

REORGANIZACIONES LOCALES EN EL ARBOL-K-D-B. SU EFICIENCIA EN SITUACIONES DINAMICAS

M. DIAZ, O. SANTANA, G. RODRIGUEZ, M. MARTIN
Departamento de Informática y Sistemas
Universidad Politécnica de Canarias
Aptdo. 550, Las Palmas de Gran Canaria, España.

RESUMEN:

En este artículo se introducen diversas reorganizaciones locales para optimizar la ocupación en la actualización de la estructura del árbol K-D-B. Se realiza un estudio experimental en tres fases: creciente, decreciente y estacionaria. En cada fase se compara la evolución de la estructura con reorganizaciones locales y sin ellas, poniéndose de manifiesto el aporte de las reorganizaciones locales en la optimización de la ocupación, sin olvidar el incremento en el número de accesos.

0) INTRODUCCION

La estructura de árbol K-D-B [RO81] es una combinación del árbol K-D [BE75], [BE79], [BF79] y del árbol-B [BM72]. Con ella se espera obtener la eficiencia de la búsqueda multidimensional de los árboles K-D y la eficiencia de accesos de los árboles-B. La partición del espacio en los árboles K-D-B, es similar a los árboles K-D: el espacio se parte en dos subespacios basándose en la comparación con algún elemento de un único dominio. Los árboles K-D-B, así como los árboles-B, son árboles multirramas perfectamente equilibrados, con nodos de tamaño fijo. Cada nodo se almacena como una página por lo que se puede tener un uso eficiente de la memoria secundaria mediante la paginación.

En la sección 1 se define la estructura del árbol K-D-B. Las divisiones de páginas de puntos y páginas regiones se describen en la sección 2. Las reorganizaciones locales que se proponen en este trabajo se presentan en la sección 3. Las secciones 4 y 5 contienen los algoritmos de inserción y extracción que contemplan dichas reorganizaciones. Los resultados del estudio experimental realizado en tres fases: creciente, decreciente y estacionaria [SM87], en los que se pone de manifiesto el aporte de las reorganizaciones locales, se muestran en la sección 6. Por último, la sección 7 contiene las conclusiones y propuestas de investigaciones futuras.

1) ESTRUCTURA DE ARBOL-K-D-B

Existen dos tipos de nodos en el árbol K-D-B:

1. Páginas regiones o nodos internos: que contienen las regiones y los punteros a otras páginas.
2. Páginas de puntos o celdas: que contienen los puntos.

Propiedades

El siguiente conjunto de propiedades definen la estructura del árbol K-D-B.

1. El árbol K-D-B es un árbol multirrama.
2. Ninguna página región contiene regiones con punteros nulos, ni está vacía.
3. La distancia, en páginas, desde la página raíz a cada una de las páginas de puntos es la misma.
4. En cada página región, las regiones de la página son disjuntas y su unión es una región.
5. Si la página raíz es una página región (puede no existir, o si hay sólo una página en el árbol será una página de puntos), la unión de sus regiones es el dominio total.
6. Si en una página región, la página hijo referida por el puntero de una 'región' es una página región, entonces la unión de las regiones en la página hijo es 'región'.
7. Refiriéndose al punto 6, si la página hijo es una página de puntos, entonces todos los puntos de la página están contenidos en 'región'.

2) DIVISIONES

Las divisiones en los árboles-K-D-B parten el espacio en dos subespacios, basándose en la comparación con un elemento de un determinado dominio. Se pueden usar varias técnicas para seleccionar el dominio y el elemento en el domino.

Una de ellas es elegirlo cíclicamente. Existen excepciones para esta técnica debido a que a veces se tiene que dividir una página por un elemento determinado (divisiones descendentes).

En general, dado el dominio divisor, el punto de división se elige de tal forma que, en la medida que sea posible, el número de puntos o regiones en cada nueva página sea el mismo, con ello se minimiza el número de veces que se ha de dividir una página. Sin embargo, en algunos casos el dominio de división debe reelegirse.

2.1 DIVISION DE UNA REGION POR EL ELEMENTO xi

Sea la región $I_1 \times I_2 \times \dots \times I_n$. Si x_i no pertenece a I_i , la región no cambia con la división. En otro caso, sea $I_i = [\text{mini}(i), \text{maxi}(i)]$; dividiendo la región resultan dos nuevas regiones:

1. $I_1 \times \dots \times [\text{mini}(i), x_i] \times \dots \times I_n$
2. $I_1 \times \dots \times [x_i, \text{maxi}(i)] \times \dots \times I_n$

La región (1) se llama región izquierda y la (2) región derecha.

