

Construcción de Redes de Pequeño Mundo mediante Selección Sesgada *

Andrés SEVILLA DE PABLO¹, Pilar MANZANO¹, Alberto MOZO², M. Araceli LORENZO², Jose Luis LÓPEZ-PRESA³, and Antonio FERNÁNDEZ ANTA⁴

¹ Dpto Informática Aplicada, U. Politécnica de Madrid, Madrid, Spain
asevilla@eui.upm.es, pmanzano@eui.upm.es

² Dpto Arquitectura y Tecnología de Computadores, U. Politécnica de Madrid, Madrid, Spain
amozo@eui.upm.es, alorenzo@eui.upm.es

³ DIATEL, U. Politécnica de Madrid, Madrid, Spain
jllopez@diatel.upm.es

⁴ Institute IMDEA Networks, Leganés, Spain
antonio.fernandez@imdea.org

Abstract. En la actualidad las redes de Pequeño Mundo están presentes en muchas aplicaciones distribuidas, pudiéndose construir estas redes añadiendo, a un grafo base, enlaces de largo alcance tomados conforme a una determinada distribución de probabilidad. Los sistemas distribuidos actuales utilizan soluciones ad hoc específicas para calcular los enlaces de largo alcance. En este artículo proponemos un nuevo algoritmo distribuido llamado Selección Sesgada (SS), que utilizando únicamente un servicio de muestreo uniforme (que puede estar implementado mediante un protocolo gossip), es capaz de seleccionar enlaces largos conforme a cualquier distribución de probabilidad. SS es un algoritmo iterativo que dispone de un único parámetro (r) para indicar el número de iteraciones que debe ejecutarse. Se ha probado que la muestra obtenida con el algoritmo SS converge a la distribución objetivo a medida que aumenta el valor de r . También se ha calculado la cota analítica del error relativo máximo, para un determinado valor de r . Aunque en este artículo se propone el algoritmo SS como una herramienta para tomar muestras de nodos en una red, puede emplearse en cualquier contexto en el que sea necesario realizar un muestreo conforme a una determinada distribución de probabilidad, necesitando para funcionar únicamente un servicio de muestreo uniforme. Se han construido redes de Pequeño Mundo, modelo Kleinberg, utilizando SS para escoger los enlaces (vecinos) de largo alcance en estructuras de tipo toro. Hemos observado que con un número reducido de iteraciones (1) SS tiene un comportamiento muy similar a la distribución armónica de Kleinberg y (2) el número medio de saltos, utilizando enrutamiento ávido, no es peor que en una red construida con la distribución de Kleinberg. También se ha observado que antes de obtener la convergencia, el número medio de saltos es menor que en las redes construidas mediante la distribución armónica de Kleinberg (14% mejor en un toro de 1000×1000)

* Este trabajo fue financiado parcialmente por el contrato S2009TIC-1692 de la Comunidad de Madrid y por el contrato TIN2008-06735-C02-01 del Ministerio de Ciencia y Tecnología del Gobierno de España.

1 Introducción

En la actualidad las redes superpuestas están presentes en multitud de aplicaciones y servicios distribuidos. Estas redes, basadas en topologías de redes de pequeño mundo, son una alternativa eficiente y flexible a las redes superpuestas estructuradas. Las redes de pequeño mundo pueden construirse añadiendo enlaces de largo alcance a una red base. Los enlaces de largo alcance se eligen utilizando una determinada distribución de probabilidad [10,7]. Las formas de construir estas redes, mediante la incorporación de enlaces, son muy variadas [5], siendo los protocolos epidémicos una de las formas más sencillas de construcción. Sin embargo, estos protocolos únicamente implementan distribuciones de probabilidad específicas para seleccionar los enlaces de largo alcance. Los protocolos basados en gossip han sido diseñados para realizar la selección de los enlaces de largo alcance, bien mediante la distribución uniforme de probabilidad o bien, a través de una aproximación a la distribución armónica de Kleinberg [9,3,2]. En este artículo proponemos un algoritmo local que, utilizando únicamente un servicio de muestreo uniforme (que podría estar implementado, por ejemplo, utilizando un protocolo gossip), permite seleccionar enlaces de largo alcance conforme a *cualquier* distribución de probabilidad. Este algoritmo es de naturaleza iterativa, y de forma experimental, hemos observado que con un número reducido de iteraciones converge a la distribución de probabilidad deseada.

