Selección de Centros para un Índice Métrico Basado en Particiones Compactas *

Cristian Mendoza Alric
Departamento de Informática
Universidad Nacional de San Luis
Argentina
calric@unsl.edu.ar

Norma Edith Herrera
Departamento de Informática
Universidad Nacional de San Luis
Argentina
nherrera@unsl.edu.ar

Abstract

The metric spaces model allows to formalize the similarity search concept in nontraditional databases. The goal is to build an index designed to save distance computations when answering similarity queries later.

A large class of algorithms to build the index is based on dividing the space in zones as compact as possible. Each zone stores a representative point, called *center*, and a little extra data that allows discarding the entire zone at query time without measuring the actual distance among the elements of the zone and the query object. The way in which the centers are selected affects the performance of the algorithm.

In this paper, we introduce two new centers selection techniques for the Geometric Near-neighbor Access Tree (GNAT), an index based on compact partitions. We show experimentally that these techniques achieve a good performance.
Keywords: Databases, Metric Spaces, Index, Centers Selection.

Resumen

El modelo de Espacios Métricos permite formalizar el concepto de búsqueda por similitud en bases de datos no tradicionales. El objetivo es construir *índices* que permitan reducir el tiempo necesario para resolver una búsqueda por similitud.

Una amplia clase de algoritmos construyen el índice dividiendo el espacio en zonas tan compactas como sea posible. Por cada zona se almacena un elemento representativo, llamado *centro* , e información adicional que permiten descartar la zona completa durante una búsqueda, sin tener que calcular la distancia entre los elementos de la zona y el objeto de búsqueda. La manera en que se seleccionan los centros afecta el desempeño del índice.

En este artículo presentamos dos nuevas políticas para la selección de centros en el Geometric Near-neighbor Access Tree (GNAT), un índice basado en particiones compactas. Experimentalmente mostramos que estas políticas logran un buen desempeño.
Palabras Claves: Bases de Datos, Espacios Métricos, Índice, Selección de centros.

*Este trabajo ha sido parcialmente subvencionado por el proyecto 22/F314, de la Universidad Nacional de San Luis

1. INTRODUCCIÓN

El concepto de *búsquedas por similitud* o *por proximidad*, es decir buscar elementos de una base de datos que sean similares o cercanos a uno dado, aparece en diversas áreas de computación, tales como reconocimiento de voz, reconocimiento de imágenes, compresión de texto, biología computacional, inteligencia artificial, minería de datos, entre otras.

En [4] se muestra que el problema se puede expresar como sigue: dado un conjunto de objetos $\mathcal{X}$ y una función de distancia d definida entre ellos que mide cuan diferentes son, el objetivo es recuperar todos aquellos elementos que sean similares a uno dado. Esta función d cumple con las propiedades características de una función de distancia: *positividad* ($d(x, y) \geq 0$), *simetría* ($d(x, y) = d(y, x)$) y *desigualdad triangular* ($d(x, y) \leq d(x, z) + d(z, y)$).

El par $(\mathcal{X}, d)$ se denomina ***espacio métrico***. La base de datos será un subconjunto finito $\mathcal{U} \subseteq \mathcal{X}$. En este nuevo modelo de bases de datos, una de las consultas típicas que implica recuperar objetos similares es la *búsqueda por rango*, que denotaremos con $(q, r)_d$. Dado un elemento $q \in \mathcal{X}$, al que llamaremos *query*, y un radio de tolerancia r, una búsqueda por rango consiste en recuperar los objetos de la base de datos cuya distancia a q no sea mayor que r, es decir, $(q, r)_d = \{u \in \mathcal{U} : d(q, u) \leq r\}$.

El tiempo total de resolución de una búsqueda contiene tres términos, a saber: *T= #evaluaciones de* $d\times$ *complejidad(d) + tiempo extra de CPU + tiempo de I/O* En muchas aplicaciones la evaluación de la función d es tan costosa que las demás componentes de la fórmula anterior pueden ser despreciadas. Éste es el modelo usado en este trabajo; por consiguiente, nuestra medida de complejidad será la cantidad de evaluaciones de la función de distancia d.

Una forma trivial de resolver una búsqueda por rango es examinando exhaustivamente la base de datos. Para evitar esta situación, se preprocesa la base de datos por medio de un *algoritmo de indexación* con el objetivo de construir una *estructura de datos o índice*, diseñada para ahorrar cálculos en el momento de resolver una búsqueda.

En [4] se presenta un desarrollo unificador de las soluciones existentes en la temática. En dicho trabajo se muestra que todos los enfoques para la construcción de índices en espacios métricos consisten en particionar el espacio en clases de equivalencia e indexar las clases de equivalencia. Luego, durante la búsqueda, por medio del índice se descartan algunas clases y se buscan exhaustivamente en las restantes.

La diferencia entre los distintos algoritmos radica en cómo construyen esta relación de equivalencia. Básicamente se pueden distinguir dos grupos: *algoritmos basados en pivotes* y *algoritmos basados en particiones compactas* .

En el caso de los algoritmos basados en pivotes, la relación de equivalencia se define tomando en cuenta la distancia de los elementos de la base a un conjunto preseleccionado de elementos denominados *pivotes*; en este sentido, dos elementos son considerados equivalentes si están exactamente a la misma de distancia de todos los pivotes.

En el caso de los algoritmos basados en particiones compactas, la relación de equivalencia se define teniendo en cuenta la cercanía de los elementos a un conjunto preseleccionado de elementos denominados *centros*; en este caso dos elementos son equivalentes si tienen al mismo centro c como su centro más cercano. La mayoría de estos algoritmos elijen los centros en forma aleatoria. Sin embargo, el conjunto de centros seleccionados afectan la performance del índice.

En este trabajo abordamos el estudio de algoritmos de indexación basados en particiones compactas. Específicamente hemos estudiado un índice de esta categoría el ***Geometric Near-neighbor Access Tree (GNAT)*** con el objetivo de diseñar políticas para la selección de centros que logren mejorar el desempeño del índice durante la resolución de una búsqueda por rango. Las políticas diseñadas se evaluaron comparándolas experimentalmente con la política trivial, selección random, lo que permi-

tió establecer la competitividad de una de ellas.