Si x_i no pertenece a I_i , ya que $x_i < \min(i)$, se dice que la región permanece a la derecha de x_i ; si $x_i \geq \max(i)$, la región permanece a la izquierda de x_i . Se dice que un punto $(y_1, y_2, \dots, y_n)$ permanece a la izquierda de x_i si $y_i < x_i$, y permanece a la derecha en otro caso.

2.2 DIVISION DE UNA PAGINA DE PUNTOS POR x_i

La página de puntos se divide por x_i creando dos nuevas páginas de puntos, la página derecha y la página izquierda; entonces, se transferirán todos los puntos a las páginas izquierda o derecha, dependiendo de si permanecen a la izquierda o a la derecha de x_i ; y luego se borrará la página vieja.

2.3 DIVISION DE UNA PAGINA REGION POR x_i

Una página región se divide por x_i creando dos nuevas páginas llamadas también página izquierda y página derecha; llenandolas con regiones derivadas de la página región vieja; y borrando la página vieja.

3) REORGANIZACIONES LOCALES

Como mejora de la ocupación de las páginas de puntos se introdujeron los procedimientos de rotación y recombinaciones.

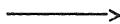
Previamente se define el concepto de regiones unibles: si la unión de dos o más regiones es una región, se dice que las regiones se pueden unir.

3.1 ROTACION

Si al insertar un punto, la página de puntos se sobrecarga, antes de partirla se comprueba si la región que contiene a esos puntos se puede unir con otra región que no este llena, si es así se redistribuyen los puntos de ambas. Este tipo de rotación, en el caso de poderse realizar, evita la creación de una página de puntos y por lo tanto, la posible sobrecarga de páginas regiones.

Por ejemplo, para tamaño máximo de página de puntos, $tmpp = 3$, al insertar el punto x_1 resultará:

A	x		x	x ₁	x		B
			x				
	x					C	x
E							
		x			x	D	
x							x



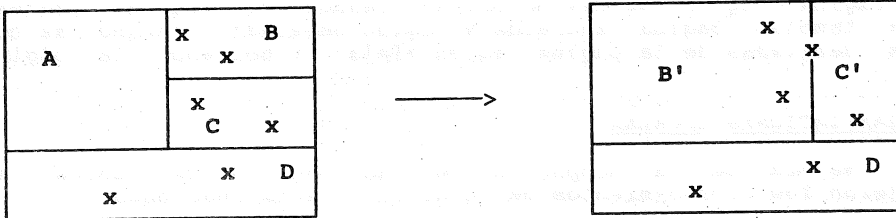
A	x				x		B
			x	x			
	x		E			C	x
				x			
					x	D	
x							x

3.2 RECOMBINACIONES DE PAGINAS VACIAS

La division descendente provocada por la partición de una página región puede originar páginas vacías. Para mejorar la ocupación se borran dichas páginas, debiendo de reestructurar las regiones de la página región padre. Se proponen dos métodos:

3.2.1 RECOMBINACION POR UNION

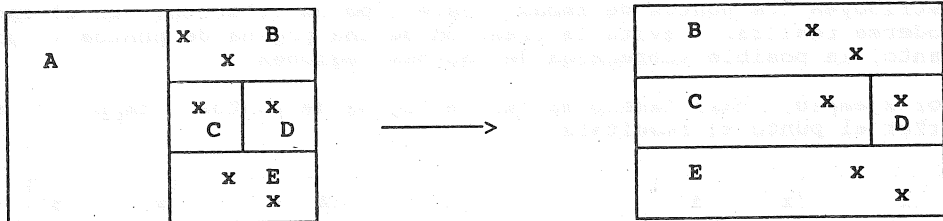
La región que señalaba a una página de puntos vacía se une con otra u otras adyacentes en una dimensión, formando una región mayor, que señalará a una página de puntos que contendrá todos los puntos correspondientes a las páginas implicadas en la unión. Si esta página se sobrecarga se divide como se muestra en la siguiente figura, para $tmp=3$.



Este tipo de recombinação se utiliza sólo en las páginas regiones del último nivel.