1.1 Trabajo Relacionado

Las redes de pequeño mundo se han utilizado para intentar explicar las propiedades de las redes sociales, tales como su reducido diámetro y enrutamiento [11]. Una de las líneas de trabajo en redes de pequeño mundo trata de generar redes sintéticas con estas propiedades. Para su construcción, se parte de una red base (representando vecinos geográficamente cercanos) a la que se van añadiendo enlaces de largo alcance (representando vecinos distantes). Este proceso se conoce como *aumento (augmentation)* de la red base o inicial. Watts and Strogatz [14] aumentaron una red con enlaces de largo alcance escogidos de manera uniforme aleatoria. Sin embargo, Kleinberg [10] demostró que un enrutamiento ávido (greedy) obtenía un número de saltos polilogarítmico solo si los enlaces largos se habían seleccionado de acuerdo a una determinada distribución de probabilidad. Las redes construidas de esta forma se conocen comúnmente como redes de Kleinberg. Los resultados de dicho trabajo han dado lugar a posteriores trabajos de construcción de redes de pequeño mundo [1,5,6,7]. La mayoría de los algoritmos propuestos son centralizados. Hasta donde conocemos, el primer algoritmo distribuido para construir redes de pequeño mundo es de Duchon et al. [5]. Bonnet et al. [3] han propuesto dos protocolos, basados en gossip, para seleccionar enlaces de largo alcance, que son versiones de Cyclon [13]. Uno selecciona los enlaces con probabilidad uniforme, mientras que el otro utiliza una aproximación a la distribución de Kleinberg. Este último protocolo ha sido mejorado en [2]. Existen más protocolos, basados en gossip, que seleccionan los nodos con probabilidad uniforme [4,9,13].

1.2 Contribuciones

En este artículo proporcionamos un algoritmo, llamado *Selección Sesgada* (SS), que implementa un servicio de muestreo para cualquier distribución de probabilidad, sobre un conjunto de elementos S . La distribución de probabilidad es proporcional a las probabilidades (representadas como pesos) asignadas a cada elemento del conjunto S . El algoritmo SS es muy sencillo y su ejecución es totalmente local. Necesita únicamente un servicio de muestreo uniforme (posiblemente implementado mediante un protocolo gossip) y los pesos asociados a cada elemento seleccionado por este servicio. El algoritmo tiene un parámetro r que determina el número de veces que se utiliza el servicio de muestreo uniforme (número de rondas) antes de devolver una muestra (de hecho, el número de veces que se utiliza el muestreo uniforme es exactamente $r + 1$). Hemos probado que la distribución de las muestras converge a la distribución deseada a medida que r aumenta. Además, hemos obtenido las cotas analíticas del error relativo máximo para un valor dado del parámetro r . Aunque en este artículo hemos propuesto el algoritmo SS como una herramienta para muestrear nodos de una red, puede utilizarse en cualquier contexto que requiera un muestreo conforme a una determinada distribución de probabilidad, siendo necesario únicamente un servicio de muestreo aleatorio.

Para evaluar el rendimiento del algoritmo, hemos seleccionado enlaces de largo alcance en una red de tipo toro, con el objetivo de construir redes de pequeño mundo similares a las propuestas por Kleinberg [10]. En esta red, cada nodo i del toro escoge otro nodo j como enlace de largo alcance con una probabilidad proporcional a $1/d(i, j)^2$, donde $d(i, j)$ es la distancia Euclídea⁵ del nodo i al j . Por simplicidad, la hemos llamado *distribución de Kleinberg*. Obviamente, para escoger un enlace de largo alcance de un nodo i se necesita conocer la distancia desde i a todos los demás nodos, y calcular las probabilidades asociadas, lo que requiere $\Omega(n)$ operaciones en una red de n nodos. Hacer esto para todos los nodos de la red requerirá $\Omega(n^2)$ operaciones. Por el contrario, SS no necesita conocer todos los nodos de la red, requiriendo únicamente la distancia desde el nodo i a los nodos devueltos por el servicio de muestreo uniforme. Si SS utiliza r rondas, el número total de operaciones para cada nodo i es $O(r)$, y $O(rn)$ operaciones para toda la red. Hemos observado experimentalmente que el valor requerido de r es más pequeño que n .

