

# Optimizando Desorden Acotado

## Optimizing Bounded Disorder

Ricardo A. Baeza-Yates<sup>1</sup> y Héctor Soza Pollman<sup>2</sup>

<sup>1</sup>Departamento de Cs. de la Computación, Universidad de Chile  
Blanco Escalada 2120, Santiago de Chile

<sup>2</sup>Departamento de Ingeniería de Sistemas de Computación, Universidad del Norte  
Av. Angamos 0610, Casilla 1280, Antofagasta, Chile  
e-mail : rbaeza@dcc.uchile.cl, hsoza@ucn.cl

*Artículo recibido en abril 09, 2001; aceptado en abril 18, 2002*

## Resumen

*En este trabajo presentamos, para la organización de archivo Desorden Acotado (Bounded Disorder), la determinación de un modelo que relaciona el tamaño del bucket de overflow secundario (medido en cantidad de registros que este bucket puede almacenar) con el tamaño de los buckets primarios del nodo de datos (supuesto distinto al de overflow) y con la cantidad de estos buckets, considerando el efecto del índice. Este resultado se obtiene al minimizar el costo de inserción de datos en la estructura, y permite determinar el tamaño óptimo del bucket de overflow. Se considera el caso sin y con expansiones parciales y se propone un modelo para la relación óptima de los parámetros involucrados.*

**Palabras claves:** Árboles B, hashing.

## Abstract

*In this paper we present for the Bounded Disorder file organization, a model that relates the secondary overflow bucket size to the primary bucket size of the data nodes (supposedly different from that of overflow) and the whole size of a data node, by considering the index. This result is obtained by minimizing the cost of inserting data into the structure and enables to compute the optimal overflow bucket size. Both, the normal and partial expansion cases are considered and a model for the optimal relation of the parameters involved is developed.*

**Keywords:** B-trees, hashing.

## 1 Introducción

La organización de archivo de Desorden Acotado, Bounded Disorder (BD) en inglés, que fuera propuesta originalmente por Litwin y Lomet (1987), se considera una estructura que provee un acceso eficiente a los datos almacenados, debido a que reduce la altura del índice integrando el acceso a ellos a través del uso de hashing (Tharp y Boswell, 1991). Un archivo BD consiste de un índice de árbol B (*B-tree*), en que cada nodo de datos está organizado como una tabla de hashing de  $m$  buckets ( $m > 0$ ) con una capacidad de almacenar  $b$  registros (o claves) cada uno y un bucket de overflow adicional, que se supondrá con una capacidad de  $c$  registros. Esta estructura se muestra en la Figura 1 y combina las ventajas del acceso rápido de hashing con la posibilidad de realizar acceso secuencial ordenado de los árboles B.

Para insertar una nueva clave se usa el índice del árbol-B para hallar el nodo de datos que corresponde, y se usa una función de hashing para determinar el bucket adecuado e insertarlo. Si éste está lleno se intenta su inserción en el bucket de overflow, y si éste también está lleno se divide el nodo de datos en dos nodos de datos, lo que se llama una división (*split*) (Litwin y Lomet, 1987). Un primer análisis de la eficiencia de esta estructura lo realiza Lomet (1988), sobre una variante de esta estructura, el cual es mejorado por Ramakrishna y Mukhopadhyay (1994) considerando un método alternativo para el crecimiento de los datos del archivo, pero sólo considerando la conducta de un nodo de datos. Matsliach (1990) considera el análisis de las estructuras de datos de memoria secundaria que usan multibuckets. Baeza-Yates (1996) completa el trabajo anterior introduciendo el índice en

\*Este trabajo ha sido parcialmente financiado por el Proyecto FONDECYT No. 1990627.

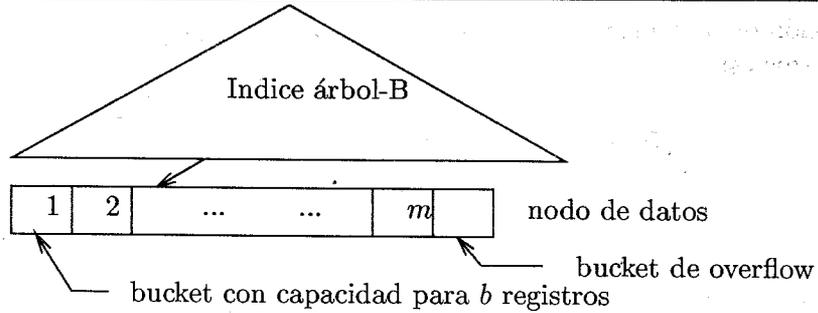


Figura 1: Ejemplo de Desorden Acotado.

