

EL ARBOL MULTIDIMENSIONAL EQUILIBRADO EN ALTURA: INFLUENCIA DE SU COMPORTAMIENTO DINAMICO EN LAS RECUPERACIONES EXACTA, EN RANGO Y DE LOS VECINOS MAS PROXIMOS.

O. SANTANA, M. DIAZ, Z. HERNANDEZ, J.C. DEL PINO
Departamento de Informatica y Sistemas
Universidad Politecnica de Canarias
Apto. 550, Las Palmas de Gran Canaria, España.

Resumen:

En este trabajo se presenta un estudio empirico del comportamiento dinamico de la estructura de arbol multidimensional equilibrado en altura [VV84], analizando los parametros fundamentales que determinan su respuesta ante inserciones y extracciones continuadas y ante una fase estacionaria. Asimismo, se estudia su efectividad frente a los árboles quintarios [LW80, SM87] en atender a las interrogaciones exacta y en rango, y su comportamiento en la busqueda de los vecinos mas proximos.

0- INTRODUCCION

El árbol multidimensional equilibrado en altura, KDHBT [VV84], es una generalización del árbol AVL [AV62]. Las principales características de esta estructura son su eficiente uso de memoria, y su capacidad dinamica de respuesta a las inserciones y extracciones mediante sencillos reequilibrados. La inserción de un dato en la estructura requiere en el peor de los casos una sola operación de reestructuración. Pero estas ventajas le hacen perder eficiencia en determinadas búsquedas con respecto a otras estructuras similares, aunque estáticas o con un alto coste en las operaciones de inserción y extracción o con una elevada demanda de memoria. Aunque han sido propuestas gran número de otras generalizaciones, los árboles KDHBT son una estructura atractiva.

En la seccion 1 se describe la estructura KDHBT como un estado mas avanzado del arbol de busqueda k-dimensional, KDT. De especificar sus condiciones de equilibrio y las acciones a tomar, reequilibrados, cuando se produzca una perturbacion del mismo, a fin de reestablecerlo, se encarga la seccion 2. En la seccion 3 se expone como se llevan a cabo las operaciones que producen modificaciones estructurales en el arbol, asi como las operaciones que deben realizarse para garantizar la estructura KDHBT. Para valorar el comportamiento de la estructura en inserción y extracción asi como ante los distintos tipos de interrogaciones se realizaron pruebas estadísticas midiendo los parámetros indicativos del coste de estas operaciones. En la seccion 4 figuran las pruebas realizadas con inserciones y extracciones: Fase creciente (comportamiento durante una inserción continuada), Fase estacionaria (comportamiento con inserción y extracción aleatoria), Fase decreciente (extracción continuada). En la seccion 5 se comparan las realizaciones con que se llevan a cabo las interrogaciones exacta y en rango en el arbol Quintario [LW80 y SM87] y en el KDBHT. La busqueda de los M-vecinos mas proximos se introduce en la seccion 6; para a continuacion estudiar su conducta en la seccion 7 mediante un proceso de simulacion. En la seccion 8 se presentan las conclusiones de este trabajo.

1- DESCRIPCION DE LA ESTRUCTURA

1.1 ARBOL KDT

Un fichero k-dimensional o fichero-k es un conjunto de N registros, en el que cada registro está identificado por una k-tupla ordenada $(X_k, \dots, X_1)$ de valores que se llaman claves o atributos.

Un árbol de búsqueda k-dimensional, T, con $k \geq 1$, para representar un fichero-k, F, está definido como: $T=(u, T_i, T_c, T_d)$, donde u es la raíz del árbol que guarda una clave, de valor s en la dimensión k; T_i , T_c , y T_d se denominan respectivamente, subárbol izquierdo, subárbol central, y subárbol derecho de u. T_i , es un árbol de búsqueda k-dimensional para todos los registros en F en los que sus claves en la dimensión k son menores que s; T_c es un árbol de búsqueda de dimensión (k-1) representando un subespacio de T cuyos registros están identificados por la (k-1)-tupla ordenada $(X_{k-1}, \dots, X_1)$, tomada de los registros en F para los que su clave X_k es igual a s; T_d , el subárbol derecho, es un árbol de búsqueda de dimensión k para todos los registros en F en los que sus claves en la dimensión k son mayores que s.

Sea un nodo R de altura h_1 que tiene un hijo izquierdo directo P, con altura h_2 (si $h_2 = 0$ implica que el subárbol está vacío). Entonces se dice que R es el padre derecho directo de P; además, los nodos no existentes en el camino entre R y P en las alturas $h_2+1, \dots, h_1-1$ también se llaman hijos directos de R (se consideran nodos no existentes, a todos los nodos que debieran existir entre dos nodos enlazados, con la posibilidad de que el receptor del enlace sea nulo, para que entre ellos existan nodos a todas las alturas posibles). Dos hijos izquierdos o derechos no existentes en las alturas h y h+1 constituyen un par (de nodos) n.e. (no existentes) a altura h.

