento al variar el numero de nodos de los grafos, usando la Matriz de Transición Modificada (α = 0.85).

Nodos	Tiempo de Cubrimiento
5	36.053
10	31.408
30	134.328
50	243.338
75	401.133
100	551.737

Podemos apreciar en la tabla que a medida que aumentamos los nodos del grafo, los tiempos de cubrimiento también aumentan, este aumento es prácticamente lineal. Esto podemos apreciarlo de una mejor manera mirando el la Figura 6.

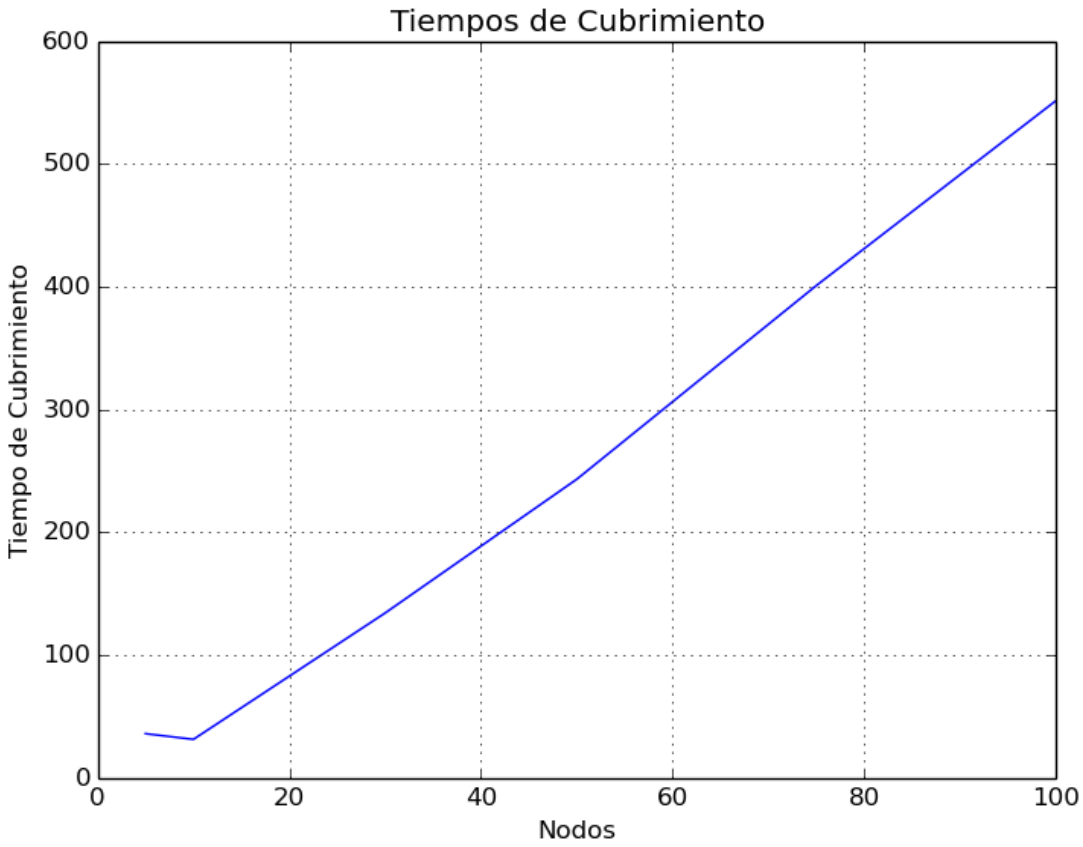


Figura 6

4.4. Problema del Spammer

En esta parte de los resultados vamos a exponer los resultados obtenidos con las dos estrategias que se le han ocurrido al Spammer para subir lo que más se pueda el ranking de su pagina.

4.4.1. Estrategias A y B sobre G1

Vamos a analizar dichas estrategias antes mencionadas sobre el grafos G1, comparando como sube o baje el ranking de su pagina según la cantidad K de paginas ficticias que se creen y hacer que todas ellas apunten a nuestra pagina, o que se hackeen agregándoles enlaces para que apunten a nuestra pagina.

Además vamos a comparar los rankings obtenido usando el Método 2 anteriormente mencionado y la Matriz de Transición Modificada ($\alpha = 0.85$).

K	Ranking A	Ranking B
1	0.023	0.036
3	0.049	0.14
5	0.072	0.12
7	0.09	0.167
10	0.112	0.205

Dicha comparación de estrategias las veremos de una mejor forma en el histograma de la Figura 7.