3.2.2 RECOMBINACION POR EXTENSION

Se reparte la región entre otra u otras adyacentes, aumentando el tamaño de dichas regiones a lo largo de una dimensión.



Tanto la recombinação por unión como por extensión dan el mismo resultado si se utiliza una única región adyacente.

A	B
x	x
x	C
x	D
	x

A	C
x	x
x	x
x	D
	x

3.2.3 RECOMBINACION DE UNA PAGINA DE PUNTOS VACIA

Sea PP la página de puntos vacía y PR la página región que contiene la región que señala a PP. El algoritmo para borrar PP es el siguiente:

1. Si el número de regiones de PR es 1, borrar PP, poner el número de regiones de PR a 0, ir a RECOMBINAR UNA PAGINA REGION VACIA y terminar.
2. En otro caso, buscar la región 'i' de PR que señale a PP. Buscar una región 'j' que se pueda unir con la 'i'. Si no existe, ir al paso 4.
3. RECOMBINAR POR EXTENSION las regiones 'i' y 'j'. Borrar PP, decrementar en 1 el número de regiones de PR y terminar.
4. Buscar un conjunto de regiones tales que se puedan extender sobre la región 'i'. Si no se encuentran, ir al paso 6.
5. RECOMBINAR POR EXTENSION las regiones seleccionadas, decrementar el número de regiones de PR en 1 y terminar.
6. Unir todas las páginas de puntos a las que señalan todas las regiones de PR en una única página de puntos, y todas las regiones de PR en una sola región (esto siempre se puede hacer, ya que la unión de todas las regiones de una página región forman una región).
7. Dividir la página de puntos hasta que no se produzca una situación de sobrecarga.
8. Terminar.

3.2.4 RECOMBINACION DE UNA PAGINA REGION VACIA

Se PR la página región que está vacía, y PADRE la página región que señala a PR. El algoritmo para borrar PR es el siguiente:

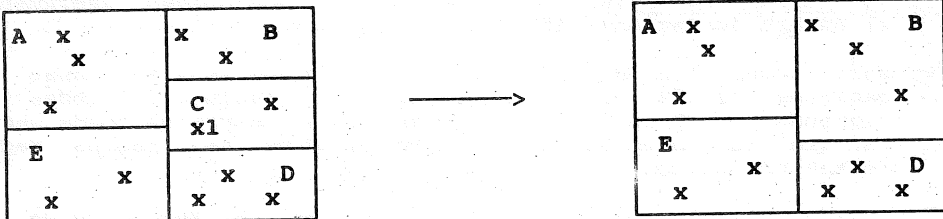
1. Si el número de regiones de PADRE es 1, borrar PR, poner el número de regiones de PADRE a 0 e ir a RECOMBINAR UNA PAGINA REGION VACIA.

2. En otro caso, buscar la región 'i' de PADRE que señala PR. Buscar una región 'j' que se pueda unir con la 'i'. Si no existe, ir al paso 4.
3. RECOMBINAR POR EXTENSION las regiones 'i' y 'j'. Borrar PR, decrementar en 1 el número de regiones de PADRE y terminar.
4. Buscar un conjunto de regiones tales que se puedan extender sobre la región 'i'. Si no es posible ir al paso 6.
5. RECOMBINAR POR EXTENSION las regiones seleccionadas, decrementar el número de regiones de PADRE en 1 y terminar.
6. Unir todos los puntos que estén por debajo de PADRE en una única página, y dejar en cada nivel una página región con una única región. Resolver cualquier situación de sobrecarga.
7. Terminar.

3.3 RECOMBINACION DE PAGINAS DE PUNTOS NO VACIAS

Al extraer un punto, de una página de puntos no vacía, se comprueba si la región que contiene a ese punto se puede unir con otra región, si es así se recombina en una única región si la suma de los puntos de las dos páginas de puntos no excede de un cierto umbral. En realidad, se trata de unir dos páginas de puntos sin que se produzca una sobrecarga (umbral $\leq$ tmpp).

Por ejemplo, para umbral = tmpp = 3, al borrar el punto x1 resultará :



4) INSERCIÓN

Para insertar un punto en un árbol K-D-B, se parte del nodo raíz y se recorre el árbol hasta llegar a la página de puntos donde hay que insertar el punto. Una vez allí se comprueba si dicho punto ya existe, en cuyo caso, no se realiza la inserción.