1.2 ARBOL KDHBT

Sea P, un nodo en un árbol de búsqueda k-dimensional, que tiene algun par n.e. de hijos izquierdo o derecho directos a altura h_1 . Sea T_1 de altura h_2 el subárbol central de P, y T_2 de altura h_3 el subárbol central del padre izquierdo o derecho, directo o indirecto de P (si existe). T_2 y T_1 son, respectivamente, los subárboles adyacentes izquierdo y derecho del par n.e. El par n.e. se dice soportado directamente por T_1 si $h_2 \geq h_1$, y se dice soportado indirectamente por T_2 si $h_2 < h_1$ y $h_3 \geq h_1$. Se dice que el par n.e. está soportado si lo está directa o indirectamente.

Un árbol k-dimensional equilibrado en altura o KDHBT, con $k \geq 1$, representa un fichero-k, F, mediante un árbol de búsqueda k-dimensional que almacena F, con la restricción de que todo par n.e. en el árbol está soportado.

2- OPERACIONES DE REEQUILIBRADO. ROTACIONES

La condición de equilibrio en altura de un nodo P, de altura h se dice que es "r" si la altura de su subárbol central es h-1, si no; la condición de equilibrio será:

- "=" si la altura de su subárbol izquierdo es igual a la de su subárbol derecho y es h-1;
- "-" si su subárbol izquierdo es más alto que su subárbol derecho;
- "+" si su subárbol derecho es más alto que su subárbol izquierdo;

Si P tiene un par n.e., f, de hijos directos izquierdos o derechos de altura h, el cual es el único posible par n.e. no soportado en el subárbol que tiene como raíz P, así como en el camino entre P y la raíz del árbol. Entonces, se dice que P tiene una violación de estructura izquierda o derecha, respectivamente, o simplemente una violación de estructura, si f no está directamente soportado. Se dice que f es el par n.e. asociado a la violación de estructura. La violación de estructura es fuerte si f tampoco está soportado indirectamente. La violación de estructura se dice que es especial si la altura del subárbol izquierdo o derecho o el subárbol central de P es h-1. Un nodo P en T se dice que tiene una doble violación de estructura a altura h si P1, un hijo directo o indirecto izquierdo de P, tiene una fuerte violación de estructura izquierda a altura h, y P2, un hijo directo o indirecto derecho de P, tiene también una fuerte violación de estructura a altura h. Los pares n.e. asociados a la fuerte violación de estructura de P1 y P2, se dice que son los pares n.e. asociados de una doble violación de estructura.

Las operaciones de reequilibrado se usan para corregir una violación de estructura. Sea T un KDHBT con un nodo P de altura h que tiene una violación de estructura a altura h-2. Sea su par n.e. asociado, el único par n.e. en el árbol que puede no estar soportado. En tal situación, se aplica en el nodo la rotación simple cuando la condición de equilibrio de Q, el hijo de P opuesto al par n.e. es "r", "=", o igual a la de P. La doble rotación tipo A se aplica cuando no se aplica la simple y la violación de estructura es especial o la altura del subárbol de Q mas proximo al par n.e. es h-3. La doble rotación tipo B se aplica en los demás casos.

3- INSERCIÓN Y EXTRACCIÓN

La inserción se realiza buscando (de la misma forma que en un árbol binario) para cada dimensión el valor del atributo del registro a insertar para esa dimensión, si se encuentra se desciende por el hijo central, repitiéndose el proceso anterior para la siguiente dimensión. Cuando se alcanza un nodo terminal (enlace nulo) en una dimensión dada, se creará el nodo para esta dimensión y las siguientes uniendolos por el subárbol central, si se encuentra la clave en la última dimensión entonces es que el registro ya existía. Después de creados los nodos del registro insertado, se comienza el ascenso en el que se comprueba si en el nodo actual tras la inserción ha variado la altura del hijo por el que se descendió, entonces se actualiza los campos altura y condición

y se comprueba si necesita un reequilibrado, en cuyo caso se realiza el tipo de rotación apropiado según las siguientes condiciones: Si la condición del subárbol mayor es "r", "=" o la misma que la del nodo desequilibrado, se realiza una rotación simple, si no, siendo h la altura del nodo desequilibrado, si la altura de alguno de los hijos del nodo desequilibrado, después de una hipotética aplicación de la doble rotación A, tiene una altura de "h-3", se aplica una rotación doble A; si no, se aplica una rotación doble B.