el modelo, y usando un análisis del borde del árbol asociado al índice (*fringe analysis* (Eisenbarth *et al.*, 1982)). En todos estos casos se ha considerado que los datos se almacenan en los nodos hojas de un árbol-B, formados por  $m+1$  buckets primarios contiguos, de los que el último es el bucket de overflow, todos del mismo tamaño ( $b$  registros por bucket). En este artículo presentamos un estudio para determinar una relación óptima entre el tamaño del bucket de overflow  $c$ , el del bucket primario  $b$  y la cantidad  $m$  de buckets primarios minimizando el costo de inserción de datos en la estructura. En la estructura original  $c = b$ . Se considera que la unidad de espacio es la necesaria para almacenar la clave (llave) primaria de la base de datos más la información adicional asociada a cada elemento. En la Sección 2 se describen brevemente los resultados más importantes del método BD. En la Sección 3 se describe el análisis desarrollado y sus resultados para nodos sin expansiones y nodos con una sola expansión. En la Sección 4 se presenta un modelo experimental que relaciona los parámetros involucrados en el punto óptimo. Finalmente en la Sección 5 se presentan las conclusiones.

## 2 Desorden Acotado

En lo que sigue se usará la misma notación de Baeza-Yates (1996) y resumimos el análisis del artículo referido que es la base de nuestros resultados.

### 2.1 Nodos sin Expansión

Sea  $2L$  la menor cantidad de registros en un nodo de datos en el momento en que una división pue-

de ocurrir y  $H \leq 2L$  la capacidad máxima de un nodo. Luego se tienen nodos de datos que tienen entre  $L$  y  $H$  claves. Para la versión actual de BD se observa que  $L = \frac{b+c}{2}$  y  $H = mb + c$ . Sea  $s_j$  la probabilidad de una división dado que hay  $j$  claves en un nodo de datos, de modo que  $s_j = 0$  para  $j < 2L$  o  $j > H$ , y  $s_j > 0$  para  $j \geq 2L$  y  $j \leq H$ . Además  $s_H = 1$ . Sea  $p_j(n)$  la probabilidad de hallar una clave en un nodo que contiene  $j$  claves, cuando hay  $n$  claves insertadas en el archivo. Asintóticamente sobre  $n$ , estas probabilidades  $p_j$  convergen a la solución del siguiente sistema de ecuaciones lineales, en que  $p_L(L) = 1$  y  $p_j(L) = 0$  para  $j \neq L$ :

$$(j+1)p_j = j(1-s_{j-1})p_{j-1} + js_{2j-2}p_{2j-2} + 2js_{2j-1}p_{2j-1} + js_{2j}p_{2j}$$

con  $j = L, \dots, H$  y  $p_j = 0$  para  $j < L$  o  $j > H$  (ver Baeza-Yates (1996)). Entonces, la probabilidad de insertar una clave en un bucket de overflow es:

$$Pr_{Ov}(n) = \sum_{j=2L}^H p_j f_j^{m,b,c}$$

Las funciones  $f_j^{m,b,c}$  son las probabilidades de insertar una clave en un bucket de un nodo de datos que está lleno. Estas probabilidades están determinadas en Ramakrishna y Mukhopadhyay (1994) y han sido modificadas considerando un tamaño  $c$  para el bucket de overflow, además de los parámetros  $j$ ,  $m$  y  $b$  ya conocidos. La probabilidad de tener una división durante una inserción es:

$$Pr_{Split}(n) = \sum_{j=2L}^H s_j p_j$$

y la utilización de almacenamiento esperada en los nodos de datos es:

$$U(n) = (H \sum_{j=L}^H p_j/j)^{-1}$$

## 2.2 Nodos con una Expansión Parcial

Si se usa una expansión, los nodos de datos tienen tamaños  $H$  y  $3H/2$  respectivamente. El valor de  $L$  es el mismo anterior. Sea  $p_j$  la probabilidad asintótica de insertar una nueva clave en un nodo de datos de tamaño  $H$  que contiene  $j$  claves. Sea  $q_j$  la probabilidad asintótica de insertar una nueva clave en un nodo de datos de tamaño  $3H/2$  que contiene  $j$  claves. Entonces,  $p_j = 0$  para  $j \leq L$  o para  $j > H$ , y  $q_j \neq 0$  para  $j = 2L + 1, \dots, 3H/2$ . Sea  $t_j$  la probabilidad de una expansión parcial dada una inserción en un nodo de datos de tamaño  $H$  y sea  $s_j$  la probabilidad de una división dada una inserción en un nodo de datos de tamaño  $3H/2$  (luego  $s_{3H/2} = 1$ ). Ya que el más pequeño  $j$  tal que  $t_j \neq 0$  es  $2L$ , tenemos que  $2L + 1$  es la menor ocupancia de un nodo de tamaño  $3H/2$ . Como en general  $s_{2L+1} \neq 0$ , la menor ocupancia para nodos de tamaño  $H$  es  $L + 1$ . Entonces, las probabilidades asintóticas mencionadas están dadas por el siguiente sistema de ecuaciones:

