RESTRICTED DYNAMIC PROGRAMMING FOR BROADCAST SCHEDULING

Shuoi Wang[1,2] and Hsing-Lung Chen[3]
[1]*Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan 106;* [2]*Department of Maritime Policing, Taiwan Police College, Taipei, Taiwan 116;* [3]*Department of Electronic Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan 106*

Abstract: Data broadcast has become a promising solution for information dissemination in the wireless environment due to the limited bandwidth of the channels and the power constraints of the portable devices. In this paper, a restricted dynamic programming approach which generates broadcast programs is proposed to partition data items over multiple channels near optimally. In our approach, a prediction function of the optimal average expected delay, in terms of the number of channels, the summation of the access frequencies of data items, and the ratio of the data items, is developed by employing curve fitting. Applying this function, we can find a cut point, which may be very close to the optimal cut. Thus, the search space in dynamic programming can be restricted to the interval around the found cut point. Therefore, our approach only takes $O(N\log K)$ time, where N is number of data items and K is the number of broadcast channels. Simulation results show that the solution obtained by our proposed algorithm is in fact very close to optimal one.

Key words: Data Broadcast; Data Allocation; Dynamic Programming; Multiple Channels.

1. INTRODUCTION

Recent advances in the development of portable computers and wireless communication networks make it possible for mobile clients to access data from anywhere at anytime. Broadcast-based data dissemination has become a widely accepted approach of communication in the mobile computing environment. Examples of these applications include weather forecasts, stock market quotes, and electronic newsletters. In these applications, a server periodically broadcasts a set of data items to a large community of users and clients tune in to the broadcast channel to retrieve their data of interest. Thus, the latency and the cost of data delivery are independent of the number of clients. On the contrary, an on-demand data delivery responding to a client's individual request inevitably incurs a scalability bottleneck under a heavy workload. However, in the broadcast-based system, the clients have to access data items in the broadcast channel sequentially. Some clients receive unwanted data before accessing desired data, and the corresponding response time is called the expected delay of that data item. This problem becomes worse when data access is skewed. Hence, how to allocate data items in the broadcast channel for efficient data access becomes an important issue.

Acharya[1] *et al.* propose "Broadcast Disks" architecture to minimize the average expected delay (*aed*) for the data allocation problem in a single broadcast channel. A broadcast disk involves generation of a broadcast program that schedules the data items based on their access frequencies. The broadcast is constructed by allocating data items to different "disks" of varying sizes and speeds, and then multiplexing the disks onto the same broadcast channel. This approach creates a memory hierarchy in which the fast disk contains few items and broadcasts them with high frequency while the slow disk contains more items and broadcasts them with less frequency.

In this paper, we study the data allocation problem over multiple disjoint physical channels. Such architecture has wider applicability[2,3,4]. The concept of broadcast disks can be applied to multi-channel system. That is, the disk containing data items with higher access frequencies may be distributed to a channel containing less data items such that the *aed* for those data items is reduced. The problem we study can be best understood by the illustrative example in Figure 1, where a data base contains nine items and the number of broadcast channels is three. The function of data allocation algorithm is to allocate data items into broadcast channels according to their access frequencies so as to minimize the *aed*. This is the very problem that we shall address in this paper.

Peng and Chen[5] explore the problem of generating hierarchical broadcast programs on the K broadcast channels. They develop a heuristic algorithm VF^K to minimize the *aed* of data items in the broadcast program. Although VF^K yields the *aed* close to the lower bound, it performs unstably. While the number of channels is not the power of 2, the expected delay of each channel is not balanced. Thus, the *aed* of all data items becomes worse.

For given N items with access frequencies p_r, where $1 \le r \le N$. Wong[6] shows that the lower bound of the *aed* for a periodic broadcast schedule in a single channel system is $(\sum_{r=1}^{N}\sqrt{p_r})^2/2$. Hsu[7] *et al.* extend this concept to multi-channel system, and derive that the minimal *aed* of all data items on K broadcast channels is $(\sum_{r=1}^{N}\sqrt{p_r})^2/2K$.

Yee[8] *et al.* use dynamic programming to optimally partition data items among given multiple channels. Although they determine the optimal cost, the proposed approach requires $O(KN^2)$ time and $O(KN)$ space to keep partial solutions, which makes that approach impractical in large databases.