Comparando las muestras obtenidas en simulación por SS y las obtenidas por el simulador de la distribución de Kleinberg, observamos que con un número pequeño de rondas (10 en un toro de 100×100), ambas simulaciones tienen un error relativo medio casi idéntico y un error relativo máximo muy similar. Después, construimos redes añadiendo a cada nodo un enlace de largo alcance. Evaluamos el rendimiento del enrutamiento ávido en redes en las que el enlace de largo alcance se había escogido mediante el algoritmo SS (redes SS) utilizando diferentes números de rondas. Comparamos los resultados con redes construidas con la distribución de Kleinberg (redes K) y con la distribución uniforme (redes U). Observamos que el número medio de saltos, utilizando enrutamiento ávido, en las redes SS convergía a los valores obtenidos en las redes K, a medida que el número de rondas aumentaba. Es más, con un número reducido de rondas, el número medio de saltos en la red SS era sensiblemente menor que

⁵ Observe que Kleinberg, en cambio, utilizó distancia Manhattan en un malla.

el de la red U (especialmente para redes grandes). Hemos observado con sorpresa que, antes de converger al rendimiento de la red K , el número medio de saltos en la red SS es significativamente menor (hasta un 14% menor en una red de 1000×1000). De hecho, el mejor rendimiento de las redes SS se obtiene con un número reducido de rondas. Este comportamiento lo estudiaremos en el futuro. Finalmente, hemos experimentado añadiendo enlaces de largo alcance en toros incompletos. Estos toros se han construido a partir de un toro completo al que se le han eliminado nodos con una probabilidad fija. Para estos experimentos se han utilizado dos probabilidades de presencia, 0.8 y 0.3, obteniendo dos tipos de red, una densa y otra dispersa. Los resultados obtenidos para estas redes son coherentes con los obtenidos en los toros completos.

1.3 Estructura del Resto del Artículo

En la Sección 2 introducimos los conceptos y la notación que utilizaremos en el resto del artículo, junto con la descripción del entorno experimental utilizado. En la Sección 3 presentamos el algoritmo SS , probamos su corrección, y obtenemos una cota de la convergencia. Finalmente, en <http://www.networks.imdea.org/Portals/8/Downloads/Publications/Biased-Selection-2010-EN.pdf> pueden encontrarse los resultados experimentales del trabajo y las conclusiones.

2 Definiciones y Configuración del Entorno de Simulación

2.1 Definiciones

Aunque en este artículo hemos propuesto el algoritmo SS como una herramienta para muestrear nodos de una red, lo presentamos de una forma más general para enfatizar el hecho de que puede ser utilizado en otros contextos. Sea S un conjunto de n elementos tal que la probabilidad de escoger un elemento i está asociada a un peso $w(i) > 0$. El problema a resolver es muestrear el conjunto S con una distribución de probabilidad p tal que la probabilidad de escoger i es proporcional a $w(i)$. Llamaremos $\eta = \sum_{j \in S} w(j)$, siendo la probabilidad de muestreo de $i \in S$ igual a $p(i) = w(i)/\eta$. Los desafíos en el muestreo del conjunto S son los siguientes:

1. Supondremos que no está disponible en su totalidad.
2. Los pesos de los elementos únicamente pueden consultarse para elementos ya conocidos.

Estas restricciones impiden, por ejemplo, el cálculo del valor η . Sin embargo, para poder resolver el problema, supondremos la existencia de una primitiva de muestreo $USel_S$ que devuelve un elemento i de S escogido con probabilidad uniforme. Una vez obtenido un elemento $i \in S$, mediante $USel_S$, también podrá obtenerse su peso $w(i)$.