$$(j+1)p_j = j(1-t_{j-1})p_{j-1} + js_{2j-2}q_{2j-2} + 2js_{2j-1}q_{2j-1} + js_{2j}q_{2j}$$

para  $j = L + 1, \dots, H$

$$(j+1)q_j = j(1-s_{j-1})q_{j-1} + jt_{j-1}p_{j-1}$$

para  $j = 2L + 1, \dots, 3H/2$ , con  $p_j = 0$  para  $j \leq L$  o  $j > H$ , y  $q_j = 0$  para  $j \leq L$  o  $j > 3H/2$ . La primera ecuación considera el efecto de la división de un nodo de datos de tamaño  $3H/2$  en nodos de datos de tamaño  $H$ . La segunda ecuación considera el efecto de la expansión de un nodo de datos de tamaño  $H$  en uno de tamaño  $3H/2$ . En general  $t_{H+1} \neq 0$  si  $H \geq 2b$  (que es el caso para BD).

A partir de este sistema podemos expresar  $q_j$  como función de  $p_j$ , o viceversa:

$$q_j = \frac{1}{j+1} \sum_{i=2L}^{j-1} (i+1)t_i p_i \prod_{k=i+1}^{j-1} (1-s_k)$$

También, para  $j > 3H/4$  se tiene:

$$p_j = \frac{3H}{4(j+1)} \prod_{k=3H/4}^{j-1} (1-t_k) p_{3H/4}$$

Reemplazando estas relaciones en el sistema de ecuaciones se obtiene un sistema lineal denso con  $3H/4 - L + 1$  ecuaciones que resolver.

En este caso redefinimos las probabilidades  $Pr_{Ov}$ ,  $Pr_{Split}$  y definimos la probabilidad de una expansión parcial de un nodo  $Pr_{Exp}$ . Estas son:

$$Pr_{Ov}(n) = \sum_{j=L}^H p_j(n) f_j^{m,b,c} + \sum_{j=2L+1}^{3H/2} q_j(n) f_j^{m,3b/2,3c/2}$$

$$Pr_{Split}(n) = \sum_{j=2L+1}^{3H/2} s_j q_j(n)$$

$$Pr_{Exp}(n) = \sum_{j=L}^H t_j p_j(n)$$

De un modo similar se puede redefinir  $U(n)$ . En la siguiente sección se muestran los resultados asociados al cálculo del costo de inserción para el caso sin expansiones y de una expansión parcial y la optimización de  $c$  en función de  $b$ .

## 3 Optimizando la Estructura

Se desea determinar una relación óptima entre el tamaño del bucket primario  $b$ , el del bucket de overflow  $c$  y la cantidad de buckets primarios  $m$ . Al intentar minimizar la utilización de almacenamiento de la estructura se observó que  $c/b \rightarrow \infty$ , lo cual conduce a que se alcance para la utilización el valor de  $U(n) = \ln(2) = 0.6931$ , como en los árboles  $B^+$ , pero no se determina ninguna relación óptima para los parámetros mencionados. Se consideró también minimizar el tiempo de búsqueda de un dato en la estructura, pero no se alcanza un óptimo debido a que los valores de esta medida son siempre crecientes para los distintos parámetros considerados. Si se obtiene una relación  $c/b$  óptima al considerar el costo de inserción, como función de los parámetros  $b$ ,  $c$  y  $m$ , como se describe a continuación.

En lo que sigue supondremos que:

- el nodo de datos entero puede leerse y escribirse en un acceso (es decir, hay contigüidad física).
- el índice del archivo BD está en memoria principal ya que en general representa menos del 1% de los datos (por ejemplo, tres niveles de un árbol B con a lo más 100 hijos por nodo ocupa alrededor de 3Mb y permite almacenar sobre 200 millones de registros usando  $m = b = 10$ ).
- el tiempo de transferencia, dado por  $tt(\text{registros}) = \frac{\text{registros}}{R}$  se expresa en unidades de tiempo para acceder a un bucket. Si  $A$  es el acceso a un bucket (ms),  $RT$  la razón de transferencia (MB) y  $TM$  el tamaño del registro (KB), entonces:  $R = \frac{A \cdot RT}{TM}$ . Se verifica que  $10 \leq R \leq 160$  para casos reales (ver Baeza-Yates (1996)).