In this paper, a prediction function which estimates the optimal *aed* of given data items on multiple channels is generated by employing curve fitting. Applying this function, a restricted dynamic programming (DP) approach is developed to allocate data items into each channel, and the resulting configuration is very close to the optimal one, but only takes $O(N\log K)$ time. The rest of paper is organized as follows. Preliminaries are given in Section 2. In Section 3, we develop a prediction function to estimate optimal *aed* by employing curve fitting. In Section 4, based on the predication function, a restricted DP approach which generates broadcast programs on multiple channels is proposed. Performance studies are presented in Section 5. This paper concludes with Section 6.

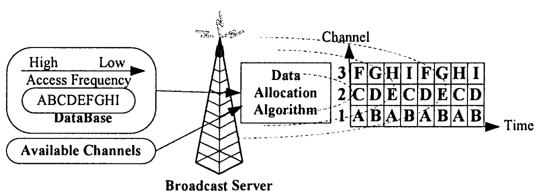

Figure 1. Data allocation problem on multiple channels.

2. PRELIMINARIES

2.1 Architectural Assumptions

This paper focuses on the wireless broadcast environment. Some assumptions should be restricted in order to make our work feasible. These assumptions include: a) A data base contains N equal-sized items, denoted as d_j, where $1 \leq j \leq N$; b) There are K equal-bandwidth physical channels for data broadcast, which can not be combined to form a single high-bandwidth one; c) Let G_i be the set of data items to be broadcast on channel i, where $1 \leq i \leq K$ and $\sum_{i=1}^{K} |G_i| = N$. The data items in each channel are sent out in a round robin manner. Each data item is broadcast only on one of these channels and a time slot is defined as the amount of time necessary to transmit an item; d) Each data item d_j has a corresponding access frequency p_j, which denotes the probability that data item d_j is requested by the clients. We assume requests are exponentially distributed, so that at each time slot the probability of a client requesting d_j is determined by p_j, where $\sum_{j=1}^{N} p_j = 1$, and; e) The mobile client can listen to multiple channels simultaneously.

2.2 Problem Statement

Our problem is to partition the data items into K groups and to allocate items in each group into an individual channel, such that the *aed* of all data is minimized. When items in set G_i are cyclically broadcast on channel i, the *expected delay* in receiving any particular data on channel i is $|G_i|$ / 2. Thus, given K channels, the *aed* of all data items can be expressed as

$$\sum_{i=1}^{K}\left(\frac{|G_i|}{2}\sum_{d_j \in G_i} p_{d_j}\right).$$

Theoretically, data allocation over multiple channels can be viewed as a *partition* problem for data, as shown in Figure 2. First, all items are sorted in descending order according to their access frequencies. Then, partition all data into G_1, G_2, …, G_K sets. Define cut_i as the cut point between G_i and G_{i+1}. For convenience, let cut_i be the index of last data item of G_i. For example, cut_1 = 2 and cut_i = 20 in Figure 2. Our goal is to find the optimal configuration set of cut points, $\{cut_i \mid 1 \le i \le K\text{-}1\}$, in a way that the *aed* is minimized, but only takes $O(N\log K)$ time.

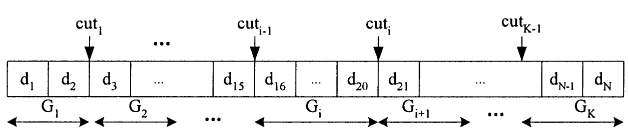

Figure 2. Partition problem for allocating N data items on K channels.

Dynamic programming[8] (DP) approach provides an optimal solution for the data allocation problem, but its time and space complexities may preclude it from practical use. Let us state the basic DP approach for the partition problem first. Data are sorted in descending order according to their access frequencies. Define $C_{i,j}$ as the *aed* of a channel containing d_i through d_j. $C_{i,j} = \frac{j-i+1}{2}\sum_{r=i}^{j} p_r$, where $1 \le i \le j \le N$. Let $opt_{k,j}$ be the optimal *aed* for allocating d_1 through d_j on k channels. Given one channel, $opt_{1,j} = \frac{j}{2}\sum_{r=1}^{j} p_r$, where $1 \le j \le N$. The optimal *aed* for allocating N items on K channels is $opt_{K,N}$. Now, we present the basic DP algorithm for solving the partition problem in Figure 3.