En la extracción, se desciende por el árbol en busca del registro a extraer. Cuando se encuentra el nodo de la última dimensión este se elimina y se asciende eliminando los nodos pertenecientes al registro que no forman parte de otro. El último nodo a eliminar será el que se encuentre en un nivel con varios nodos (excepto cuando el árbol no tenga sino un registro), para eliminar este nodo existen distintas posibilidades según tenga hijos o no. Si tiene los dos hijos (izquierdo y derecho) se busca el predecesor y se sustituye por él. Si no, simplemente se elimina y se sustituye por su hijo si lo hubiera. En este proceso se debe verificar si la disminución de altura provoca algún desequilibrio en cuyo caso se corregirá. Seguidamente se continúa el ascenso arreglando las posibles violaciones de estructura debidas a la disminución de altura. Si esta se ha producido en un hijo central se examina la posible doble violación, corrigiéndola si la hubiera.

4- PRUEBAS DE COMPORTAMIENTO DE LA ESTRUCTURA

Las pruebas realizadas para estudiar el comportamiento normal de la estructura son las siguientes: Fase Creciente, estudio de la estructura durante la inserción; Fase Estacionaria, inserción y extracción aleatoria; Fase Decreciente, comportamiento en extracción. En estas pruebas se midió el porcentaje de rotaciones realizadas.

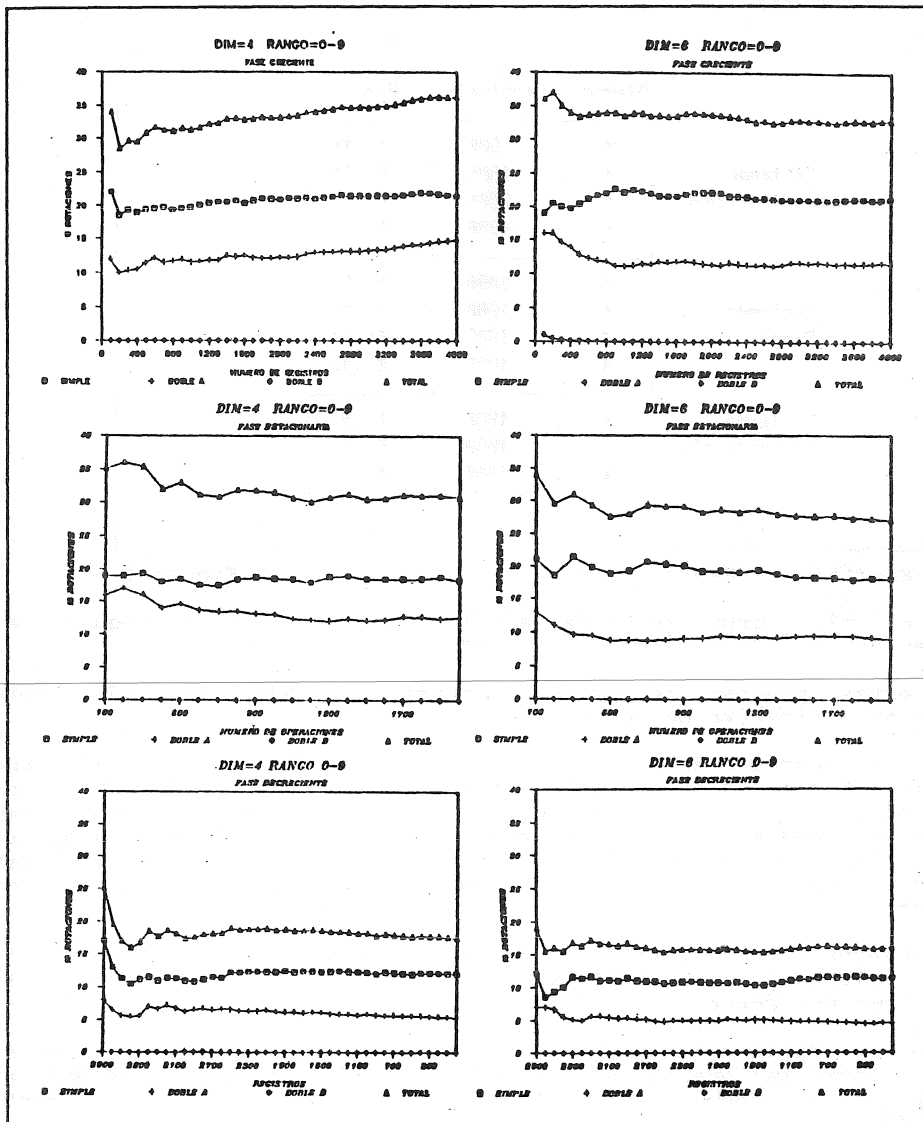
Se realizaron pruebas para dimensión 4 y 6, insertando 4000 registros en la Fase Creciente, después insertando y extrayendo 1000 registros aleatorios en la Fase Estacionaria y finalmente extrayendo sucesivamente los 4000 registros de forma aleatoria en la Fase Decreciente. El rango de claves usados en todas las pruebas fue de 0 a 9.