4.4.2. Estrategias A y B sobre G2

Ahora procederemos a hacer lo mismo que en la Sección 4.4.1. pero sobre el grafo G2.

Además vamos a comparar los rankings obtenido usando el Método 2 anteriormente mencionado y la Matriz de Transición Modificada ($\alpha = 0.85$).

K	Ranking A	Ranking B
10	0.014	0.005
25	0.031	0.008
50	0.057	0.017
75	0.08	0.021
100	0.101	0.028

Dicha comparación de estrategias las veremos de una mejor forma en el histograma de la Figura 8.

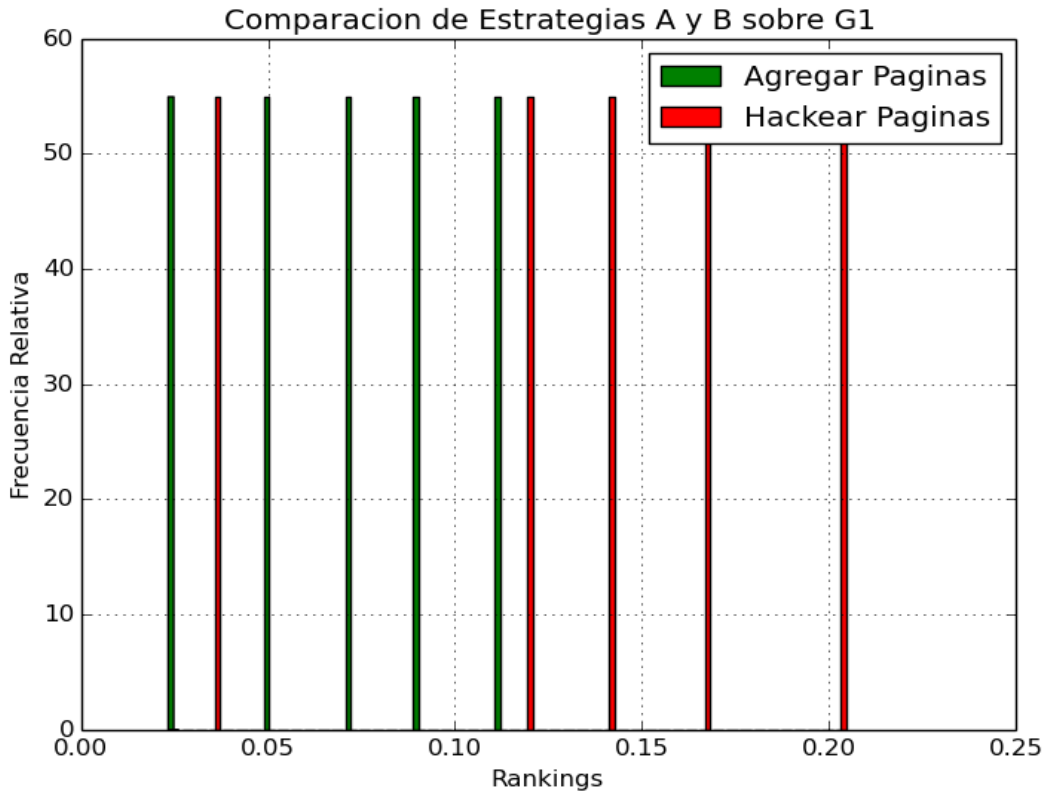


Figura 7

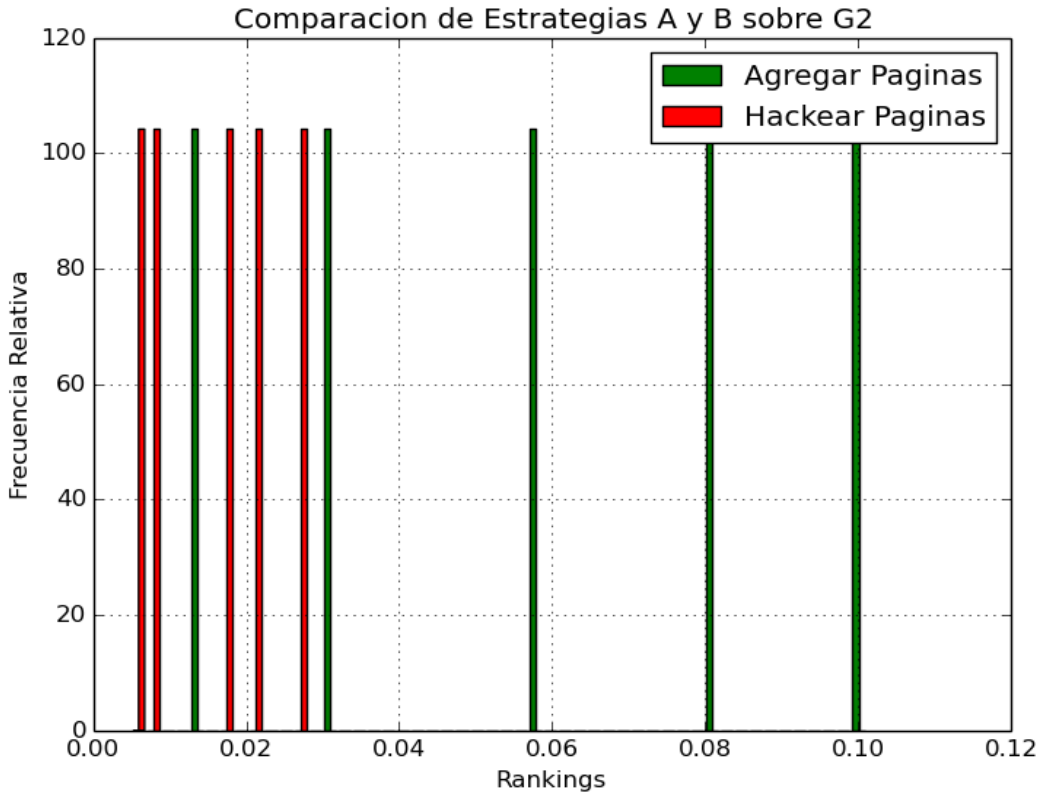


Figura 8

5. Conclusiones

A continuación expondremos las conclusiones que hemos obtenido al analizar los resultados y los histogramas mostrados en anteriores secciones:

Distribución Estacionaria

Con respecto a la distribución estacionaria calculada por medio del Método 1 y 2 con las diferentes matrices de transición, podemos apreciar que los resultados son prácticamente los mismo, por lo que concluimos que tanto el método del caminante aleatorio, como el de las potencias son muy buenos para grafos que sean conexos.

Tiempos de Cruce

Como podemos apreciar en la Figura 3 y la lista de tiempos mostrada, en los tiempos de cruce para G1 con la Matriz de Transición Original, hay un nodo el cual lo cruzamos un gran cantidad de veces, esto quiere decir que durante las caminatas realizadas debe pasar mucho tiempo para volver a pasar por sobre dicho nodo. Esto genera que el tiempo medio de cruce aumente considerablemente.

Pero también podemos ver que si usamos la Matriz de Transición Modificada el tiempo de cruce de cada nodo disminuye, especialmente el del que nombramos anteriormente, esto causa que el tiempo medio de cruce disminuya bastante con respecto al obtenido con la Matriz de Transición Original.