2.2 Configuración del Entorno de Simulación

En los experimentos realizados para este artículo, hemos utilizado una topología toro de 2 dimensiones. Para posicionar cada nodo se ha utilizado un par de valores (x, y)

en un espacio de 2 dimensiones. Las coordenadas de los nodos van desde 0 a $m - 1$, teniendo, por lo tanto, la red un total de m^2 nodos. En esta topología, la distancia entre dos nodos con posiciones (x_1, y_1) y (x_2, y_2) respectivamente, es la distancia Euclídea en el toro, calculada como:

$$d_e = \sqrt{(\min(|x_1 - x_2|, m - |x_1 - x_2|))^2 + (\min(|y_1 - y_2|, m - |y_1 - y_2|))^2}$$

Para probar el algoritmo SS se han diseñado dos tipos de experimentos. El primer experimento (sección 4) muestra el error relativo medio y máximo del algoritmo SS con respecto a la distribución de probabilidad de Kleinberg. Estos valores también se han comparado con el error relativo obtenido en una simulación de la distribución de Kleinberg. El segundo grupo de experimentos (sección 4) compara el número medio de saltos, utilizando enrutamiento ávido, en dos redes, una construida con el algoritmo SS y otra con el simulador de Kleinberg. Este experimento se ejecutó en dos escenarios diferentes. En el primero se utilizó un toro completo con m^2 nodos, en el que cada nodo tenía cuatro vecinos cercanos y un vecino lejano. Para el segundo, se usó un toro incompleto, en el que se habían eliminado nodos de forma uniformemente aleatoria. En esta red, el número medio de nodos presentes es m^2q , siendo q la probabilidad de presencia de un nodo. Observe que q es igual para todos los nodos, y la presencia de cada nodo es independiente del resto. Para permitir en este escenario el enrutamiento ávido, cada nodo i tiene siete enlaces, seis a nodos cercanos y uno a un nodo lejano. Los seis enlaces cercanos están distribuidos en cada uno de los seis ángulos de 60° de una circunferencia centrada en el nodo i [15].

Cuando se utiliza el algoritmo SS, el número de rondas r determinará la exactitud de los valores obtenidos. Los experimentos se ejecutaron varias veces. Comenzamos con un valor de $r = 0$ rondas (distribución uniforme) y gradualmente incrementamos r hasta que el algoritmo SS convergió a la distribución de Kleinberg. Cada experimento se ejecutó 10 veces con diferentes semillas, aunque los datos mostrados en los resultados corresponden a la media de estas ejecuciones.

3 Selección Sesgada

3.1 Algoritmo de Selección Sesgada

A continuación presentamos el algoritmo para muestrear elementos del conjunto S , de acuerdo a la distribución de probabilidad deseada, tal y como se definió en la Sección 2. El código del algoritmo *Selección Sesgada* aparece en la Figura 1. Dispone de un único parámetro r que indica el número de rondas a ejecutar antes de devolver una muestra. Más adelante mostraremos, que a medida que aumenta el número de rondas, el valor devuelto por $SSel_S(r)$ sea acerca más a la distribución deseada.

3.2 Corrección

Primero mostraremos que, a medida que el valor de r tiende a infinito, la distribución de los valores de salida de $SSel_S(r)$ converge a la distribución de probabilidad deseada

```

1 function  $SSel_S(r)$ 
2  $x \leftarrow USel_S$ 
3 for  $i \leftarrow 1$  to  $r$  do
4    $y \leftarrow USel_S$ 
5   set  $x \leftarrow y$  con probabilidad  $\frac{w(y)}{w(x)+w(y)}$ 
6 end for
7 return  $x$ 

```

Fig. 1. Algoritmo de Selección Sesgada para el conjunto S .

p . Sea x_i el valor almacenado en la variable x después de i iteraciones del bucle, siendo x_0 el valor inicial asignado a x en la Línea 2. Consideremos la ejecución de $SSel_S(\infty)$. La secuencia de infinitos valores x_0, x_1, \dots puede verse como una cadena de Markov⁶ M sobre el espacio finito de estados S . Primero mostraremos que M tiene una única distribución estacionaria π tal que $\pi = A\pi$, donde A es la matriz de transición de M . Por último, probaremos que π se ajusta a la distribución de probabilidad p . La matriz de transición $A = [a_{ij}]$ de la cadena de Markov M puede obtenerse del algoritmo de la Figura 1 de la forma siguiente. Para cada $i, j \in S$ y $i \neq j$,

$$a_{ij} = \Pr[x_{t+1} = j | x_t = i] = \frac{1}{n} \frac{w(j)}{w(j) + w(i)} = \frac{1}{n} \frac{p(j)}{p(j) + p(i)}.$$

Además, $a_{ii} = 1 - \sum_{j \neq i} a_{ij}$. Observe que para todo $j \neq i$, $a_{ij} < 1/n$ (siendo $n = |S|$), y por lo tanto $a_{ii} > 0$. Ahora mostraremos que M es *ergódica*. Sea $a_{ij}^{(s)}$ la probabilidad de alcanzar el estado j en s pasos comenzando desde el estado i .