Este artículo está organizado de la siguiente manera. Comenzamos en la sección 2 dando una breve explicación de técnicas de indexación en espacios métricos. En la sección 3 explicamos en detalle el GNAT, índice en el que está basado este trabajo. Las secciones 4 y 5 están dedicadas a las políticas de selección de centros que hemos diseñado; damos una explicación y la evaluación experimental de las mismas. Finalizamos en la sección 6 dando las conclusiones y el trabajo futuro.

2. TÉCNICAS DE INDEXACIÓN PARA ESPACIOS MÉTRICOS

Tal como lo mencionáramos en la introducción, los algoritmos de indexación para espacios métricos pueden clasificarse en dos grandes categorías: *algoritmos basados en pivotes* y *algoritmos basados en particiones compactas* . A continuación explicamos brevemente cada una de ellas.

Algoritmos basados en pivotes:

La idea subyacente de los algoritmos de indexación basados en pivotes es la siguiente. Se seleccionan k pivotes $\{p_1, p_2, \ldots, p_k\}$, y se le asigna a cada elemento a el vector o firma $\delta(a) = (d(a, p_1), d(a, p_2), \ldots, d(a, p_k))$.

Ante una búsqueda $(q, r)_d$, se usa la desigualdad triangular junto con los pivotes para filtrar elementos de la base de datos sin medir su distancia a la query q. Para ello se computa la distancia de q a cada uno de los pivotes p_i, y luego se descartan todos aquellos elementos a, tales que para algún pivote p_i se cumple que $\mid d(q, p_i) - d(a, p_i) \mid > r$. Los elementos no descartados pasan a formar parte de un conjunto de elementos que se comparan directamente con q para determinar si forman o no parte de la respuesta.

Algoritmos basados en particiones compactas:

En este caso la idea es dividir el espacio en zonas tan compactas como sea posible. Para ello seleccionan un conjunto de *centros* $\{c_1, c_2, \ldots, c_k\}$ y dividen el espacio asociando a cada centro c_i la clase o parte $[c_i]$ formada por el conjunto de puntos que tiene a c_i como su centro más cercano.

Existen muchos criterios posibles para descartar zonas durante una búsqueda. Los dos más populares son:

a. Criterio del hiperplano: es el más básico y el que mejor expresa la idea de partición compacta. Básicamente, si c es el centro de la clase $[q]$ (es decir, el centro más cercano a q) entonces la bola con centro q no intersecta $[c_i]$ si $d(q, c) + r < d(q, c_i) - r$. Es decir, si la bola asociada a q no intersecta el hiperplano que divide su centro más cercano c y el centro c_i, entonces cae fuera de la clase de c_i.

b. Criterio del radio de cobertura: en este caso se trata de limitar la clase $[c_i]$ considerando la bola centrada en c_i que contiene todos los elementos de $\mathcal{U}$ que caen en la clase. Definimos el radio de cobertura de c en el espacio $\mathcal{U}$ como $cr(c) = max_{\ u \in [c] \cap \mathcal{U}}\ d(c, u)$. Luego, podemos descartar $[c_i]$ si $d(q, c_i) - r > cr(c_i)$.

Uno de los principales obstáculos en el diseño de buenas técnicas de indexación es lo que se conoce con el nombre de *maldición de la dimensionalidad*. El concepto de dimensionalidad está relacionado a la dificultad o facilidad de buscar en un determinado espacio métrico. La dimensión intrínseca de un espacio métrico se define en [4] como $\rho = \frac{\mu^2}{2\sigma^2}$, siendo μ y σ^2 la media y la varianza respectivamente de su histograma de distancias. Es decir que, a medida que la dimensionalidad intrínseca crece,

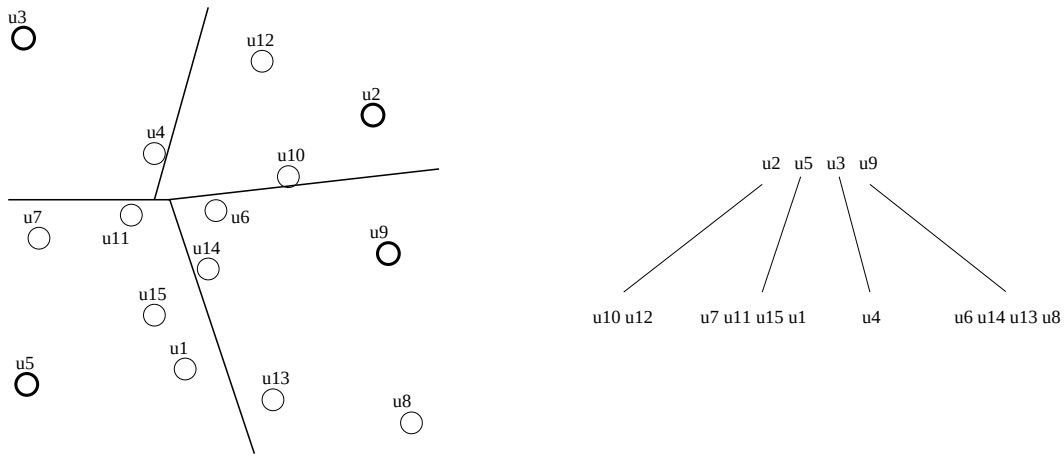

Figura 1: *Primer nivel de un GNAT de aridad 4, usando u_2, u_3, u_5 y u_9 como centros.*

la media aumenta y su varianza se reduce. Esto significa que el histograma de distancia se concentra más alrededor de su media, lo que influye negativamente en los algoritmos de indexación.

3. GEOMETRIC NEAR-NEIGHBOR ACCESS TREE

Este índice, presentado por Sergey Brin en el año 1995 [3], es una extensión del *Generalized Hyperplane Tree (GHT)* [5]. El objetivo que se persigue, es que la estructura actúe como un modelo geométricamente jerárquico de los datos. Más específicamente, a partir del nodo raíz se obtiene una una idea de los datos como espacio métrico, y a medida que avanzamos en la jerarquía del árbol se logra una idea más exacta de la geometría de los mismos. Para lograr esto se construye una jerarquía basada en *Diagramas de Voronoi* [1].