In (1) as shown in Figure 3, the search space of determining the $opt_{k,j}$ is linear. Thus, by inspecting the nested loop structure of the basic DP algorithm, its running time is $O(KN^2)$. Suppose we can approximately estimate the *aed* of given data items on multiple channels, then we can find a cut point, s, which may be very close to the optimal cut. Thus the search space of the possible cut point, $\boldsymbol{r}$, can be restricted to some promising range (say seven cut points), that is, $s\text{-}3 \leq \boldsymbol{r} \leq s\text{+}3$, then the time to determine $opt_{k,j}$ becomes constant. Thus, the search space in DP approach can be reduced.

```
For all cut points j from 1 to N
    opt_{1,j} = (j/2) Σ_{r=1}^{j} p_r
End
For all stages k from 2 to K
    For all cut points j from k to N
        opt_{k,j} = min{opt_{k-1,r} + C_{r+1,j}}, where k-1 ≤ r ≤ j-1        (1)
    End
End
```

Figure 3. The basic dynamic programming algorithm.

2.3 Formulation of Estimated *aed* on Multiple Channels

Assume $p_1 \geq p_2 \geq \ldots \geq p_N$, and $\sum_{r=1}^{N} p_r = 1$. Denote $optimal_k^{\alpha}$ as the optimal *aed* of the last $n \leq N$ data items allocated to $k \leq K$ channels, where α be the summation of access frequencies of the last n data items, $\alpha = \sum_{r=N-n+1}^{N} p_r$. Thus, $optimal_K^1$ is the *aed* of all N items allocated to K channels.

Denote LB_k^{α} as the lower bound of the *aed* when the last n items are allocated to k channels. Thus, LB_1^1 is the lower bound[6] of the *aed* when all N items are allocated to a single channel, that is, $LB_1^1 = (\sum_{r=1}^{N}\sqrt{p_r})^2/2$.

Assume a high-bandwidth channel has bandwidth B bytes/sec, and each data item is of size D bytes. Therefore, broadcast one item on the channel will take D/B seconds. If this fast channel divides into k sub-channels, and each sub-channel has bandwidth B/k bytes/sec. Then, broadcasting each item in one of these sub-channels will take kD/B seconds. Thus, $LB_k^1 = LB_1^1/k$. Let

$$factor_k^1 = optimal_k^1 / LB_k^1 = optimal_k^1 \Big/ \frac{LB_1^1}{k} = k \times optimal_k^1 / LB_1^1 .$$

When the summation of access frequencies of items is not equal to 1, factor can be normalized as follows.

$$factor_k^{\alpha} = k \times \frac{\sum_{j=K-k+1}^{K} \frac{|G_j|}{2}\left(\sum_{r\in G_j} \frac{p_r}{\alpha}\right)}{\frac{1}{2}\left(\sum_{r=N-n+1}^{N}\sqrt{\frac{p_r}{\alpha}}\right)^2} = k \times \frac{\frac{1}{\alpha}\sum_{j=K-k+1}^{K} \frac{|G_j|}{2}\left(\sum_{r\in G_j} p_r\right)}{\frac{1}{2\alpha}\left(\sum_{r=N-n+1}^{N}\sqrt{p_r}\right)^2}$$

Let $LB_1^{\alpha} = (\sum_{r=N-n+1}^{N}\sqrt{p_r})^2/2$. Thus,

$$factor_k^{\alpha} = k \times optimal_k^{\alpha} / LB_1^{\alpha} \qquad (2)$$

$$=> optimal_k^\alpha = factor_k^\alpha \times LB_1^\alpha / k \quad (3)$$

Given different data items with access frequencies and number of channels in (2), we can get different values of factor. With these values, if we can find a function to predict $factor_k^\alpha$ precisely, then we can get an estimated value very close to $optimal_k^\alpha$. Our idea is inspired by Hsu[7] *et al.* Assume data items in G_1, G_2, ..., and G_i have been allocated to the first i channels. Thus, the optimal *aed* of the unallocated data items on the (K-i) channels is $optimal_{K-i}^\alpha = factor_{K-i}^\alpha \times LB_1^\alpha / (K-i)$, where α is the summation of access frequencies of the data items on the (K-i) channels. But Hsu *et al.* use $1 \times LB_1^\alpha / (K-i)$ to estimate it.