Una vez realizada la inserción, si la página de puntos queda sobrecargada, se intenta hacer una rotación, si no se realiza la división de dicha página. Esta división origina la división de la región que le señala. Si por esta causa se sobrecarga la página

región, también habrá que dividirla, tal como se vio en 2.3.

IMPLEMENTACION DEL PROCESO DE INSERCIÓN

El algoritmo para insertar un 'punto' es el siguiente:

- I1: Si Raíz es nula, crear una página de puntos conteniendo el 'punto', poner su dirección en raíz y terminar.
- I2: En otro caso, buscar la página de puntos en la que se debe añadir 'punto'. Si 'punto' ya está en la página no insertarlo, dar un mensaje de error y terminar.
- I3: Añadir 'punto' a la página de puntos. Si la página no se sobrecarga, terminar.

Si se sobrecarga:

- I3.1: Si se trata de la página raíz, dividirla en dos nuevas páginas de puntos, crear una página región, con dos regiones que señalen a dichas páginas, que será la nueva raíz del árbol y terminar.
- I3.2: Si no es la raíz, se comprueba si se puede ROTAR, y en tal caso se rota y termina. Si no es posible, se divide la página de puntos, y en la página región PADRE se divide la región que señala a la página de puntos en cuestión.
- I4: Si PADRE se sobrecarga. Se elige un dominio(i) y un elemento xi de este dominio tal que PADRE se divide por xi resultando dos nuevas páginas regiones izquierda y derecha. Si alguna de las dos queda sobrecargada, volver al punto I4.
- I5: Si la página PADRE era la página raíz, ir a (I6). En otro caso, obtener la página que sea la página padre de la página dividida. Reemplazar la región vieja por las dos nuevas regiones (región izquierda y región derecha), que se obtienen dividiendo la región vieja por xi. Si esto causa sobrecarga en una página, repetir desde I4, si no ir a I7.
- I6: Crear una nueva página región conteniendo dos regiones y sus respectivos punteros hacia las nuevas páginas regiones izquierda y derecha, respectivamente. Colocar en esta nueva página región la raíz del árbol.
- I7: Si en el proceso de división se ha quedado alguna página vacía, borrarla, RECOMBINAR las regiones y terminar.

5) EXTRACCION

La estructura del árbol K-D-B propuesta por J. T. Robinson no excluye las páginas de puntos vacías, y no exige nada sobre la utilización del almacenamiento. El algoritmo de extracción que se presenta en este trabajo elimina dichas páginas y mejora la ocupación utilizando el procedimiento de recombinación de páginas no vacías.

IMPLEMENTACION DEL PROCESO DE EXTRACCION

El algoritmo para extraer un 'punto' del árbol es el siguiente:

- E1. Obtener la página de puntos donde debe de estar el punto a extraer. Si no existe, dar un mensaje de error y terminar.
- E2. Extraer el punto de la página de puntos.
- E3. Si la página de puntos queda VACIA, eliminarla, RECOMBINAR y terminar. Si es NO VACIA intentar una RECOMBINACION con otra página de puntos, si la suma de los puntos no supera un cierto umbral y terminar.

6) DESCRIPCION DEL ESTUDIO EXPERIMENTAL

El rendimiento de una estructura de datos esta determinado por dos criterios: el tiempo de procesamiento y la utilización de memoria.

Se han realizado varias simulaciones para estudiar el rendimiento del árbol K-D-B, midiendo estos criterios en base a los siguientes parámetros:

- a) número de páginas regiones totales.
- b) número de páginas de puntos totales.
- c) número de páginas regiones accedidas.
- d) número de páginas de puntos accedidas.
- e) ocupación de las páginas regiones.
- f) ocupación de las páginas de puntos.
- g) altura del árbol.

De acuerdo con el caracter dinámico de la estructura el estudio se ha realizado en las siguientes fases:

- 1) Estado creciente. Inserciones repetidas.
- 2) Estado estacionario. Inserciones y extracciones manteniendo aproximadamente constante el número de puntos.
- 3) Estado decreciente. Extracciones repetidas.

Todos los números aleatorios utilizados en los experimentos se han obtenido a partir de un generador de números pseudoaleatorios de distribución continua uniforme en $[0,1)$. Para cada punto se generaron nd valores aleatorios, siendo nd la dimensión del espacio.