La construcción de un GNAT de aridad m procede de la siguiente manera: en el primer nivel se seleccionan m centros $c_1, c_2, \cdots, c_m$ de $\mathcal{U}$, que se almacenan en el nodo raíz. A cada centro c_i se le asocia el conjunto $\mathcal{U}_{c_i}$ formado por aquellos objetos que están más cerca de c_i que de cualquier otro centro c_j; en símbolos:

$$\mathcal{U}_{c_i} = \{x \in \mathcal{U} / d(c_i, x) < d(c_j, x),\ \forall j = 1 \ldots m,\ j \neq i\}$$

Para cada $\mathcal{U}_{c_i}$, si su cardinalidad es mayor que m se construye recursivamente un GNAT, caso contrario se construye con esos elementos un nodo terminal (ver figura 1).

En cada nodo del GNAT se almacena además una tabla, a la que denotaremos con ρ, de tamaño $O(m^2)$. Esta tabla mantiene información sobre las distancias mínimas y máximas, desde el centro c_i a los conjuntos $\mathcal{U}_{c_j}$:

$$\rho_{i,j} = [min_{x \in \mathcal{U}_{c_j}} d(c_i, x), max_{x \in \mathcal{U}_{c_j}} d(c_i, x)] \text{ , con } i, j = 1, \ldots, m$$

Esta información, junto con la desigualdad triangular, se usa en el momento de la búsqueda para descartar subárboles. Para una búsqueda por rango $(q, r)_d$, se compara q con algún centro c_i, y se descartan todos aquellos centros c_j (y sus correspondientes $\mathcal{U}_{c_j}$) tales que:

$$[d(q, c_i) - r, d(q, c_i) + r] \cap \rho_{i,j} = \emptyset$$

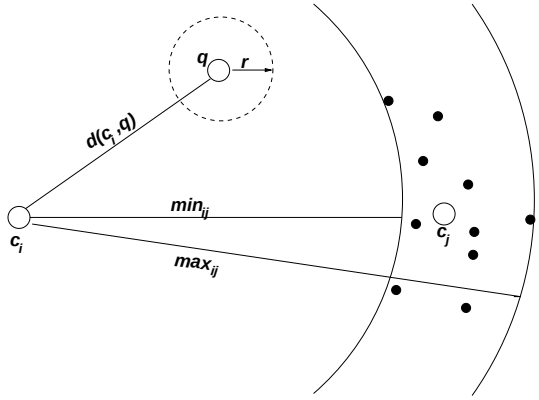

Figura 2: *Eliminación de subárboles usando ρ_{ij}. En este caso podemos eliminar $\mathcal{U}_{c_j}$ dado que $d(q, c_i) + r < min_{x \in \mathcal{U}_{c_j}} d(c_i, x)$*

La razón por la que esto puede hacerse es muy sencilla. Sea $y \in \mathcal{U}_{c_j}$, si $d(c_i, y) < d(c_i, q) - r$, luego por la desigualdad triangular se sigue que $d(c_i, y) < d(c_i, y) + d(y, q) - r$, de donde se deduce que $d(y, q) > r$. Análogamente, si $d(c_i, y) > d(c_i, q) + r$, por la desigualdad triangular se sigue que $d(c_i, q) + d(q, y) > d(c_i, q) + r$, de donde deducimos que $d(y, q) > r$ (ver figura 2).

Este proceso se repite hasta que ningún centro pueda descartarse. La búsqueda continúa luego recursivamente en aquellos subárboles no eliminados. Durante este proceso se agregan al resultado todos aquellos centros c_i tales que $d(c_i, q) \leq r$.

La aridad elegida para la construcción del GNAT influye notablemente en la performance del mismo. Para algunos espacios métricos, una aridad alta puede ser una buena elección, mientras que para un espacio métrico diferente una aridad pequeña puede producir mejores resultados [2].

De igual manera, la política usada para seleccionar los centros durante la construcción del GNAT afectan la performance del mismo en el momento de resolver una búsqueda. Por esta razón en este trabajo nos hemos centrado en el estudio y diseño de técnicas de selección de centros que logren mejorar el desempeño del índice durante una búsqueda.

4. POLÍTICAS PARA LA SELECCIÓN DE CENTROS

Descubrir la estructura subyacente del conjunto de datos es sumamente útil en el diseño de algoritmos de indexación. En particular, saber cómo se agrupan los elementos del espacio métrico nos sirve para identificar la zona de búsqueda más difícil.

Una forma de visualizar la distribución de los datos del espacio métrico, es por medio de los histogramas de distancias. Dado un espacio métrico $(\mathcal{X}, d)$ y un elemento $p \in \mathcal{X}$, *el histograma local* respecto del punto de referencia p es la distribución de distancias de p a los elementos $x \in \mathcal{X}$ (ver figura 3).

En [2] los autores definen y caracterizan el *núcleo duro* y el *núcleo blando* de un espacio métrico. El núcleo duro está formado por aquellos elementos que se encuentran en la zona de mayor concentración de objetos; el núcleo blando está conformado por los restantes elementos del espacio. En dicho trabajo también se da el algoritmo que permite encontrar el núcleo duro de un espacio; dicho algoritmo consiste básicamente en intersectar la zona central de varios histogramas locales.