3. PREDICTION FUNCTION OF FACTOR

To predict the value of $factor_k^\alpha$, we employ curve fitting to find a prediction function f in terms of α, k, and *ratio*, where α denotes the summation of the access frequencies of unallocated data items ($\alpha \le 1$), k denotes the number of unallocated channels ($k \le$ K), and *ratio* denotes the distribution of the data items. Intuitively, f is related to these three parameters. Simulation results confirm this intuition. Ratio[9] means (100×*ratio*)% users focus on 100×(1-*ratio*)% data items, where $0 \le ratio \le 1$.

DEFINITION: *ratio* is the summation of the access frequencies of the first 100×(1-*ratio*)% data items. In other words, $ratio = \frac{p_1 + p_2 + \ldots + p_{(1-ratio)N}}{p_1 + p_2 + \ldots + p_N}$

We can determine the ratio of the data items by examining whether the first N_1 data items satisfy $\sum_{j=1}^{N_1} p_j + (N_1/N) = 1$, where N_1 varies from 1 to N. Thus, $ratio = 1 - (N_1/N)$ if satisfied.

Let $est_{k,r}$ denote as the estimated *aed* of d_r through d_N allocated on k channels. Once, $f(\alpha, k, ratio)$ is found, from (3),

$$est_{k,r} = f(\alpha, k, ratio) \times LB_1^\alpha / k, \text{ where } \alpha = \sum_{q=r}^{N} p_q .$$

We call this approach as PKR estimation. It is better than Hsu *et al.*, because their $factor_k^\alpha$ are always 1.

LEMMA 1. $est_{k,r}$ can be computed in constant time.

PROOF. In initialization stage, for each data item d_j, $1 \le j \le N$, we associate a $accp_j$ with d_j, which is defined as the sum of access frequencies of data items d_1 through d_j. Initially, $accp_1 = p_1$. Other $accp_j$ can be computed by the recurrence equation $accp_j = p_j + accp_{j-1}$. Thus, $\alpha = accp_N - accp_{r-1}$. Given α, k, and *ratio*, we can compute $f(\alpha, k, ratio)$ in constant time. We also associate a acc_sqrtp_j with d_j, which is defined as $\sum_{r=1}^{j} \sqrt{p_r}$. Initially,

acc_sqrtp_1 = $\sqrt{p_1}$. Other acc_sqrtp_i can be computed recursively by $acc\_sqrtp_j = \sqrt{p_j}$ + $acc\_sqrtp_{j\text{-}1}$ Thus, $LB_1^\alpha = (\sum_{q=r}^{N}\sqrt{p_q})^2/2$ = (acc_sqrtp_N – $acc\_sqrtp_{r\text{-}1})^2/2$, which also can be computed in constant time. Since $est_{k,r}$ = $f(\alpha,k,ratio)\times LB_1^\alpha/k$, this lemma follows. ■

The derivation of $f(\alpha, k, ratio)$ contains the following steps.

Step A. Prediction function in terms of α.

Given data items with specific *ratio* and *k*, we can compute $optimal_k^\alpha$ and LB_1^α during the dynamic programming process. Thus, we can obtain a value of factor by (2). Let α_n be the summation of access frequencies of the last *n* data items, where $k \le n \le N$. For each α_n, we have a computed $factor_n$. Thus, we can gather many data points (α_n, $factor_n$). Plotting these data points and using the least-squares method for fitting data, we can get a result similar to Figure 4. The dotted line shown in Figure 4 denotes the optimal data points and the dashed line denotes the function of curve fitting we adopt, which is very close to the optimal one. The function in Figure 4 is described by

$$factor = 1 + a1\ ((m_9 - ratio) \times m_{10} \times \alpha)^{a2}, \quad (4)$$

where $a1$ and $a2$ are coefficients with specific *k* and *ratio*, m_9 = 0.78, and m_{10} = 1. Figure 4 is for the case when *k* is 3 and *ratio* is 0.80.

Step B. $a2$ in terms of $a1$.

For each input combination of *k* (e.g., 3, 4, …, 100) and *ratio* (e.g., 0.55, 0.60, …, 0.85), we obtain a figure similar to Figure 5 and a function has the form like (4), with different coefficients of ($a1$, $a2$). Thus, we can gather many data points ($a1_{k,ratio}$, $a2_{k,ratio}$). Plotting these data points and then employing curve fitting, we can derive a function in terms of $a1$ to denote $a2$, as shown in Figure 5. That function is

$$a2 = m_1\ (a1 - m_2) + m_3, \quad (5)$$

where m_1, m_2, and m_3 are coefficients.