La figura 1 muestra los resultados de rotaciones en cada fase. Los valores se calcularon como $[\text{rotaciones} \times 100 / \text{operaciones(I+E)}]$.

Se puede apreciar que la diferencia de una dimensión a otra no es excesiva, aunque con tendencia a disminuir el número de rotaciones al aumentar la dimensión. En las tres fases el número de rotaciones casi permanece constante, aunque hay diferencias entre ellas. La fase decreciente es la que menos rotaciones necesita, al igual que ocurre en los AVL, esto es debido a que es más difícil desequilibrar un árbol al extraer un registro que al insertarlo. Esta es la razón por la que en la fase estacionaria se necesiten menos rotaciones que en la creciente, ya que en ella se intercalan aleatoriamente inserciones y extracciones.

El comportamiento de los diferentes tipos de rotaciones, parece ser el previsto: la rotación simple es la que se produce más frecuentemente,

debido a las condiciones poco restrictivas para su aplicación, en contraste con los tipos dobles, en los que, sobre todo en el tipo B, su aparición viene muy condicionada a unas restricciones que en muy pocos casos se dan.



Rotaciones en las tres fases para DIM=4,6.

Figura 1

5- BUSQUEDA EXACTA Y EN RANGO, COMPARACION CON LOS ARBOLES QUINTARIOS

Las pruebas realizadas para las búsquedas exactas y en rango se realizaron variando la Cardinalidad, Dimensionalidad y Rango de claves. En la figura 2 se muestran las pruebas realizadas variando los parámetros indicados.

	Dimenci.	Cardinali.	Rango
Variando Cardinalidad	4	500	0..14
	4	1000	0..14
	4	2000	0..14
	4	4000	0..14
Variando Rango	4	1000	0..9
	4	1000	0..11
	4	1000	0..14
	4	1000	0..16
Variando Dimensionali.	4	1000	0..9
	6	1000	0..9
	8	1000	0..9

Búsqueda Exacta y en Rango.

Figura 2

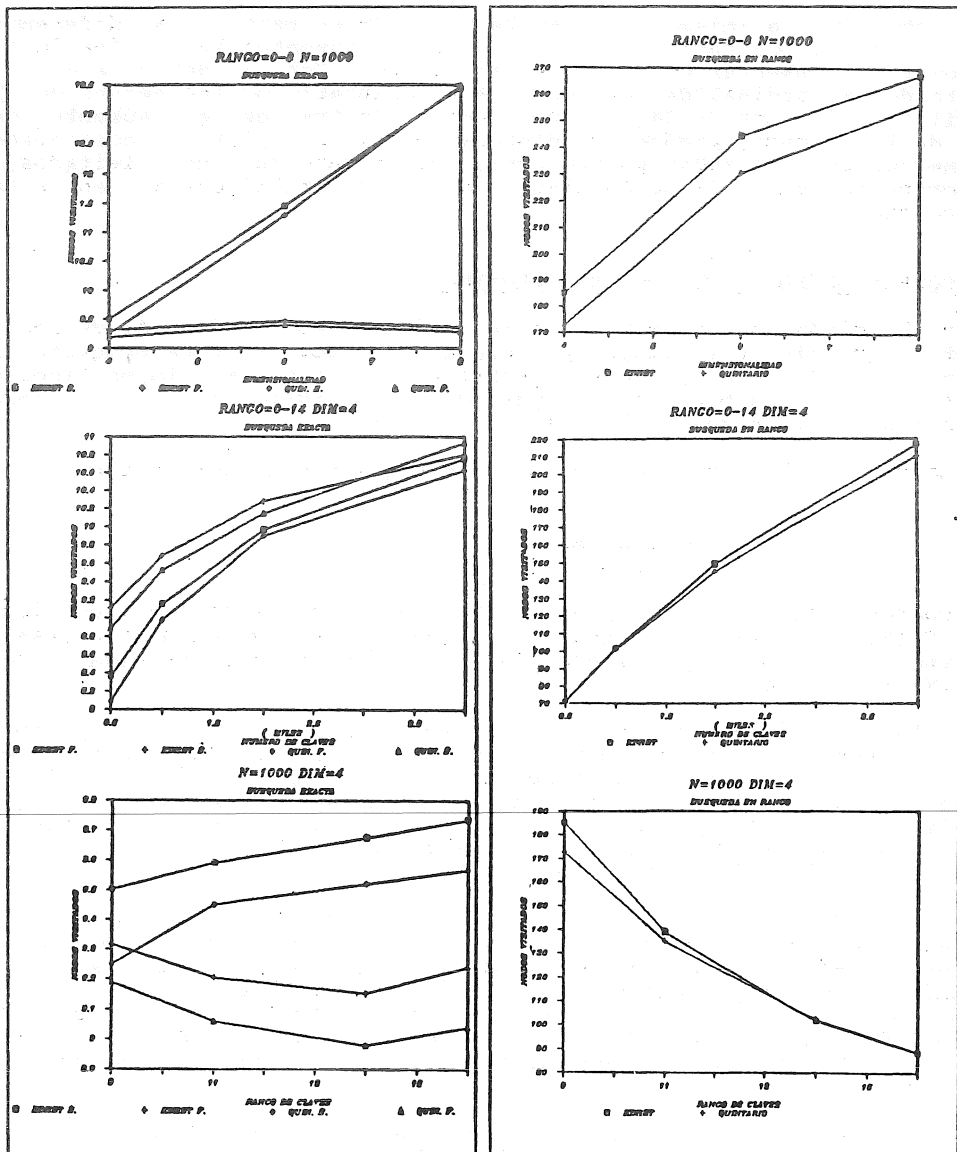
En el primer grupo se ha variado la cardinalidad en el segundo el rango y en el tercero la dimensionalidad.