Una cuestión previa es el estudio comparativo de la estructura para los dos tipos de re combinaciones de páginas de puntos vacías: por unión o por extensión. Se construyeron dos árboles de dimensión 2, tamaño máximo de las páginas de puntos, $tmpp=20$, y tamaño máximo de las páginas regiones, $tmpr=10$; obteniéndose resultados similares. Para realizar el estudio experimental se eligió la re combinación por extensión porque es más rápida, al no tener que comprobar la condición de que la unión de todas las regiones formen una región, y en ningún caso provoca sobrecarga.

ESTADO CRECIENTE

El estado creciente corresponde a la fase de crecimiento del árbol a través de sucesivas inserciones.

Se construyeron árboles de 10000 puntos. Los parámetros estadísticos anteriormente descritos, se midieron en intervalos regulares de 100 inserciones.

Inicialmente se comprueba el efecto que produce la introducción del procedimiento de rotación, para ello se construyen dos árboles con $nd=4$, $tmpp=25$ y $tmpr=13$.

- El número de páginas regiones y páginas de puntos en el árbol construido CON ROTACIONES fue de 65 y 486, respectivamente. Y en el que se construyó SIN ROTACIONES fue de 113 y 748, respectivamente.

- La ocupación de las páginas de puntos presenta un notable incremento, figura 1. El efecto que produce la rotación en las páginas regiones es que pospone la partición de dichas páginas, esto da lugar a que la figura 2 presente intervalos en los que la diferencia es notable.

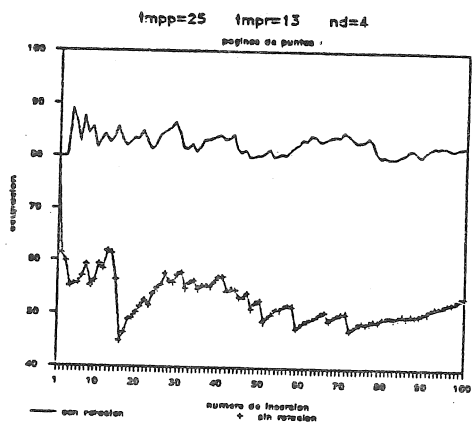


Figura 1

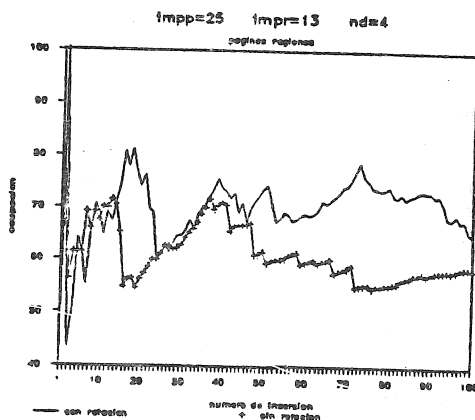


Figura 2

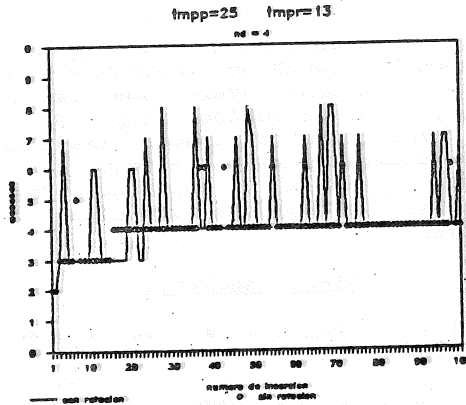


Figura 3

- El aumento del número de accesos es debido a que la ocupación es mucho mayor y las sobrecargas son por consiguiente más frecuentes. Estas sobrecargas se pueden resolver mediante rotaciones o particiones. Una rotación requiere tres accesos y una partición de una página de puntos que no provoca sobrecarga en una página región da lugar a dos accesos. En la práctica existen intentos fallidos de rotación, que conllevan uno o ningún acceso, figura 3.

Debido a estos resultados, todos los experimentos presentados a continuación se llevaron a cabo utilizando la rotación.