Tomando como base estas ideas, diseñamos dos políticas para la selección de centros. Una de ellas consiste en tomar los centros del núcleo blando y otra es tomar los centros del núcleo duro. En ambos casos, no calculamos los núcleos reales del espacio haciendo intersecciones de varios histogramas de

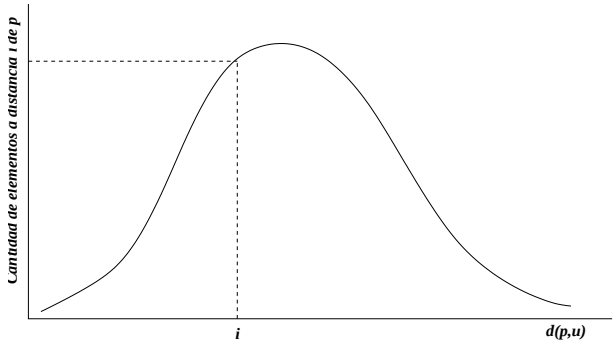

Figura 3: *Ejemplo de un histograma local para un elemento p.*

distancia, sino que trabajamos en cada paso de la selección sólo con el histograma del último centro elegido. A continuación explicamos detalladamente estas ideas:

Elemento más cercano :
Esta política procede de la siguiente manera. El primer centro c_1 se elije aleatoriamente. El segundo centro se elije de la zona que c_1 consideraría como núcleo blando, es decir, de los extremos del histograma, si el mismo tiene forma de campana de Gauss. Para ello se calcula el histograma local de c_1 y se toma como segundo centro al elemento más cercano a c_1. En general el centro c_{i+1} será el elemento más cercano a c_i.

Zona de mayor concentración :
Nuevamente, el primer centro c_1 se elije aleatoriamente. Habiendo elegido c_i, el centro c_{i+1} se elije de la zona que c_i consideraría como núcleo duro. Para ello, se calcula el histograma local de c_i y se toma como centro c_{i+1} a un elemento que se encuentre en la zona central del histograma local de c_i. Esta zona es la región de mayor concentración de elementos, si el histograma tiene forma de campana de Gauss.

Siguiendo los lineamientos dados en [2], la zona de mayor concentración de elementos se puede determinar usando la media del histograma local de c_i. La idea es elegir como centro c_{i+1} a un elemento cuya distancia a c_i se encuentre en el intervalo $[\mu - x, \mu + x]$ donde μ es la media del histograma local de c_i y x es un número entero. El valor más conveniente para x se determinó experimentalmente y se explica en detalle en la sección 5.

5. EVALUACIÓN EXPERIMENTAL

5.1. Descripción de los Experimentos

Los experimentos fueron realizados sobre diccionarios de palabras usando como función de distancia la distancia de edición. Esta función es discreta y calcula la mínima cantidad de caracteres que hay que agregar, cambiar y/o eliminar a una palabra para obtener otra. Este modelo es comúnmente usado en recuperación de texto, procesamiento de señales y aplicaciones de biología computacional.

Se utilizaron en total 4 diccionarios: Español (de 86.061 palabras), Francés (de 138.257 palabras), Italiano (de 116.879 palabras) e Inglés (de 69.069 palabras).

En la indexación se utilizaron aridades $2, 4, 8, 16, 32, 64, 128, 256$ y 512

Los experimentos se realizaron en dos etapas. La primera estuvo dedicada a determinar el valor más adecuado para x en la política de selección *zona de mayor concentración*. Habiendo establecido

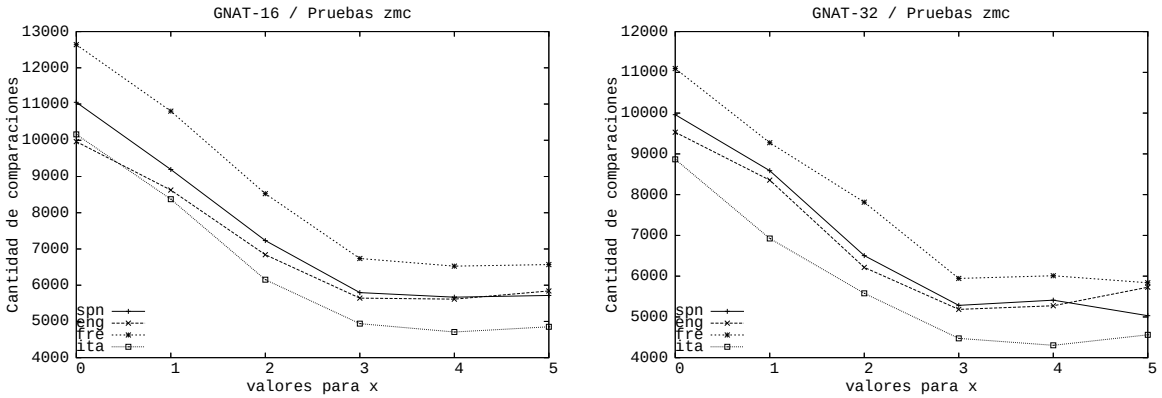

Figura 4: *Resultados para la política zona de mayor concentración con aridades* 16 *y* 32.

este valor, procedimos a comparar las políticas diseñadas con la política de selección de centros trivial, selección random, a fin de establecer el desempeño de las mismas.

Por cuestiones de espacio, en las próximas secciones sólo mostramos las gráficas de los resultados que consideramos más significativos.

5.2. Política zona de mayor concentración

Para determinar el valor más adecuado para x se realizaron experimentos con $x = 0, 1, 2, 3, 4, 5$, para todas las aridades.

De cada diccionario, para indexar, se tomó sólo una muestra aleatoria del $50\,\%$ de elementos. Sobre cada GNAT creado, se realizaron búsquedas por rango con radio $r = 1$ con un lote de prueba formado por una muestra aleatoria del $5\,\%$ de los elementos. En esta etapa no se utilizaron los diccionarios completos porque el objetivo era obtener resultados orientativos respecto del valor más conveniente para x.

Las figuras 4 y 5 muestra los resultados obtenidos con los diccionarios Español (denotado con spn en la gráfica), Francés (denotado con fre), Italiano (denotado con ita) e Inglés (denotado con eng). Sobre el eje x están representados los distintos valores de x y sobre el eje y el número medio de comparaciones necesitadas para resolver una búsqueda por rango con radio $r = 1$.

En cada una de las aridades puede apreciarse que las curvas, para los diferentes diccionarios, conservan cierto patrón que nos permite determinar el valor de x con el que se realizarán el resto de los experimentos.

Se puede observar que, para un GNAT de aridad menor a 64 (figura 4), el valor más adecuado oscila entre 3 y 4. Puede apreciarse una brusca mejora la principio y una tendencia a estabilizarse a partir de $x = 3$. Curiosamente, las curvas de aridad mayor a 64 (figura 5) pierden uniformidad entre los diferentes diccionarios y también su tendencia a ser decrecientes a medida que aumentamos el valor de x.