En estas pruebas se evaluó el número de nodos visitados como parámetro significativo de comparación.

Para la obtención de los resultados se generó un árbol de cada uno de los tipos mostrados en el cuadro anterior y se realizaron sobre ellos 500 búsquedas aleatorias exactas con éxito y 500 con fallo, tomando como resultado la media de las 500 pruebas. Para la búsqueda en rango se varió el hipercubo entre uno y cinco y se eligieron 500 claves aleatorias para cada hipercubo, tomando como resultado la media de las 2500 pruebas.

Se observa en la figura 3 que el comportamiento del quintario en la búsqueda exitosa y con fallo es mejor que el del KDHBT aunque la diferencia es muy pequeña.

Que el quintario sea mejor, es el resultado de que su estructura se genera de forma tal que si se eliminan las proyecciones (no usadas en este tipo de búsqueda) queda un árbol multidimensional completamente equilibrado.



Búsqueda exacta y en rango.

Figura 3

En las pruebas de búsqueda en rango los árboles quintarios son superiores a los KDHT debido al uso que se hace en los primeros de las proyecciones, lo que agiliza parte de la búsqueda. De cualquier forma la diferencia entre uno y otro no es importante.

Se observa, asimismo, que en la búsqueda en rango, la diferencia entre el árbol Quintario y el KDHT crece al aumentar la cardinalidad y disminuye al aumentar el rango de las claves. Esto es debido a que el aumento de la cardinalidad proporciona al Quintario un incremento de la posibilidad de uso de las proyecciones, mientras que el aumento del rango de las claves, produce un dispersión de los registros, ocasionando la disminución, en ambas estructuras, del número de nodos visitados, y un decremento, en el árbol Quintario, de las posibilidades de uso de las proyecciones.

6- BUSQUEDA DE LOS M-VECINOS MAS PROXIMOS

La búsqueda de los M vecinos más próximos es la búsqueda de los M registros que distan lo mínimo posible a uno dado. En esta búsqueda la distancia usada es un parámetro importante. Existen multiples tipos de distancia. En el campo numérico para datos multidimensionales se pueden generalizar de la siguiente forma:

La métrica Ln entre dos puntos de dimensión m se define como:

$$d = \eta \sqrt{\sum_{i=1}^m (X_i - Y_i)^2}$$

Cuando n toma el valor 2 se le llama distancia euclídea y es igual a $\sqrt{\sum (X_i - Y_i)^2}$ $i=1..m$. Cuando n es igual a ∞ se le denomina distancia infinita y es equivalente a la distancia definida como $d = \max |X_i - Y_i|$ $i=1..m$.

Estas dos distancias serán las empleadas para realizar las pruebas. Al encontrar los M vecinos más próximos a un punto X, determinados con distancia euclídea, estos se encuentran en una hiperesfera de radio d y centro en X, siendo d la mayor distancia de los M vecinos a X. La infinita a su vez representa un hipercubo centrado en X de lado 2d (siendo d la mayor distancia de los M vecinos a X) con aristas en la union de dos hiperplanos coordenados.