Pero para grafos grandes como G2 en el tiempo medio de cruce no cambia casi nada si usamos la Matriz de Transición Original o la Modificada.

Por lo que podemos concluir que para grafos chicos, el caso de G1, el tiempo medio de cruce varia mucho si usamos una u otra matriz, pero para grafos grades no varia prácticamente nada.

Tiempos de Cubrimiento

Para los tiempos de cubrimiento, usamos la Matriz de Transición Original y la Modificada variando el α.

Analizando los resultados que se muestran en la tabla con el uso de la Matriz de Transición Original vemos que los tiempos de cubrimiento para ambos grafos son altos, en contraste con el uso de la Matriz de Transición Modificada notamos que a medida que vamos aumentando el α los tiempos de cubrimiento de los grafos van aumentando hasta llegar a un punto que se vuelve exponencial el aumento del tiempo, esto que acabamos de decir lo podemos ver claramente en las Figuras 4 y 5. Por lo que mientras mas chico se el α, menor sera el tiempo de cubrimiento.

Otro punto a discutir era como afectaba el hecho de aumentar los nodos del grafo al tiempo de cubrimiento del mismo, mirando tranquilamente la tabla de la Sección 4.2.1 vemos que el aumento del tiempo es lineal, en la Figura 5 también podemos evidenciar esta forma de aumento.

Problema del Spammer

Analizando los rankings y histogramas obtenidos en la Sección 4.4. con ambas estrategias y para los dos grafos, llegamos a la conclusión de que para grafos chicos conviene hackear pagina para así poder aumentar el ranking de su pagina, en contraste con lo que pasa para los grafos grandes donde es mejor agregar paginas para aumentar el ranking.

Por lo que concluimos que depende del tamaño del grafo conviene el aplicar una u otra estrategia.