### 3.1 Costo de Inserción para Nodos sin Expansión

El costo de inserción  $I_c(n)$  es el número de accesos a un nodo de datos durante una inserción, y se define como:

$$I_c(n) = 2(1 + tt(b)) + Pr_{Ov}(n)(1 + tt(c)) + Pr_{Split}(n)(2 + 3tt(mb + c))$$

El primer término es el número normal de accesos, uno para leer un bucket y otro para escribir un nuevo bucket. Si el bucket está lleno, se necesita acceder el bucket de overflow, por lo cual se tiene el segundo término. El último término es el resultado de considerar la división del nodo de datos cuando el bucket de overflow se llena, en cuyo caso se necesitan dos accesos más para leer el nodo de datos entero y otro para escribir en los nuevos nodos de datos.

Esta expresión tiene un mínimo ya que el tiempo de transferencia aumenta con  $b$  al mismo tiempo que las probabilidades de overflow o división disminuyen. Para los cálculos se consideró el valor  $R = 10$ . A medida que este valor crece disminuye la posibilidad de alcanzar un mínimo para  $I_c(n)$ . La Figura 2 muestra un ejemplo del mínimo para  $I_c(n)$ , considerando  $b = 5$  (a la izquierda) y  $b = 10$

a la derecha. La Figura 3 (izquierda) muestra la relación entre  $I_c(n)$  y  $m$  para valores de  $b = 5, 10, 15$  y la Figura 3 (derecha) muestra la relación entre  $I_c(n)$  y  $b$  para  $m = 5, 10, 15$ . Es importante observar que, debido a que la resolución exacta de los sistemas de ecuaciones lineales mencionados en la Sección 2 depende del tamaño de éste, a medida que el valor de  $m$  o  $b$  crecían eran menos los valores de  $I_c(n)$  calculables. Por ejemplo para  $b = 5$  se alcanza hasta  $m = 41$  pero para  $b = 15$  sólo se puede calcular hasta  $m = 20$ . Análogamente para el caso en que  $m$  está fijo. En la Figura 3 (izquierda) se observa que los valores óptimos de  $I_c(n)$  crecen con  $m$  para valores fijos de  $b$ , y se observa una convergencia hacia un valor fijo de  $I_c(n)$  para  $m$  grande. En la Figura 3 (derecha) en cambio, se observa que hay un crecimiento definido (casi lineal) de  $I_c(n)$  con respecto a  $b$  para valores fijos de  $m$  y todos en el mismo rango de valores, de lo cual se deduce que para un valor de  $R$  dado, e independiente del parámetro  $m$ , los valores de  $I_c(n)$  son crecientes e iguales para los valores de  $b$  considerados. A partir de los resultados obtenidos, fijando el parámetro  $b$  se obtuvo la relación entre  $c$  y  $m$  en Figura 4 (izquierda) y fijando el parámetro  $m$  se obtuvo la relación entre  $c$  y  $b$  como se ve en Figura 4 (derecha). Se observa que en estas figuras aparecen discontinuidades crecientes a medida que  $m$  crece (ver Figura 4 izquierda) y menores a medida que  $b$  crece (ver Figura 4 derecha).

### 3.2 Costo de Inserción para Nodos con Expansiones

Con las mismas consideraciones de la subsección anterior, podemos definir el costo de inserción  $I_c(n)$  a un nodo de datos considerando una expansión parcial como:

$$I_c(n) = 2(1 + tt(b)) + Pr_{Ov}(1 + tt(c)) + Pr_{Exp}(1 + tt(5(mb + c)/2)) + Pr_{Split}(2 + tt(7(mb + c)/2))$$

El primer término es el caso normal y el segundo es el acceso extra si el bucket inicial está lleno. El tercer término es el acceso extra para leer el nodo a expandir y el tiempo de transferencia asociado a