Se recorre la estructura desde el primer nivel, localizando el nodo con valor de clave, en esa dimension, mas cercano al correspondiente de la interrogacion, acumulando la diferencia entre estos valores y descendiendo por su hijo central. El proceso se repite hasta alcanzar el ultimo nivel, en el que se inicia la vuelta atras con exploracion de alternativas por proximidad de los valores de clave en cada nivel, que no entrañen una distancia superior a la del vecino mas alejado hasta ese momento.

La localizacion ordenada por proximidad a un cierto valor, de los valores de clave de un árbol binario de búsqueda que no sobrepasen una diferencia determinada es un problema sencillo. Basta con dividir en dos la secuencia en orden simetrico del árbol e ir recorriendo ambas subsecuencias a partir del punto de corte de forma coherente con la ordenacion creciente de las diferencias al valor de clave dado. En el momento en que una de las subsecuencias se agote, o presente una

diferencia superior a la maxima especificada, se suspende su recorrido. El proceso termina al suspenderse el recorrido de ambas subsecuencias. Este método de búsqueda se basa en el uso de dos pilas para poder realizar la búsqueda de los predecesores y sucesores ordenadamente. El algoritmo realiza los siguientes pasos:

1 Se busca en el árbol el valor de clave dado. Esta búsqueda se realiza de la misma forma que en cualquier árbol binario, con la única diferencia de que se van almacenando los valores de clave que se encuentren durante el descenso, en la pila de sucesores si su valor de clave es mayor que el buscado y en la de predecesores si es menor. Con esto se logra dividir el árbol en dos partes: la de los sucesores y la de los predecesores a la clave buscada, figura 4.

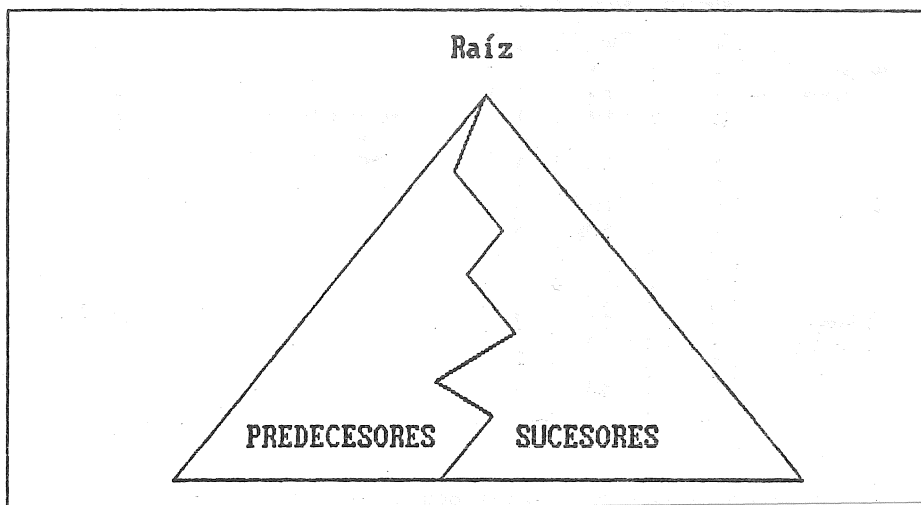


Figura 4

La búsqueda de la clave puede finalizar de dos formas: cuando se encuentre la clave buscada o cuando se encuentre un enlace nulo (el valor de clave buscada no existe).

2.a Si se encuentra el valor de clave buscada, se inserta en ambas pilas, se considera que se ha encontrado un vecino y se procesa. Después se buscará el sucesor y el predecesor de la clave usando cada una de las pilas, si el sucesor o predecesor supera la diferencia permitida se eliminaría de la búsqueda la pila correspondiente.

2.b Si lo que ha detenido la búsqueda es un puntero nulo, entonces se tendrá en la cima de cada pila el sucesor y predecesor de el valor de clave, si alguno de las cimas supera la diferencia permitida se elimina su pila de la búsqueda.

3 Mientras exista predecesor y sucesor se elige el que tenga menor diferencia a la clave y se procesa. A continuación, se busca el sucesor

o predecesor según el caso. El proceso se repite hasta que se cumplan las condiciones de finalización especificadas anteriormente.