Dado que las gráficas de las figuras 4 y 5 fueron realizadas sólo para radio de búsqueda $r = 1$ y con una muestra del diccionario, son sólo orientativas para realizar una selección del valor de x. Luego, experimentalmente se comprobó que, para una minoría de aridades, los valores no eran los mejores para todos los radios de búsqueda, sino que para radios de búsqueda de menor selectividad otras desviaciones eran mucho más beneficiosas. Como las búsquedas se realizarán con distintos valores de r no podemos en el momento de indexar realizar una elección que dependa de r. Por esto seleccionamos como mejor valor de x a aquel que, si bien no obtiene el mejor desempeño en cada radio de búsqueda, se mantiene cerca del mínimo en todos ellos.

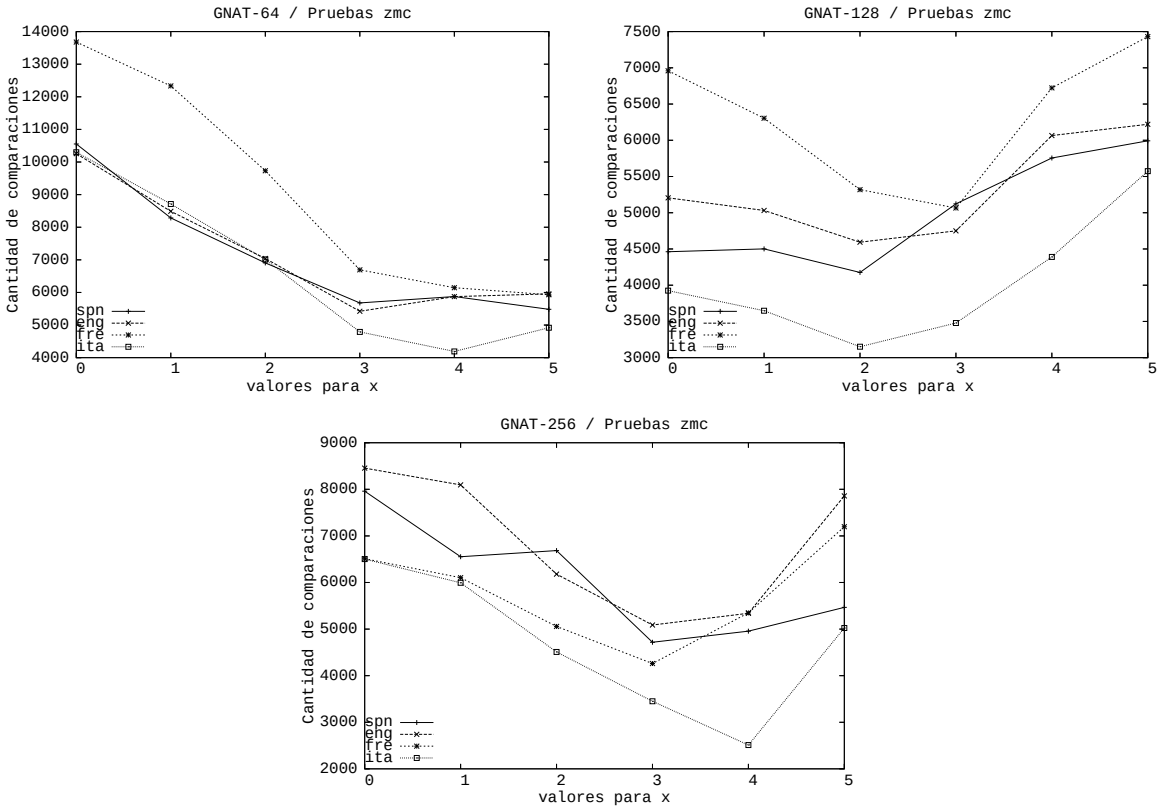

Figura 5: *Resultados para la política zona de mayor concentración con aridades* 64*,* 128 *y* 256*.*

La tabla 1 muestra los valores que finalmente resultaron elegidos para cada aridad en los distintos diccionarios utilizados.

Aridad	Español	Francés	Italiano	Inglés
2	$x = 4$	$x = 4$	$x = 4$	$x = 4$
4	$x = 4$	$x = 4$	$x = 4$	$x = 4$
8	$x = 4$	$x = 4$	$x = 4$	$x = 4$
16	$x = 4$	$x = 4$	$x = 4$	$x = 4$
32	$x = 3$	$x = 4$	$x = 4$	$x = 3$
64	$x = 3$	$x = 4$	$x = 4$	$x = 3$
128	$x = 4$	$x = 3$	$x = 3$	$x = 3$
256	$x = 2$	$x = 4$	$x = 2$	$x = 3$
512	$x = 3$	$x = 3$	$x = 4$	$x = 2$