Se construyeron 12 árboles, teniendo en cuenta la variación de la dimensionalidad, nd , el tamaño máximo de las páginas de puntos, y el tamaño máximo de las páginas regiones, para los siguientes valores:

$$nd = 2, 4, 6$$

$$tmpp-tmpr = 10-5, 15-8, 20-10, 25-13$$

obteniéndose que:

- La ocupación de las páginas de puntos aumentaba al aumentar $tmpp$ y $tmpr$, estando en todos los casos entre un 65% y un 85 %, como se puede observar en las figuras 4 y 5, en las que cada curva representa una dimensión.

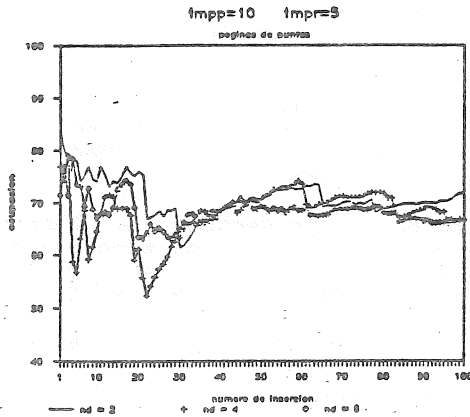


Figura 4

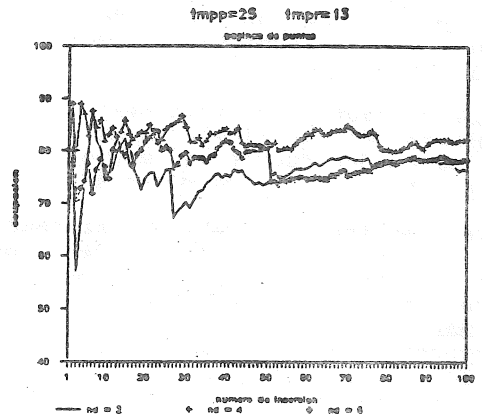


Figura 5

- La ocupación de las páginas regiones se mantiene alrededor de un 70%, el incremento de tmpp y tmpr provoca una mayor variabilidad, como se observa en las figuras 6 y 7.

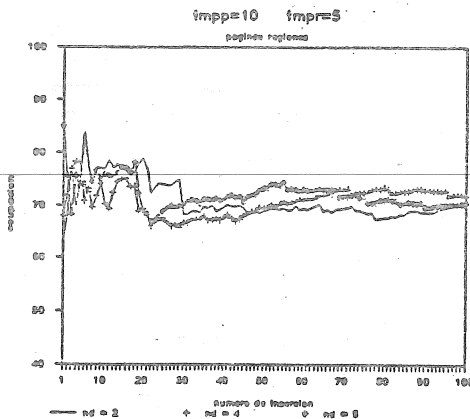


Figura 6

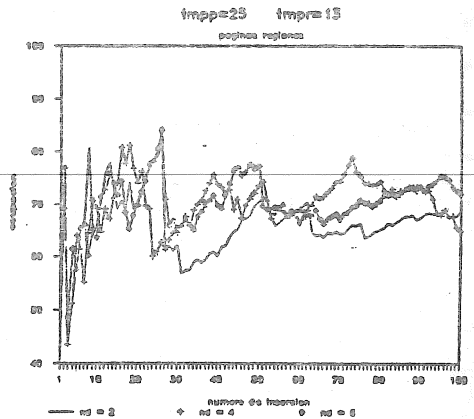


Figura 7

- La dimensionalidad apenas influye en la ocupación, manteniéndose en la banda antes mencionada, dependiendo de tmpp y tmpr.

ESTADO ESTACIONARIO

El estado estacionario corresponde a la fase de actualización del árbol, a través de la selección de inserciones y extracciones de forma aleatoria.

A partir de un árbol de 10000 puntos construido en el estado creciente, se insertaron 2000 nuevos puntos y se extrajeron 2000 de los ya existentes.

Los parámetros anteriormente descritos, se midieron en intervalos regulares de 100 inserciones + extracciones. Las reorganizaciones locales usadas fueron las mismas que en el estado creciente, añadiendo las recombinaciones de páginas de puntos no vacías.

Se realizaron pruebas manteniendo fijos los parámetros nd , $tmpp$ y $tmpr$, y variando el umbral de recombinación para ver los efectos de este parámetro en la ocupación. Los valores dados a dicho umbral fueron los siguientes: 100%, 80%, 60% y 0% del tamaño máximo de las páginas de puntos. Como se observa en las figuras 8 y 9, la disminución del umbral de recombinación produce un efecto negativo, más acusado para umbrales altos, en el porcentaje de ocupación de las páginas de puntos y opuesto en las páginas regiones.