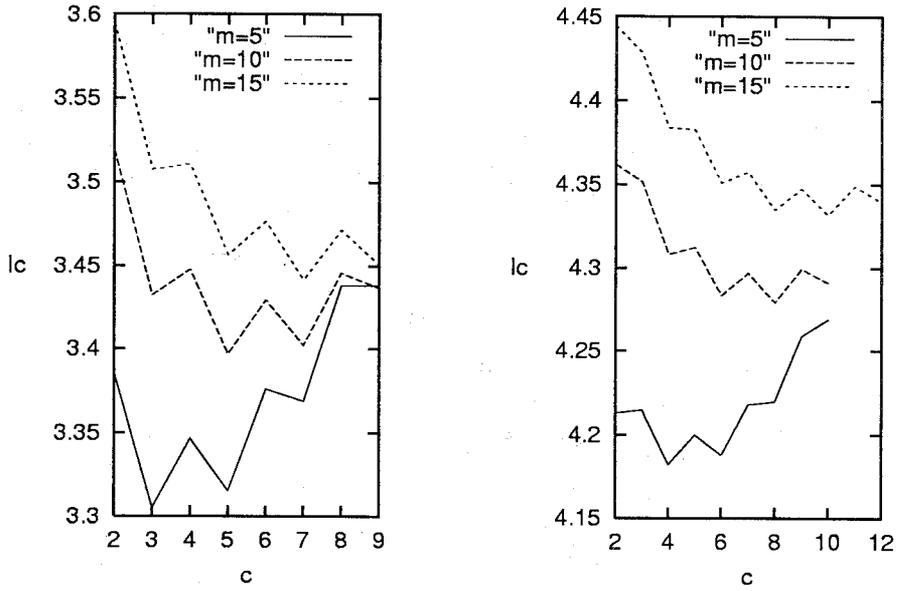


Figura 2:  $I_c(n)$  versus  $c$  con  $R = 10$  y  $b = 5$  (izq.) y  $b = 10$  (der.).

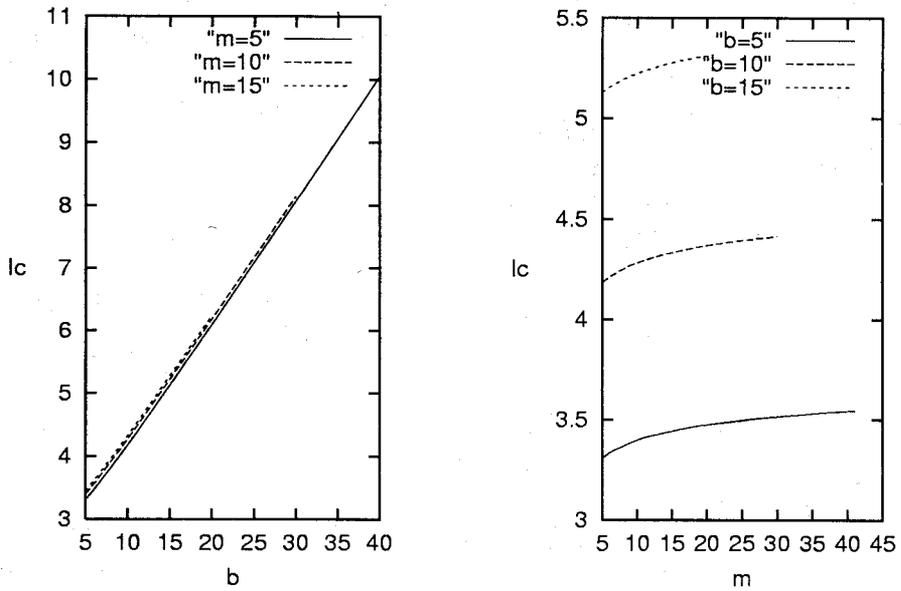


Figura 3:  $I_c(n)$  óptimo versus  $b$  con  $m$  fijo (izq.) e  $I_c(n)$  óptimo versus  $m$  con  $b$  fijo (der.).

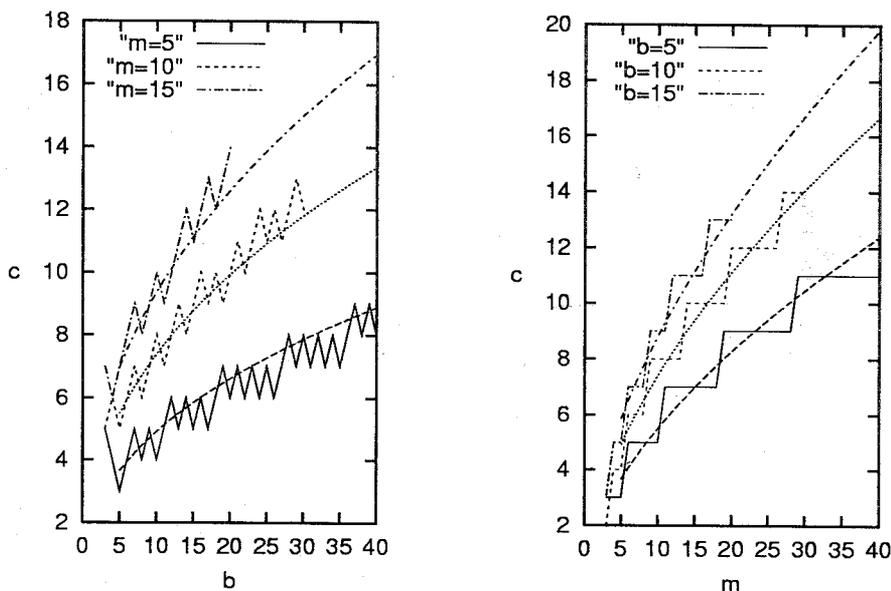


Figura 4:  $c$  óptimo versus  $b$  con  $m$  fijo (izq.) y  $c$  óptimo versus  $m$  con  $b$  fijo (der.).

escribir el nodo completo. El último término considera los accesos extra para leer el nodo expandido completo y escribir uno de los nuevos nodos sin expandir, agregando el tiempo de transferencia asociado a la división.