Tabla 1: *Valores de* x *para la política zona de mayor concentración.*

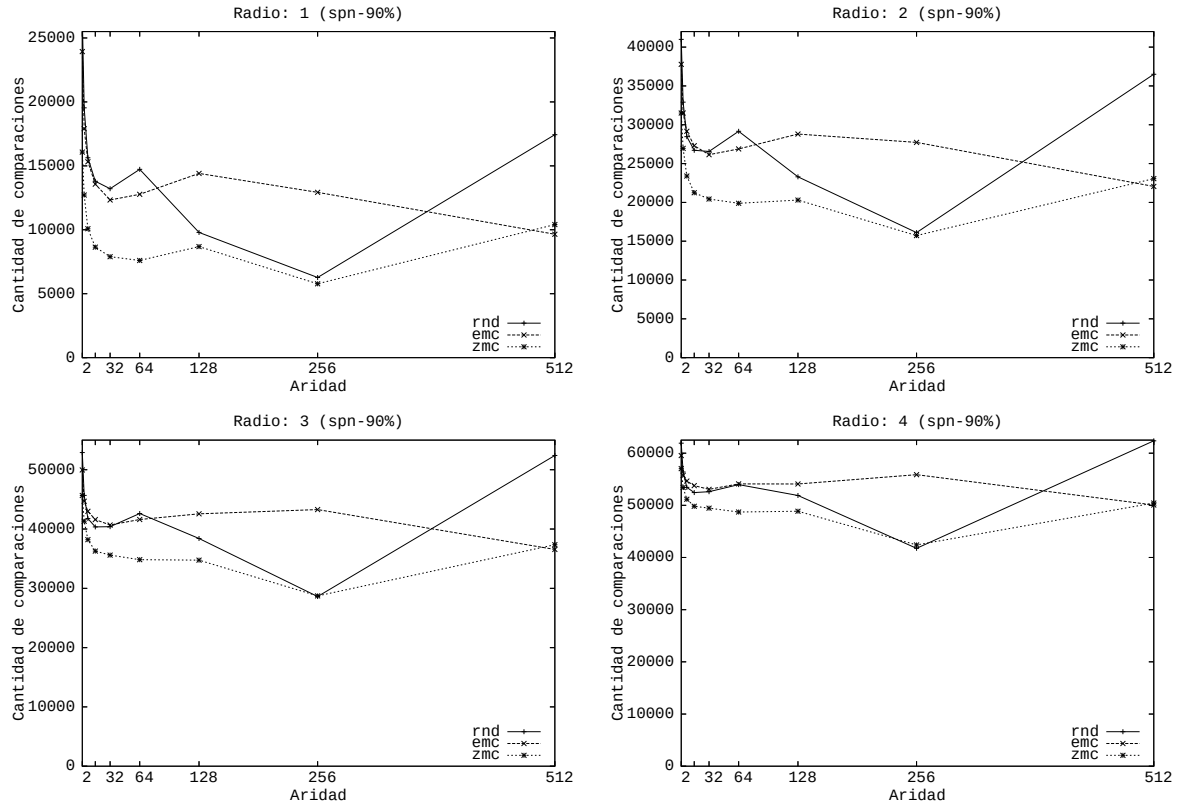

Figura 6: *Resultados para el diccionario Español*

5.3. Comparación entre las distintas políticas

En esta etapa, los experimentos se orientaron a determinar el desempeño de las políticas de selección de centros propuestas. Para ello, de cada diccionario se indexó el $90\,\%$ de los elementos y el $10\,\%$ restante se utilizó como lote de prueba para realizar búsquedas por rango con radio $r = 1, 2, 3, 4$.

Dado que cada nodo del GNAT ocupa una espacio $O(m^2)$ (donde m es la aridad del árbol), nuestros experimentos persiguieron dos objetivos. En principio analizamos qué política logra mejores resultados bajo condiciones de igual uso de memoria (es decir, con la misma aridad). Luego, estudiamos cuál es la política que logra el mejor desempeño global, más alla de cuál sea el espacio utilizado.

La razón para hacer esto es la siguiente. Si bien nuestra medida de complejidad es la cantidad de evaluaciones de la función de distancia d, hay que tener presente que si el índice no entra en memoria principal el tiempo consumido en I/O degrada el tiempo de respuesta. Por eso es importante conseguir buenos resultados pero sin aumentar excesivamente la aridad y, en consecuencia, la memoria necesaria para el índice.

La figura 6 muestra los gráficos comparativos de las diferentes políticas para el diccionario Español. Sobre el eje x hemos representado las distintas aridades utilizadas y sobre el eje y el número medio de comparaciones necesitadas para resolver una búsqueda por rango con radio $r = 1, 2$ (arriba) y $r = 3, 4$ (abajo). En los gráficos se ha denotado con rnd a la política random, emc a la política elemento más cercano y zmc a la política zona de mayor concentración.

Se puede observar que para aridades menor o igual que 64 el comportamiento de las distintas políticas de selección es similar, en el sentido de que todas tienen a decrecer a medida que aumenta la aridad. La política de selección por *zona de mayor concentración* proporciona mejoras sobre las anteriores de alrededor del $40\,\%$ en cantidad de comparaciones para búsquedas de mayor selectividad ($r = 1$) y hasta un $10\,\%$ en búsquedas de menor selectividad ($r = 4$).