Una cadena de Markov es *ergódica*, si es *finita*, *irreducible* ($\forall i, j \in S, \exists s : a_{ij}^{(s)} > 0$), y *aperiódica* ($\forall i, j \in S, \gcd\{s : a_{ij}^{(s)} > 0\} = 1$) [12]. La ergodicidad implica que la distribución estacionaria es única.

Lema 1 *La cadena de Markov M es ergódica, y por lo tanto tiene una única distribución π estacionaria que satisface $\pi = \pi A$.*

Demostración. Al cumplirse que $a_{ij} > 0$ para todo $i, j \in S$, la cadena de Markov M es irreducible. Además, al ser $a_{ii} > 0$ para todo $i \in S$, M es aperiódica. Por lo tanto, al ser ergódica, y tener una única distribución estacionaria π se cumple que $\pi = \pi A$ [12]. Con esto probamos que la distribución estacionaria es la distribución deseada p .

Teorema 1 *La salida de $SSel_S(r)$ converge a la distribución de probabilidad p a medida que r tiende a infinito.*

Demostración. De acuerdo al anterior lema, la distribución de valores x_r devueltos por el algoritmo converge a la distribución estacionaria π de la cadena de Markov M a medida que r tiende al infinito. Todo lo que se necesita probar es que $p = \pi$, es decir,

⁶ Se supone que el lector tiene conocimientos sobre cadenas de Markov.

que $p = pA$. Para cualquier $i \in S$, necesitamos probar que $p(i) = \sum_{j \in S} p(j)a_{ji}$. Sustituyendo

$$\begin{aligned}
\frac{w(i)}{\eta} &= \sum_{j \in S} \frac{w(j)}{\eta} a_{ji} \\
&= \sum_{j \neq i} \frac{w(j)}{\eta} \frac{1}{n} \frac{w(i)}{w(j) + w(i)} + \frac{w(i)}{\eta} (1 - \sum_{j \neq i} a_{ij}) \\
&= \sum_{j \neq i} \frac{w(i)}{\eta} \frac{1}{n} \frac{w(j)}{w(j) + w(i)} + \frac{w(i)}{\eta} (1 - \sum_{j \neq i} a_{ij}) \\
&= \frac{w(i)}{\eta} \left(\sum_{j \neq i} a_{ij} + 1 - \sum_{j \neq i} a_{ij} \right) \\
&= \frac{w(i)}{\eta}
\end{aligned}$$

Ya que se cumple para todo $i \in S$, la demostración está completada.

3.3 Velocidad de Convergencia

Nos centraremos ahora en el número de rondas r que son necesarias para que la distribución de los valores devueltos por $SSel_S(r)$ sean *casi* los mismos que los de la distribución de probabilidad p . Para medir la distancia entre ambas distribuciones, utilizaremos la distancia puntual relativa (*relative pointwise distance*) como está definida en [12]. Este parámetro mide el mayor error relativo entre ambas distribuciones, para cualquier posible valor inicial o final. Observe que $a_{ij}^{(r)}$ es la probabilidad de que $SSel_S(r)$ devuelva el valor j si el valor inicial de x es i , es decir, $a_{ij}^{(r)} = \Pr[x_r = j | x_0 = i]$. Por lo tanto, el *error relativo máximo* está definido como:

$$\Delta(r) = \max_{i,j \in S} \frac{|a_{ij}^{(r)} - p(j)|}{p(j)}.$$

Para acotar la expresión $\Delta(r)$ probaremos primero que la cadena de Markov M es *reversible*, si cumple que $a_{ij}p(i) = a_{ji}p(j)$ [12].

Lema 2 *La cadena de Markov M es reversible.*

Demostración. Sustituyendo en $a_{ij}p(i) = a_{ji}p(j)$, obtenemos

$$a_{ij}p(i) = \frac{1}{n} \frac{w(j)}{w(j) + w(i)} \frac{w(i)}{\eta} = \frac{1}{n} \frac{w(i)}{w(j) + w(i)} \frac{w(j)}{\eta} = a_{ji}p(j)$$

Para acotar $\Delta(r)$ son útiles los lemas 1 y 2 por el siguiente resultado, derivado de la Proposición 3.1 en [12].