Se verifica que  $I_c(n)$  alcanza un mínimo, como se ve en la Figura 5, en que se grafica la relación entre  $I_c(n)$  y  $c$  para valores de  $b = 4$  y  $b = 10$ , considerando  $R = 10$ . Al igual que en el caso sin expansión, la resolución de los sistemas de ecuaciones lineales consideradas dependen de los valores de los parámetros  $m$ ,  $b$  y  $c$ , y a medida que estos crecen son menos los valores de  $I_c(n)$  calculables, como se observa en la Figura 5 y siguientes.

Las Figuras 6 muestran la relación de  $I_c(n)$  con  $b$  para valores de  $m = 5, 10, 15$  (a la izquierda) y de  $I_c(n)$  con  $m$  para valores de  $b = 4, 10, 16$  (a la derecha), ambas para  $R = 10$ . Se observa que la primera relación coincide para los distintos valores de  $m$  considerados, y que es casi una línea recta. En cambio, la relación entre  $I_c(n)$  y  $m$  representan curvas descendentes para los distintos valores de  $b$  considerados.

Las Figuras 7 muestran la relación de  $c$  con  $b$  (a la izquierda para valores de  $m = 5, 10, 15$ ) y de  $c$  con  $m$  (a la derecha para valores de  $b = 4, 10, 16$ ).

## 4 Modelo Experimental

Se encontró que un modelo que se ajustaba bien para los datos obtenidos, que relaciona  $c$  con  $m$  y  $b$ , es de la forma

$$c = \gamma m^\alpha b^\beta$$

Los parámetros del modelo en cada caso se obtuvieron usando mínimos cuadrados y considerando todos los datos obtenidos. Los valores de los parámetros cuando  $R = 10$  se presentan en la Tabla 1.

Caso	$\alpha$	$\beta$	$\gamma$
Sin expansión	0.5878	0.4133	0.7337
Con una expansión	0.8392	0.4748	1.3793

Tabla 1: Parámetros del modelo.

Los modelos mencionados se presentan en las Figuras 4 y 7 considerando un rango de valores de  $b$  entre 5 y 40 para la relación entre  $c$  y  $b$  con  $m$  fijo (izquierda), y con valores de  $m$  entre 5 y 40 para la relación entre  $c$  y  $m$  con  $b$  fijo (derecha) para los casos con y sin expansión respectivamente.

Aunque hemos analizado el caso general, en la práctica  $c$  no puede ser cualquier valor debido a la

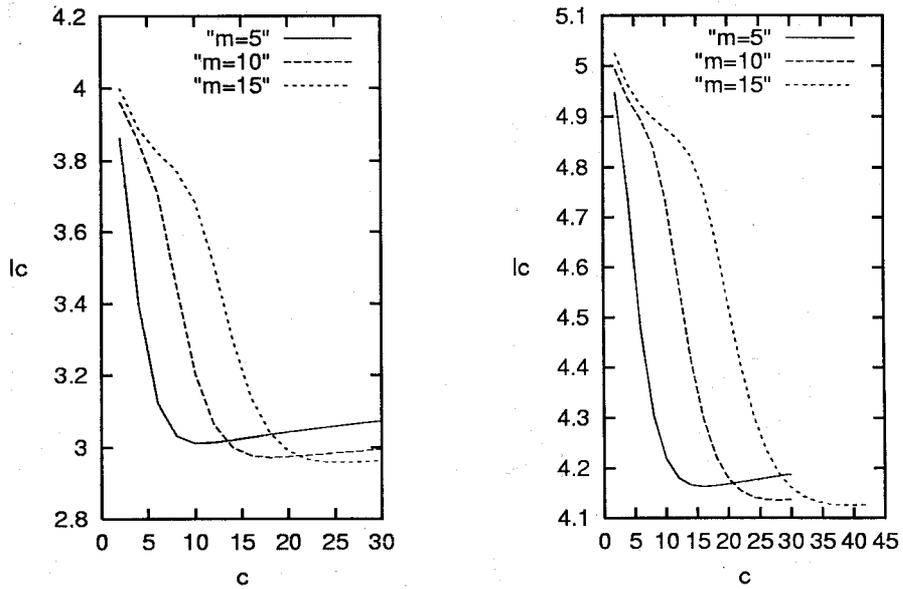


Figura 5:  $I_c(n)$  versus  $c$  con  $R = 10$  y  $b = 4$  (izq.) y  $b = 10$  (der.).