Step C. Prediction function in terms of *ratio*.

For a given *k*, we have a respective value of $a1_{k,ratio}$ for each *ratio*. Thus, we can gather many data points (*ratio*, $a1_{k,ratio}$). Plotting these data points, as the dotted line shows in Figure 6. Using curve fitting, we can get a function in terms of *ratio* to denote $a1$, as the dashed line in Figure 6. That function is

$$a1 = b1\ (ratio)^{m_8}, \quad (6)$$

where m_8 = 8, $b1$ is a coefficient of a specific *k*.

Step D. Prediction function in terms of *k*.

Figure 6 is for the case when *k* is 3. For each value of *k*, we can obtain a respective coefficient of $b1_k$. Plotting these data points ($b1_k$, *k*), as the dotted line shows in Figure 7. Using curve fitting, we can obtain a function in terms of k to denote $b1$, as the dashed line in Figure 7. That function is

$$b1 = m_4 (k + m_7)^{-m_5} + m_6. \qquad (7)$$

where m_4 to m_7 are coefficients.

Step E. *factor* = $f(\alpha, k, ratio)$.

Thus, combining (4) to (7), we obtain a prediction function of factor f in terms of α, k and *ratio* with initial values of m_1 to m_{10}. For all data points $\langle factor_n, \alpha_n, k_n, ratio_n \rangle$, we proceed the curve fitting again to obtain a better fit. Thus, we have the final values of m_1 to m_{10} in $f(\alpha, k, ratio)$. Table 1 lists the coefficients derived in f.

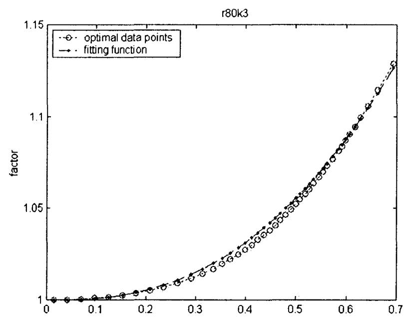

Figure 4. factor in terms of α.

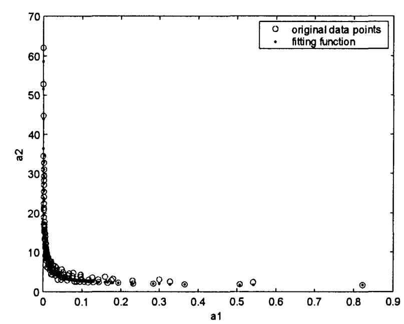

Figure 5. a2 in terms of a1.

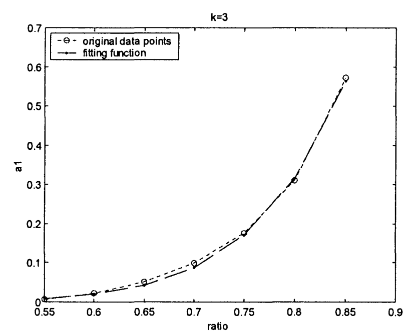

Figure 6. a1 in terms of ratio.

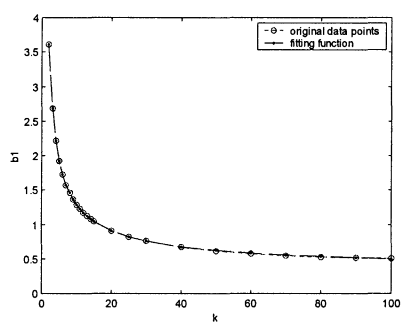

Figure 7. b1 in terms of k.

Table 1. Coefficients used in function f.

m_1	2.101260517	m_6	-0.12243
m_2	0.304910179	m_7	-0.19996
m_3	0.353111223	m_8	8.878037
m_4	11.72018907	m_9	28.20646
m_5	0.87676042	m_{10}	0.030049

4. A RESTRICTED DP ALGORITHM

The main idea of our proposed approach uses PKR estimation to find a cut point, which may be very close to the optimal cut. Therefore, the search space in DP can be reduced, thus, saves computations.