Lema 3 ([12]) Sea A la matriz de transición de la cadena de Markov ergódica y reversible, p su distribución estacionaria, y $1 = \lambda_0 > \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{n-1}$ sus autovalores. Entonces, para todo $r \geq 1$ el error relativo máximo satisface

$$\Delta(r) \leq \frac{\lambda^r}{p_{\min}},$$

donde $\lambda = \max_{k \geq 1} |\lambda_k|$ y $p_{\min} = \min_{i \in S} p(i)$.

Claramente, $\lambda = \max(\lambda_1, |\lambda_{n-1}|)$, donde $\lambda_1 < |\lambda_{n-1}|$ si y solo si $\lambda_{n-1} < 0$. Definiendo $a_{\min} = \min_{i \in S} a_{ii}$, por el teorema del Círculo de Gershgorin [8], tenemos que $\lambda_{n-1} \geq 2a_{\min} - 1$. Por lo tanto, $\lambda \leq \max\{\lambda_1, 1 - 2a_{\min}\}$. Para acotar λ_1 usaremos la conductancia de M .

Definición 1 Considerando la cadena de Markov M , para cualquier conjunto $B \subset S$, representado como $C(B) = \sum_{i \in B} p(i)$ y $F(B) = \sum_{i \in B, j \notin B} a_{ij}p(i)$, la conductancia de M se define como

$$\Phi = \min_{\emptyset \subset B \subset S: C(B) \leq 1/2} \frac{F(B)}{C(B)}.$$

El Lema 3.3 en [12] muestra que $\lambda_1 \leq 1 - \frac{\Phi^2}{2}$. Por lo tanto, podemos acotar λ_1 de la forma siguiente:

Lema 4 Los autovalores λ_1 de la cadena de Markov M ergódica y reversible satisfacen

$$\lambda_1 \leq 1 - \frac{1}{2} (\max\{(1 - 1/n)p_{\min}, 1/(4np_{\max})\})^2,$$

donde $p_{\min} = \min_{i \in S} p(i)$ y $p_{\max} = \max_{i \in S} p(i)$.

Demostración. Si consideramos cualquier conjunto B tal que $\emptyset \subset B \subset S$ y $C(B) \leq 1/2$, podemos representar la conductancia como $\Phi(B) = \frac{F(B)}{C(B)}$, obteniendo dos cotas inferiores para $\Phi(B)$.

Ya que $C(B) \leq 1/2$, entonces $\Phi(B) \geq 2F(B)$. Para acotar $F(B)$, observamos que $\frac{p(i)p(j)}{p(j)+p(i)}$ decrece con los valores de $p(i)$ y $p(j)$, lo que implica que $\frac{p(i)p(j)}{p(j)+p(i)} \geq \frac{p_{\min}^2}{2p_{\min}} = \frac{p_{\min}}{2}$. Por otro lado, $|B| \cdot |S \setminus B| \geq n - 1$. Por lo tanto,

$$F(B) = \sum_{i \in B, j \notin B} a_{ij}p(i) = \frac{1}{n} \sum_{i \in B, j \notin B} \frac{p(i)p(j)}{p(j) + p(i)} \geq \frac{1}{n} \sum_{i \in B, j \notin B} \frac{p_{\min}}{2} \geq \frac{n-1}{n} \frac{p_{\min}}{2}.$$

lo que implica que $\Phi(B) \geq (1 - 1/n)p_{\min}$.

En cuanto a la segunda cota, observamos que

$$F(B) = \frac{1}{n} \sum_{i \in B, j \notin B} \frac{p(i)p(j)}{p(j) + p(i)} \geq \frac{1}{2np_{\max}} \sum_{i \in B, j \notin B} p(i)p(j) = \frac{(\sum_{i \in B} p(i)) (\sum_{j \notin B} p(j))}{2np_{\max}}.$$

y al ser $C(B) = \sum_{i \in B} p(i)$,

$$\Phi(B) = \frac{F(B)}{C(B)} \geq \frac{\sum_{j \notin B} p(j)}{2np_{\max}} \geq \frac{1}{4np_{\max}},$$

donde la segunda desigualdad se obtiene de $\sum_{j \notin B} p(j) = 1 - C(B)$ y $C(B) \leq 1/2$.