7- PRUEBAS REALIZADAS EN LA BÚSQUEDA DE LOS M VECINOS MAS PROXIMOS

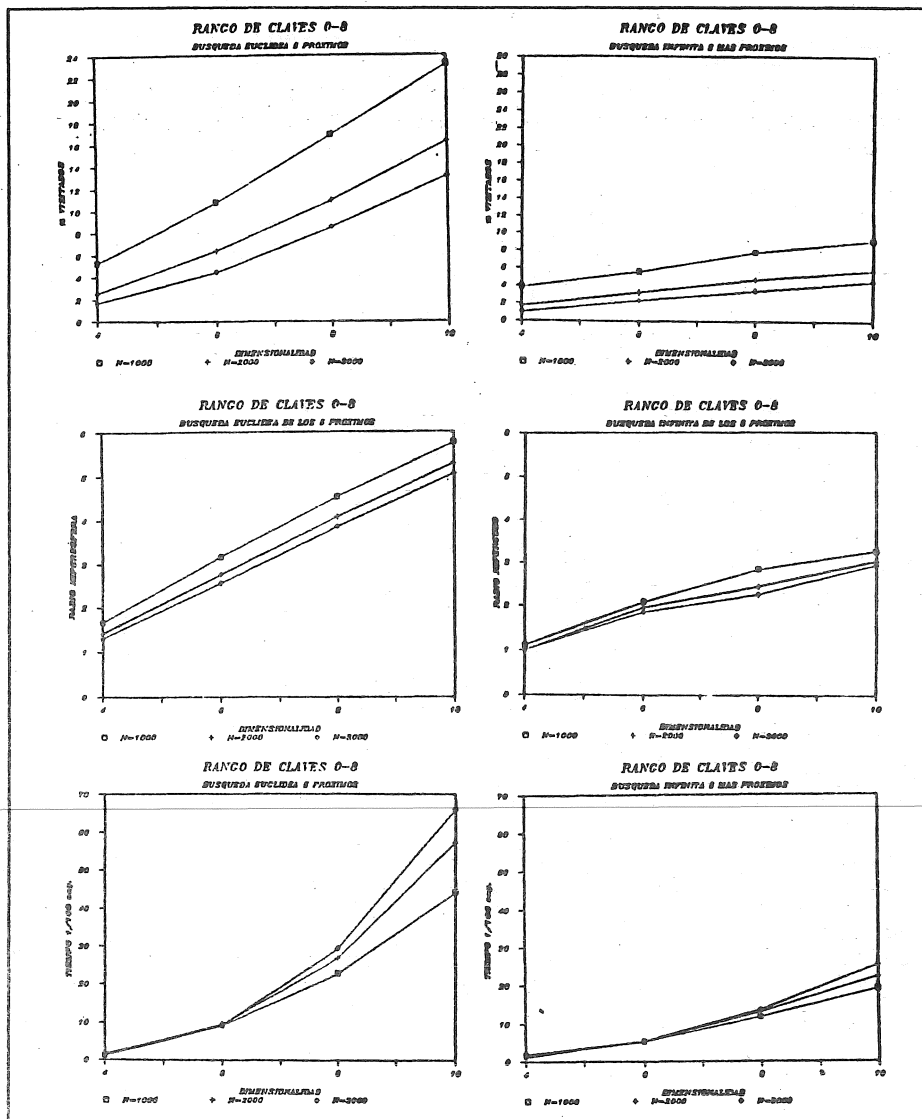
En las pruebas realizadas se variaron los parámetros significativos de la estructura, realizando las mismas pruebas para la distancia euclídea que para la infinita, formando el cuadro de pruebas mostrado en la figura 5.

	Dimensio.	Rango	Nº veci.	
Variando Dimensionalidad	4	0-8	5	Tomando en todas las pruebas valores para 1000, 2500 y 3000 registros.
	6	0-8	5	
	8	0-8	5	
	10	0-8	5	
Variando Rango	8	0-9	5	
	8	0-11	5	
	8	0-14	5	
	8	0-16	5	
Variando Nº vecinos	8	0-8	1	
	8	0-8	5	
	8	0-8	10	
	8	0-8	15	
	8	0-8	20	
	8	0-8	25	

Pruebas en la búsqueda de los M vecinos más próximos. Figura 5

Se realizaron 2000 búsquedas por prueba tomando como resultado el promedio. Se midieron el número de nodos visitados, la distancia máxima de los vecinos encontrados y el tiempo de búsqueda.