At stage k in the basic DP algorithm, if the cut point of $(k\text{-}1)_{th}$ partition is r, then the estimated optimal *aed* of all data items on K channels is the sum of $opt_{k-1,r}$ and $est_{K-k+1,r+1}$. Let s be a cut point such that $opt_{k-1,s} + est_{K-k+1,s+1} = \min\{opt_{k-1,r} + est_{K-k+1,r+1}\}$, where $k \le r \le N\text{-}1$. If we restrict the search space of determining $opt_{k,j}$ to the interval [s-3, s+3] in (1), then the time to determine $opt_{k,j}$ is constant. Note that, if the optimal cut point of $(k\text{-}1)_{th}$ partition is outside the interval [s-3, s+3] we predict, computed $opt_{k,j}$ is not optimal, resulting in a non-optimal solution. Now, we present our proposed Restricted DP algorithm (RDP) in Figure 8.

In Figure 8, when the minimum value of $opt_{k,j}$ is determined, the corresponding r will be stored in $lastp_{k,j}$, which can help us construct an optimal configuration. Define $selcut_i$ as the cut point of selected solution, where $1 \le i \le K\text{-}1$. Thus, $selcut_{K-1} = estcut_{K-1}$. The earlier selected cut points can be retrieved recursively by $selcut_i = lastp_{i+1,selcut_{i+1}}$, where $1 \le i \le K\text{-}2$. Therefore, the set of selected cut points can be retrieved by the backward process.

With the help of PKR estimation, the RDP algorithm can restrict the search space to some promising range. It is unlike the basic DP have to examine all possible cut points, therefore, our approach only takes $O(KN)$ time.

Some modifications in the RDP algorithm can improve the performance if some subproblems in the subproblem space need not be solved at all. As stated in Lemmas[7] 3 and 4, the number of the data items allocated to each channel possesses the hierarchical property. That is,

$$|G_{i-1}| \le |G_i| \le \frac{N - \sum_{j=1}^{i-1}|G_j|}{K-(i-1)} \le \frac{N-(i-1)}{K-(i-1)}.$$

This inequality can be used to reduce the time and space requirements in the RDP algorithm. We implement a slightly modified version of the RDP in Figure 9, called Bounded-Restricted Dynamic Programming (BRDP).

Theorem 1. The time and space complexities of the BRDP algorithm are both $O(N\log K)$

Proof. In the worst case, creating entries of the first row of the table *opt* requires N/K computations. Thus, determining $estcut_1$ takes N/K computations. Similarly, creating entries of the second row of the table *opt* requires $(N\text{-}1)/(K\text{-}1)$ computations. Then, determining $estcut_2$ spends $(N\text{-}1)/(K\text{-}1)$ computations. This process is repeatedly done until cut point $estcut_{K-1}$ is found. Thus,

$$\frac{N}{K}+\frac{N-1}{K-1}+\ldots+\frac{N-K+2}{2} \leq \frac{N}{K}+\frac{N}{K-1}+\ldots+\frac{N}{1}=N\sum_{i=1}^{K}\frac{1}{i}=NH_K,$$

where $H_K=lnK+O(1)$. Therefore, The time and space complexities of the BRDP algorithm are both $O(N\log K)$. ■

For all cut points j from 1 to N

$opt_{1,j}=\frac{j}{2}\sum_{r=1}^{j}p_r$

End

Let s be a cut point $estcut_1$ s.t. $opt_{1,s}+est_{K-1,s+1}=\min\{opt_{1,r}+est_{K-1,r+1}\}$, where $1 \leq r \leq N-1$.

For all stages k from 2 to K-1

For all cut points j from $\max\{|s-2|, k\}$ to N

$opt_{k,j}=\min\{opt_{k-1,r}+C_{r+1,j}\}$, where $\max\{|s-3|, k\} \leq r \leq \min\{s+3, j-1\}$.

End

Let s be a cut point $estcut_k$ such that

$opt_{k,s}+est_{K-k,s+1}=\min\{opt_{k,r}+est_{K-k,r+1}\}$, where $\max\{|estcut_{k-1}-2|, k\} \leq r \leq N-1$.

Figure 8. The Restricted DP Algorithm.

For all cut points j from 1 to **N/K**

$opt_{1,j}=\frac{j}{2}\sum_{r=1}^{j}p_r$

End

Let s be a cut point $estcut_1$ such that

$opt_{1,s}+est_{K-1,s+1}=\min\{opt_{1,r}+est_{K-1,r+1}\}$, where $1 \leq r \leq$ **N/K**.