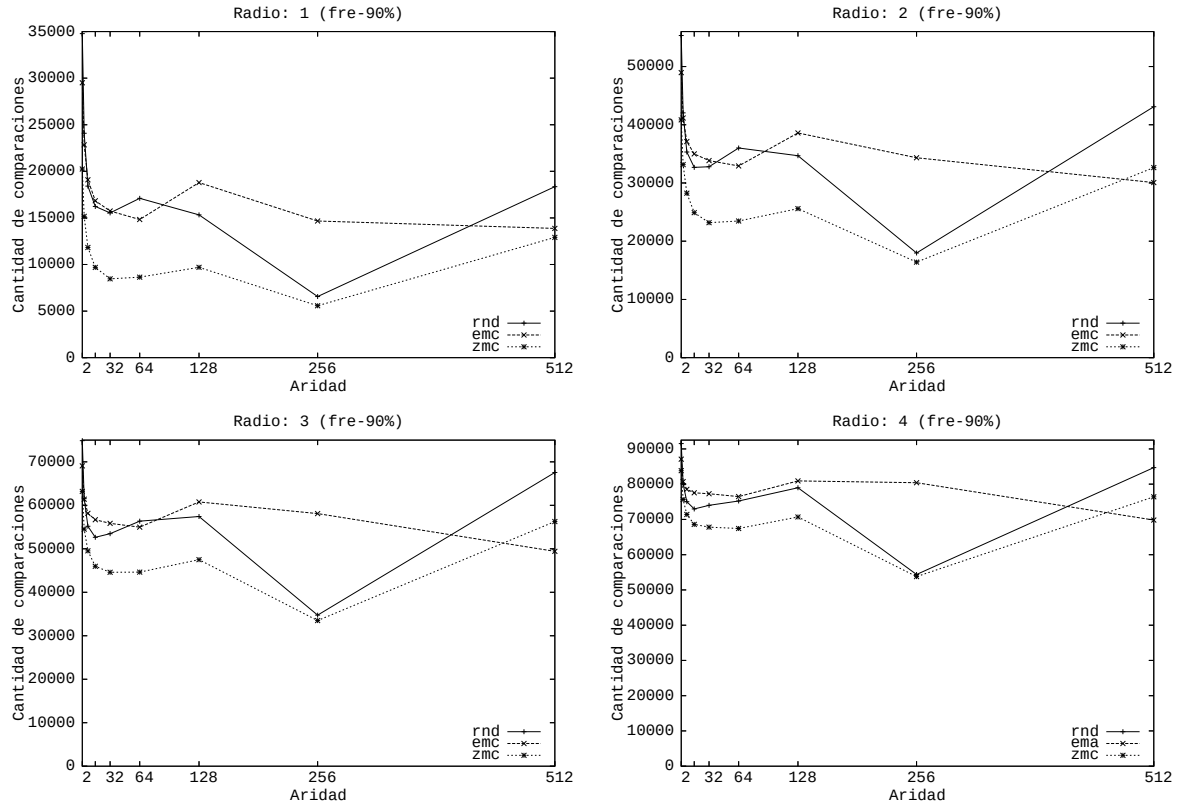

Figura 7: *Resultados para el diccionario Francés*

Para aridades mayores a 64 puede observase claramente lo que ya mencionáramos en secciones anteriores: aumentar la aridad del árbol no siempre produce mejora en el desempeño del índice. En el caso de las políticas *random* y *zona de mayor concentración*, se nota una fuerte degradación de la performance cuando se aumenta la aridad de 256 a 512, y en la política *elemento más cercano* cuando se aumenta de 64 a 128. En este último caso, podemos ver que *elemento más cercano* tiene peor desempeño que una selección random.

Las aridades que producen un mejor desempeño resultan ser 256 para *random* y *zona de mayor concentración*, y 512 para *elemento más cercano*. Para este último caso, *elemento más cercano* duplicando el espacio utilizado no logra superar los resultados obtenidos con las otras políticas.

Para el diccionario Español podemos concluir que, para la mayoría de los casos, la política más competitiva es la de *zona de mayor concentración*. La política *random* sólo logra supera mínimamente a *zona de mayor concentración* con aridad 256 y radio de búsqueda $r = 4$. Si la memoria es insuficiente para mantener un GNAT de aridad 256 claramente la mejor opción es usar *zona de mayor concentración* con aridad 64.

Los resultados obtenidos con los demás diccionarios presentan las mismas características descriptas para el diccionario de español. Las figuras 7, 8 y 9 muestran los resultados obtenidos con los diccionarios Francés, Italiano e Inglés respectivamente.

En el caso del diccionario Inglés, si bien los resultados son similares a los patrones del resto de los diccionarios, llama la atención la mejora sustancial que tiene *elemento más cercano* sobre *zona de mayor concentración* con aridad 512. Es decir, *elemento más cercano* logra superar a *zona de mayor concentración* cuando la aridad es 512 en ambos casos, pero *elemento más cercano* con aridad 512 no logra superar a *zona de mayor concentración* con aridad 256.