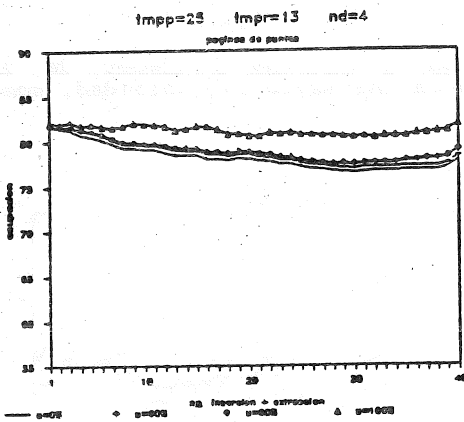


Figura 8

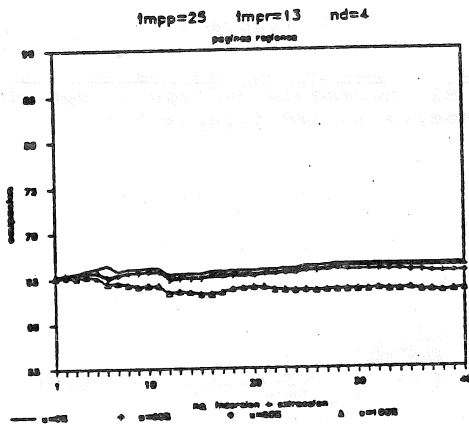


Figura 9

- La diferencia de accesos para umbrales del 100% y del 0% se debe al uso de las recombinaciones de páginas de puntos no vacías. Dichas recombinaciones pueden dar lugar a: 2 accesos en el caso de que se realice la recombinación, 1 acceso si la suma de puntos supera el umbral y ningún acceso en el caso de que no exista la región adyacente, figura 10.

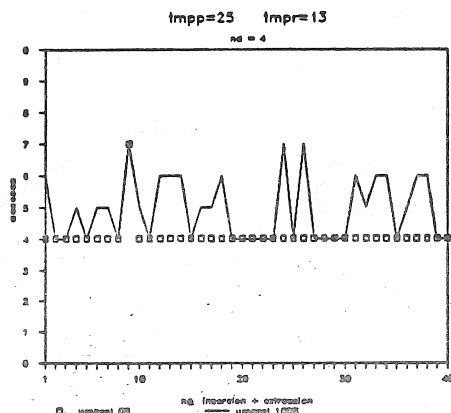


Figura 10

La variación de la dimensión no presenta diferencias significativas.

ESTADO DECRECIENTE

El estado decreciente corresponde a la fase de disminución del árbol a través de sucesivas extracciones.

A partir del árbol final del estado creciente se realizaron las extracciones de los 10000 puntos, midiendo los mismos parámetros y en iguales intervalos que en el estado creciente. Al igual que en el estado estacionario, se realizaron pruebas para distintos umbrales de recombinación de páginas de puntos no vacías, obteniéndose los resultados que se muestran en las figuras 11, 12 y 13.