Ya que ambas cotas se mantienen para cualquier B , obtenemos que $\Phi \geq \max\{(1 - 1/n)p_{\min}, 1/(4np_{\max})\}$. Combinando esta cota con el Lema 3.3 de [12], el Lema se cumple.

A partir de los resultados anteriores podemos acotar la distancia puntual relativa (*relative pointwise distance*) de la forma siguiente:

Teorema 2 *El error relativo máximo de la cadena de Markov M cumple*

$$\Delta(r) \leq \left(1 - \min\left\{ \frac{1}{2} (\max\{(1 - 1/n)p_{\min}, 1/(4np_{\max})\})^2, 2a_{\min} \right\} \right)^r / p_{\min},$$

donde $p_{\min} = \min_{i \in S} p(i)$, $p_{\max} = \max_{i \in S} p(i)$, y $a_{\min} = \min_{i \in S} a_{ii}$.

4 Resultados

En <http://www.networks.imdea.org/Portals/8/Downloads/Publications/Biased-Selection-2010-EN.pdf> están disponibles los resultados.

Agradecimientos

Los autores quieren dar las gracias a Juan A. Carrasco por sus valiosos comentarios sobre las cadenas de Markov y por indicarnos la existencia del Teorema del Círculo de Gershgorin.

References

1. L. Barrière, P. Fraigniaud, E. Kranakis, and D. Krizanc. Efficient routing in networks with long range contacts. In J. L. Welch, editor, *Proc. DISC*, volume 2180 of *Lecture Notes in Computer Science*, pages 270–284. Springer, 2001.
2. Marin Bertier, François Bonnet, Anne-Marie Kermarrec, Vincent Leroy, Sathya Peri, and Michel Raynal. D2ht: The best of both worlds, integrating rps and dht. In *EDCC*, pages 135–144. IEEE Computer Society, 2010.
3. F. Bonnet, A.-M. Kermarrec, and M. Raynal. Small-world networks: From theoretical bounds to practical systems. In *Proc. OPODIS*, pages 372–385, 2007.
4. F. Bonnet, F. Tronel, and S. Voulgaris. Brief announcement: Performance analysis of cyclon, an inexpensive membership management for unstructured P2P overlays. In S. Dolev, editor, *Proc. DISC*, volume 4167 of *Lecture Notes in Computer Science*, pages 560–562. Springer, 2006.

5. Philippe Duchon, Nicolas Hanusse, Emmanuelle Lebhar, and Nicolas Schabanel. Towards small world emergence. In Phillip B. Gibbons and Uzi Vishkin, editors, *SPAA*, pages 225–232. ACM, 2006.
6. P. Fraigniaud, C. Gavoille, and C. Paul. Eclecticism shrinks even small worlds. *Distributed Computing*, 18(4):279–291, 2006.
7. Pierre Fraigniaud and George Giakkoupis. On the searchability of small-world networks with arbitrary underlying structure. In Leonard J. Schulman, editor, *STOC*, pages 389–398. ACM, 2010.
8. S. Gerschgorin. Über die abgrenzung der eigenwerte einer matrix. *Izv. Akad. Nauk. USSR Otd. Fiz.-Mat. Nauk*, 7:749–754, 1931.
9. M. Jelasity, S. Voulgaris, R. Guerraoui, A.-M. Kermarrec, and M. van Steen. Gossip-based peer sampling. *ACM Trans. Comput. Syst.*, 25(3), 2007.
10. J. M. Kleinberg. Navigation in a small world. *Nature*, 406(6798), August 2000.
11. Stanley Milgram. The small world problem. *Psychology Today*, 2:60–67, 1967.
12. Alistair Sinclair and Mark Jerrum. Approximate counting, uniform generation and rapidly mixing markov chains. *Inf. Comput.*, 82(1):93–133, 1989.
13. S. Voulgaris, D. Gavidia, and M. van Steen. Cyclon: Inexpensive membership management for unstructured P2P overlays. *J. Network Syst. Manage.*, 13(2), 2005.
14. D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 1998.
15. A.C.C. Yao. On constructing minimum spanning trees in k-dimensional space and related problems. *SIAM Journal of Computing*, 1982.