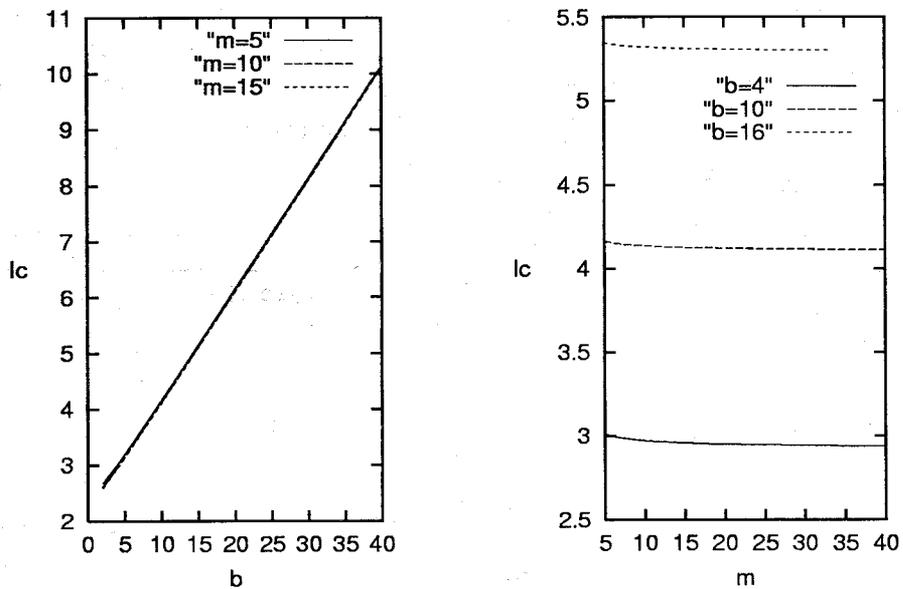


Figura 6:  $I_c(n)$  óptimo versus  $b$  con  $m$  fijo (izq.) e  $I_c(n)$  óptimo versus  $m$  con  $b$  fijo (der.).

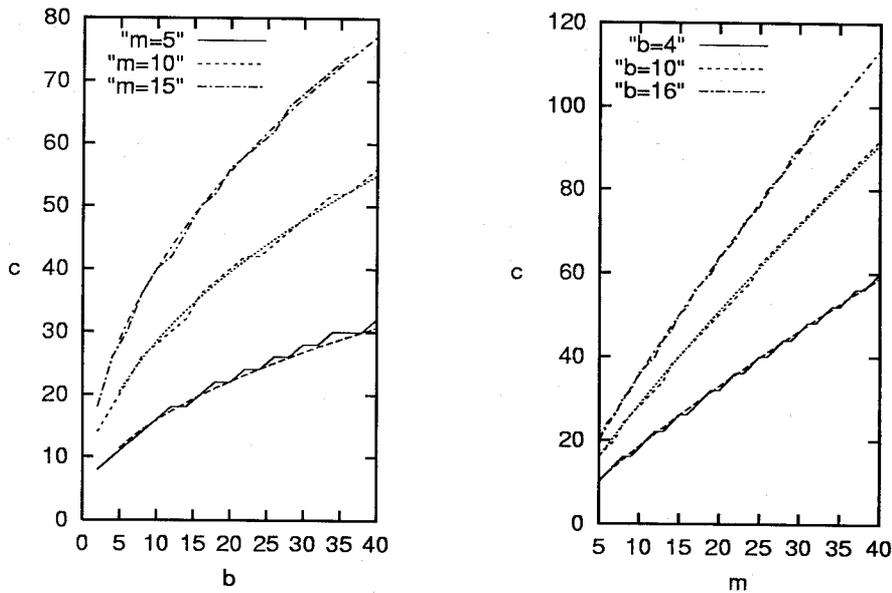


Figura 7:  $c$  óptimo versus  $b$  con  $m$  fijo (izq.) y  $c$  óptimo versus  $m$  con  $b$  fijo (der.).

granularidad de la memoria secundaria asociada a los buckets. Por esa razón el valor de  $c$  debiera ser un múltiplo entero de  $b$ . Si  $c = k b$ , reemplazando en nuestro modelo y despejando obtenemos  $m = \left(\frac{k b^{1-\alpha}}{\gamma}\right)^{1/\beta}$ . En el caso sin expansión se obtiene

$$m \approx 2.15k^{2.44}b$$

que para  $k = 1$  indica que  $m$  debe ser siempre mayor que  $2b$ . Para el caso con una expansión parcial, obtenemos

$$m \approx 0.5k^{2.13}b^{0.34}$$

que implica que  $m$  es en general menor que  $b$  y refleja el hecho de que cuando hay expansiones queremos disminuir la probabilidad de que ellas ocurran, al igual que las divisiones.