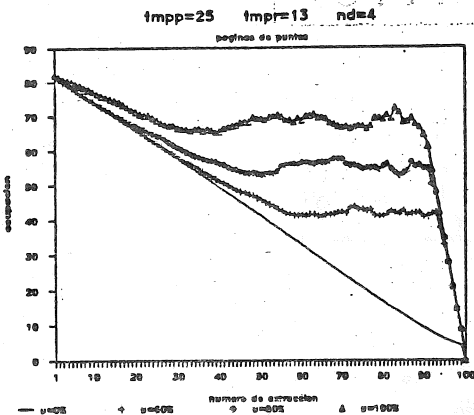


Figura 11

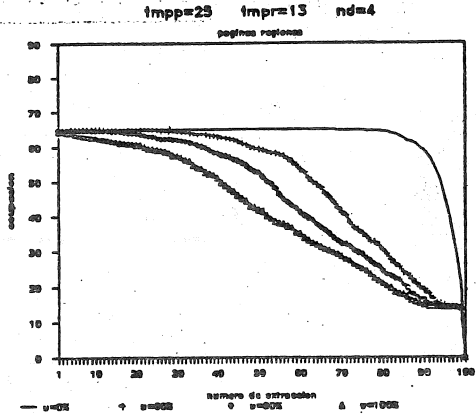


Figura 12

- A umbrales altos la ocupación de páginas de puntos presenta una suave caída inicial y un mantenimiento posterior a lo largo del proceso. Tan solo en la extracción de los últimos puntos aparece un descenso brusco. Esto hace pensar en lo adecuado del sistema que permite mantener un alto nivel de ocupación hasta alcanzar las últimas extracciones. La ocupación de las páginas regiones presenta un comportamiento contrario a la de las páginas de puntos, donde la caída brusca final se pone de manifiesto en la ocupación de las páginas regiones cuando el umbral de las páginas de puntos es del 0%.

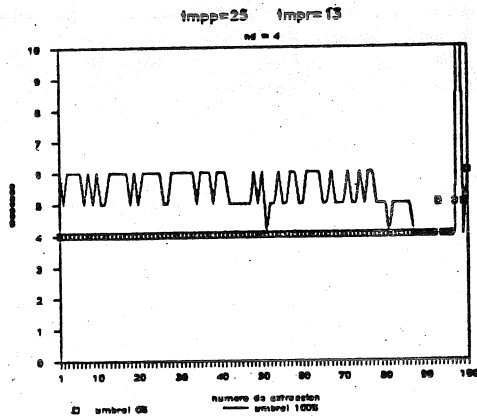


Figura 13

- Los accesos que se muestran en la figura 13, para umbral 100%, están provocados por la recombinación de páginas de puntos no vacías, salvo en las últimas extracciones.

La variación de la dimensión no presenta diferencias

significativas.

CONCLUSIONES

El esquema de mantenimiento que se propone para el árbol K-D-B que describió J. T. Robinson, elimina páginas de puntos vacías y la resolución de sobrecargas se realiza con: particiones, tanto de páginas regiones como de páginas de puntos; rotaciones, sólo en estas últimas. Se incorpora la operación de recombinación para páginas de puntos no vacías.

El presente estudio experimental en la fase creciente introduce las rotaciones, como alternativa a las particiones, y recombinaciones de páginas vacías consiguiendo mejorar la ocupación en función del crecimiento del tamaño máximo de las páginas de puntos y de las páginas regiones, llegando a un 85% y un 70% respectivamente, para $tmpp=25$ y $tmpr=13$.

La extracción sucesiva de puntos conduce a un descenso en la ocupación de páginas de puntos, para evitarlo se propone la recombinación de páginas de puntos no vacías.

En la fase estacionaria la combinación de rotaciones y recombinaciones para umbral 100% consigue mantener la ocupación.

La fase decreciente muestra que la ocupación presenta un suave descenso inicial y un mantenimiento posterior, hasta casi las últimas extracciones, si se utilizan recombinaciones a umbrales altos.

Las operaciones de eliminación de páginas de puntos vacías, la rotación como alternativa a la partición y la recombinación de páginas de puntos no vacías, si bien aumentan la ocupación, se ve incrementado el número de accesos.

Nuestra línea de investigación continúa estudiando la respuesta del árbol K-D-B a los distintos tipos de interrogaciones: parcial, en rango y los m-vecinos más próximos.

REFERENCIAS

- [BE75] BENTLEY, J.L.
Multidimensional binary search trees used for associative searching.
Comm. ACM 18, 9 (1975), 509-517.
- [BE79] BENTLEY, J.L.
Multidimensional binary search trees in database applications.
IEEE Trans. Software Eng, SE-5, 4 (1979), 333-340.
- [BF79] BENTLEY, J.L. AND FRIEDMAN, J.H.
Data structures for range searching.
Computing Surveys 11, 4 (1979), 397-409.
- [BM72] BAYER, R. AND MCCREIGHT, E.
Organization and maintenance of large ordered indexes.
Acta informática 1, 3 (1972), 173-189.
- [RO81] ROBINSON, J.T.
The K-D-B-Tree: A search structure for large multidimensional dynamic indexes.
Department of Computer Science, Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213, 1981.
- [SM87] SANTANA O., MAYOR O., DIAZ M., LOPEZ G.
Comportamiento del árbol-BD en las fases creciente, decreciente y estacionaria.
E.U.Informática, Universidad Politécnica de Canarias.
España 1987.