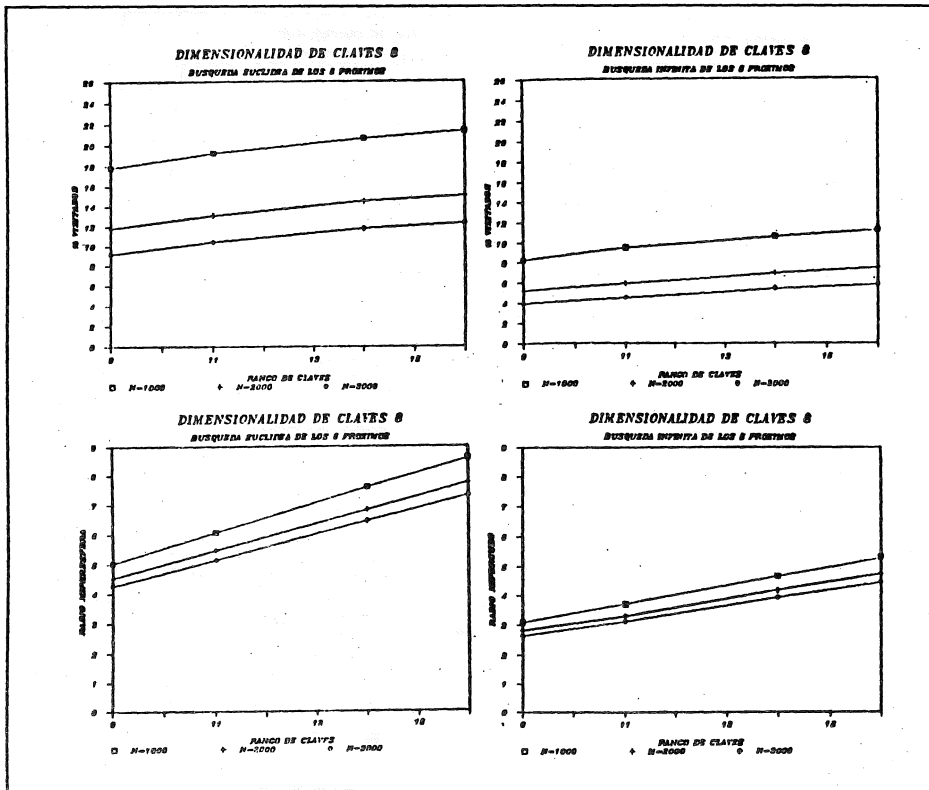
En los resultados de la figura 6, en los que se varía la dimensionalidad de los registros, se aprecia que en la distancia euclídea el tanto por ciento de nodos visitados es mayor que en la búsqueda con distancia infinita. El aumentar el número de registros en el árbol el porcentaje de nodos visitados disminuye, aunque aumenta en valor absoluto. El número de nodos visitados esta directamente relacionado con los radios de una y otra distancia. El tiempo de búsqueda está relacionado con el número de nodos tratados.



Resultados variando la dimensionalidad.

Figura 6

Al variar el rango de los valores de claves se observa en la figura 7, que como es lógico esperar, el aumento de distancia está directamente relacionado con el aumento del rango de las claves. El número de nodos visitados no parece sensible al aumento de dispersión.



Resultados variando el Rango.

Figura 7

El tiempo de búsqueda y el número de nodos visitados, dependen del número de vecinos a buscar, figura 8, aunque con un crecimiento no tan rápido como en las pruebas en que se varía la dimensionalidad. Los radios que se obtienen con una y otra distancia vuelven a ser la explicación de las diferencias que se aprecian entre la búsqueda euclídea e infinita.

8- CONCLUSIONES

A lo largo de este trabajo se han visto distintos aspectos de la estructura KDHBT. El primer punto importante que surge es que sin ser una estructura complicada ofrece dos características que muchas de las otras estructuras en su ambito no ofrecen, como son: dinamismo sin perdida de eficiencia ni coste excesivo de memoria. De las pruebas de comportamiento en fase creciente, estacionaria y decreciente se deduce su capacidad de reaccion ante las acciones de modificación de los datos, teniendo como respuesta pocas acciones de reestructuración y mantenimiento de la altura media. En las pruebas de búsqueda exacta y en rango al ser comparadas con el árbol quintario se comprueba que la perdida de eficiencia debida al dinamismo no es significativa, sobre todo en la búsqueda exacta. En la búsqueda de los vecinos más próximos el comportamiento de la estructura se ve influenciado por la distancia con que se determinan. Con distancia infinita en la realizacion de los experimentos parece conducirse mejor que con la euclidea.

REFERENCIAS

- [AV62] G.M. ADEL'SON-VELSKIJ AND Y. M. LANDIS: An algorithm for the organization of information. Sov. Math. Doklady. Vol 3, pp. 1259-1263, 1962
- [LW80] LEE, D. T. AND WONG, C. K.: Quintary trees: A file structure for multidimensional database systems. Vol 5, N° 3, pp 339-353. 1980
- [SM87] SANTANA, O.; MAYOR, O.; DIAZ, M.; REINA, S.: Arboles Quintarios: Estudio esperimental para interrogaciones exactas, parciales, en rango y en rango parcial. XIII Conferencia Latinoamericana de Informatica. Vol II, pp 1148-1168. 1987
- [VV84] VIJAY K. VAISHNAVI: Multidimensional height-balanced trees. IEEE Transactions on Computers. Vol C-33, N° 4, pp 334-343. 1984.