## 5 Conclusiones

Se analizó la estructura de archivo Desorden Acotado, para determinar una relación entre el tamaño del bucket de overflow  $c$ , la cantidad de buckets primarios  $m$  y la cantidad de registros  $b$  que estos buckets pueden almacenar, minimizando el costo de inserción de los datos en la estructura. Con los resultados se graficó la relación entre el costo de inserción y los parámetros  $m$  y  $b$  mencionados,

manteniendo el otro fijo así como se graficó la relación entre  $c$  y cada uno de estos parámetros. Se estableció además un modelo experimental de los datos obtenidos, que representa una buena aproximación para la relación entre los tres parámetros considerados.

Al optimizar el tamaño del bucket de overflow se mejora la utilización de almacenamiento del archivo y el tiempo de búsqueda disminuye, lo cual permite aplicar esta mejora a las bases de datos que hagan uso de esta técnica.

Extensiones futuras a este trabajo incluyen considerar más expansiones y realizar una verificación experimental del análisis realizado.

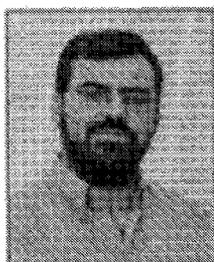
## Referencias

- R. A. Baeza-Yates, Bounded disorder: the effect of the index, *Theoretical Computer Science* 168, 1996, 21-38.
- B. Einsenbarth, N. Ziviani, G. Gonnet, K. Mehlhorn, D. Wood, The theory of fringe analysis and its application to 2-3 trees and B-trees, *Information and Control* 55, 1982, 125-174.
- W. Litwin, D. Lomet, A new method for fast data searches with keys, *IEEE Software* 4, 1987, 16-24.
- D. Lomet, A simple bounded disorder file organization with good performance, *ACMTODS* 13, 1988, 525-551.

**G. Matsliach**, Performance analysis of file organizations that use multi-bucket data leaves, *Information Processing Letters*, 36 (6), 1990, 301-310.

**A. Thrap, W. Boswell**, B+ Trees, bounded disorder and adaptive hashing, *Information Systems* 16, 1991, 56-71.

**M. Ramakrishna, P. Mukhopadhyay**, Bounded Disorder file organization, *IEEE Trans. Knowledge Eng.* 6, 1994, 79-85.



**Ricardo Baeza-Yates** es Ph.D. en Computer Science (Univ. of Waterloo, Canadá, "1989), Magister en Ing. Eléctrica (1986) y Cs. de la Computación (1985) "de la Univ. de Chile; e Ingeniero Civil Eléctrico de la misma universidad. "Actualmente es catedrático en el Depto. de Cs. de la Computación de la "Univ. de Chile y director del Centro de Investigación de la Web ([www.ciw.cl](http://www.ciw.cl)). "Sus áreas de investigación son recuperación de información, minería de la Web, "algoritmos y visualización de información. Es co-autor de un libro en "recuperación de información (Addison-Wesley, 1999), de un manual de referencia "en algoritmos y estructuras de datos (Addison-Wesley, 1991) y co-editor "de un libro en recuperación de la información (Prentice-Hall, 1992). "Ha sido dos veces presidente de la Sociedad Chilena de Ciencia de la "Computación y ha recibido premios de la Organización de Estados Americanos "y el Instituto de Ingenieros de Chile. Actualmente, entre otros cargos, "es presidente del CLEI (Centro Latinoamericano de Estudios en Informática), "miembro del directorio de IEEE-CS "y es coordinador internacional del subprograma de informática y electrónica "aplicadas de CYTED (Programa de Cooperación Iberoamericano). Durante "el año 2000 comenzó un startup de Internet para buscar en la Web Chilena" ([www.todo.cl](http://www.todo.cl)) y actualmente es director del Centro de Investigación "de la Web ([www.ciw.cl](http://www.ciw.cl)). ""



**Héctor Soza Pollman** es profesor de tiempo completo del Departamento de Ingeniería e Sisemas de la Universidad Católica del Norte, Antofagasta, Chile. Es Ingeniero Civil Matemático por la Universidad de Chile y es M. en C. en Computación por la misma Universidad. Obtuvo el primero lugar en el Concurso de Tesis del CLEI Panel 1994, otorgado por el ITESM, Cd. de México. Actualmente es Secretario Académico de la Facultad de Ingeniería y Ciencias Geológicas de la Universidad Católica del Norte. Su área de interés es el análisis de Algoritmos.