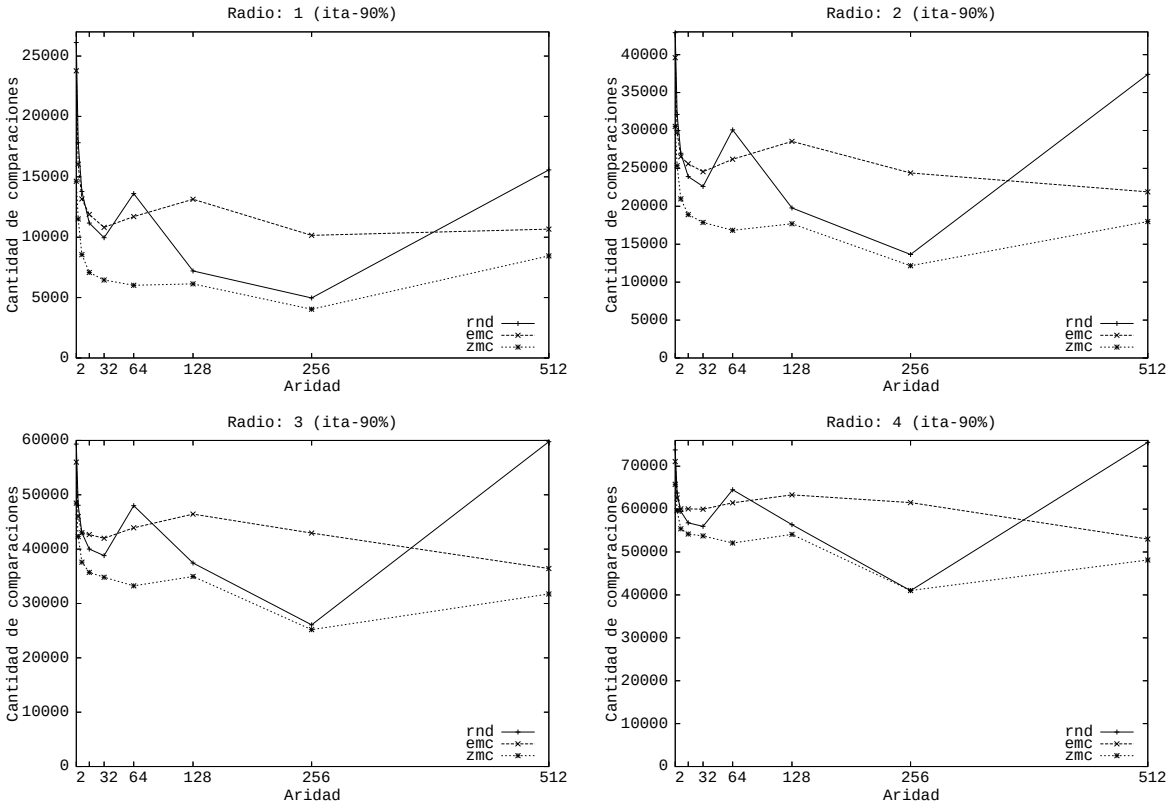

Figura 8: *Resultados para el diccionario Italiano*

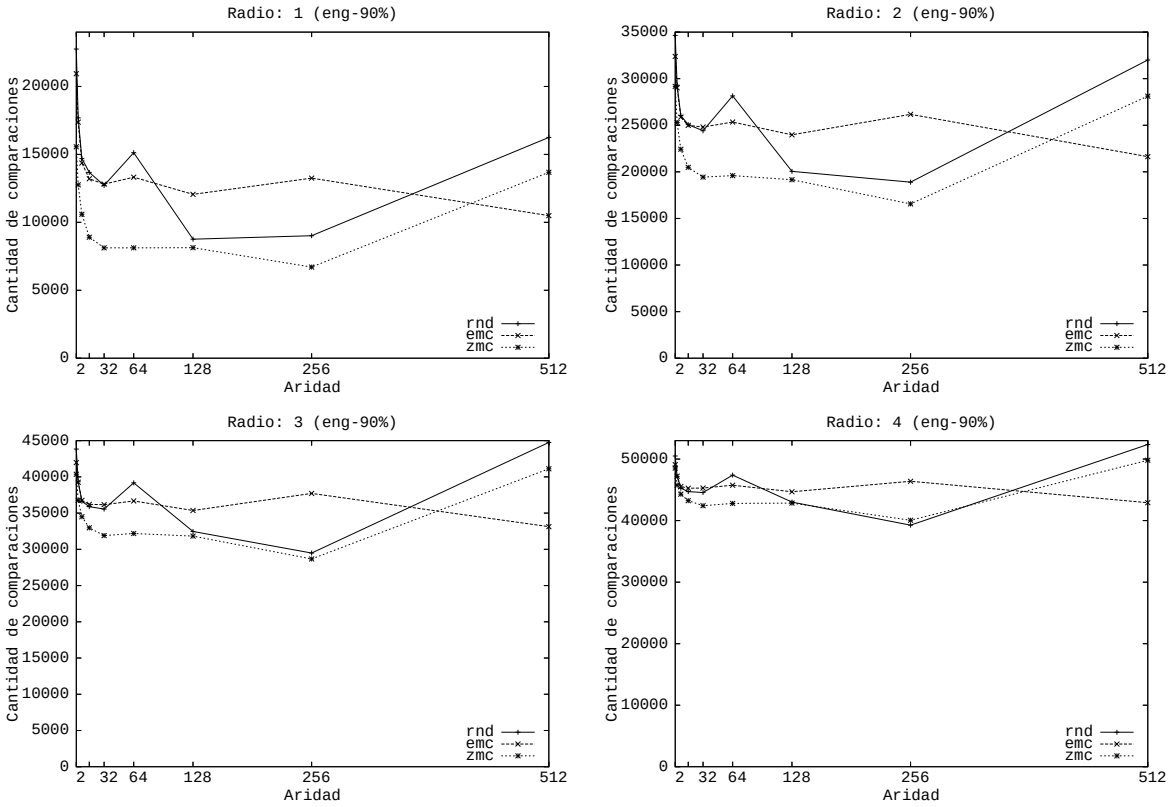

Figura 9: *Resultados para el diccionario Inglés*

Resumiendo, de los resultados mostrados se pueden considerar dos grandes enfoques: uno de ellos es la búsqueda de balance entre cantidad media de comparaciones y espacio requerido en memoria, y el otro camino es minimizar la cantidad media de comparaciones.

En el primero de los enfoques, el uso del GNAT con aridades 16, 32 o 64 con selección de centros por *zona de mayor concentración* es el más competitivo; el primero es el que tiene menor rendimiento, especialmente para el diccionario de italiano. Para el segundo enfoque, el uso de un GNAT con aridad 256 con política de selección por *zona de mayor concentración* provee muy buenos resultados en la mayoría de los casos; sólo para búsqueda de menor selectividad el GNAT de aridad 256 con selección de centros *random* realiza algunas comparaciones menos en los diccionarios Español e Inglés.

La política de selección por *elemento más cercano* iguala el desempeño de una selección aleatoria en aridades menores a 32 en los diccionarios Español e Inglés para búsquedas de alta selectividad. En general, se comporta peor que una selección aleatoria para aridades 128 y 256. Sin embargo, tiene buen rendimiento con aridades grandes para todos los radios de búsqueda. Intuimos que el problema está en el sesgo que implica elegir como próximo centro el elemento más cercano al anterior. La única manera que tiene esta técnica de introducir diversidad en los centros elegidos es aumentar la cantidad de centros llevando la aridad del GNAT a 512. Todo lo expuesto la posiciona como una política que requiere mayor desarrollo e investigación.

6. CONCLUSIONES Y TRABAJO FUTURO

En este trabajo hemos presentado dos nuevas políticas para la selección de centros durante la construcción del GNAT, a saber: *elementos más cercano* y *zona de mayor concentración*. La menos competitiva fue *elemento más cercano* logrando un peor desempeño que una selección aleatoria. La más competitiva fue *zona de mayor concentración*, logrando reducciones importantes en la cantidad de evaluaciones de la función de distancia respecto de una selección aleatoria.

Con respecto al trabajo futuro nos proponemos estudiar el comportamiento de estas políticas sobre otros espacios métricos, adaptándolas en caso de ser necesario. Recordemos que las políticas presentadas se basan en histogramas en forma de campana, algo que no sucede en todos los espacios.

También no proponemos estudiar en detalle las causas del bajo rendimiento de la política *elemento más cercano* y, en función de ello, modificarla para lograr mejorar su desempeño.

REFERENCIAS

[1] F. Aurenhammer. Voronoi diagrams – a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3), 1991.

[2] R. Baeza-Yates, B. Bustos, E. Chávez, N. Herrera, and G. Navarro. *Clustering in Metric Spaces and Its Application to Information Retrieval*. Kluwer Academic Publishers, 2003. ISBN 1-4020-7682-7.

[3] S. Brin. Near neighbor search in large metric spaces. In *Proc. 21st Conference on Very Large Databases (VLDB'95)*, pages 574–584, 1995.

[4] E. Chávez, G. Navarro, R. Baeza-Yates, and J.L. Marroquín. Searching in metric spaces. *ACM Computing Surveys*, 33(3):273–321, September 2001.

[5] J. Uhlmann. Satisfying general proximity/similarity queries with metric trees. *Information Processing Letters*, 40:175–179, 1991.