For all stages k from 2 to K-1

For all cut points j from $\max\{|s-2|, k\}$ to **$(N-(k-1))/(K-(k-1))$**

$opt_{k,j}=\min\{opt_{k-1,r}+C_{r+1,j}\}$, where $\max\{|s-3|, k\} \leq r \leq \min\{s+3, j-1\}$.

End

Let s be a cut point $estcut_k$ such that

$opt_{k,s}+est_{K-k,s+1}=\min\{opt_{k,r}+est_{K-k,r+1}\}$,

where $\max\{|estcut_{k-1}-2|, k\} \leq r \leq$ **$(N-(k-1))/(K-(k-1))$**.

End

Figure 9. The Bounded-Restricted DP Algorithm.

Table 2. Parameters used in performance evaluation.

Definition	**Notation**	**Range**
Number of data items to be broadcast	N	1000 - 5000
Number of broadcast channels	K	2 - 100
Zipf distribution parameter	*ratio*	0.55 - 0.85

Table 3. Algorithms compared in performance evaluation.

Algorithm	**Notation**
Optimal Solution[9]	OPT
Data-Based Algorithm[2]	HSU
PKR Approach	PKR
Bounded-Restricted Dynamic Programming Algorithm	BRDP

5. PERFORMANCE EVALUATION

Our experiments are developed by C on the computer with Intel Pentium III 1 GHz and 256MB RAM, running Windows XP. The access frequencies of broadcast items are modeled by the Zipf distribution. Let

$$p_j = \frac{j^{\theta} - (j-1)^{\theta}}{N^{\theta}}, \; 1 \le j \le N,$$

where θ is the parameter of Zipf distribution. θ is computed[9] by

$$\theta = \frac{\log(ratio)}{\log(1-ratio)}.$$

In the Zipf distribution, the access frequencies of the data follows the *ratio*/(1-*ratio*) rule. For example, 80/20 rule means that 80 percent users are usually interested in 20 percent data items.

The simulation parameter settings for our experiments are listed in Table 2. Table 3 lists the algorithms we compared. All algorithms are implemented as described by their respective authors. Instead of using lower bound[7], PKR algorithm uses PKR estimation to estimate the *aed* of the unallocated data items on the unallocated channels.

The simulation experiments aim at studying the performance of our proposed PKR estimation approach and the BRDP algorithm, compared with another two algorithms[7,8]. Given data items with specific ratio and the number of broadcast channels *K*, if the set of the cut points we found is same as the set of optimal cut points, we call it *hit*. The performance metric of the algorithms is the *hit rate* (the distance to the best solution), which is the percentage of number of hits to number of experiments. For each given number of items with specific ratio, we performed 99 experiments, that is, the range of the number of channels varying from 2 to 100.

Figure 10 shows the effect of the number of the data items on the hit rate for three different approaches under different ratios. Algorithm PKR is better than Hsu's algorithm. The reason is that PKR estimation can generally predict the *aed* of all data items more precisely than the lower bound in the partition operation. Due to the fitting deviation, when the distribution is near uniform (e.g., ratio = 0.55), PKR estimation is not good enough. Perhaps there are different choices of prediction function *f* which would lead to a better fit. Obviously, BRDP can tolerate the estimation error. In all simulation runs under the change of skew factor ratio, number of items, and number of channels, we observe in Figure 10 the results obtained by BRDP is of high quality and is in fact very close to the optimal one. In most cases, the hit rate of BRDP is higher than 95%, and it outperforms Hsu's algorithm about 2 times higher.

6. CONCLUSIONS

We propose a restricted dynamic programming algorithm to solve the problem of allocating data items into multiple broadcast channels, and the proposed approach only takes $O(N\log K)$ time and space. In our approach, we adopt a prediction model which estimates the optimal *aed* of data items on multiple channels more precisely than lower bound. Simulation results show that the solution obtained by our algorithm is of very high quality and is very close to the optimal one.

Error! Not a valid link.Error! Not a valid link.Error! Not a valid link.Error! Not a valid link.

Figure 10. The effect of the number of data items under different ratios.

REFERENCES

1. S. Acharya, M. Franklin, S. Zdonik, and R. Alonso, Broadcast disks: data management for asymmetric communication environments, *ACM SIGMOD*, pp. 199-210, 1995.
2. C. Hsu, G. Lee, and A.L.P. Chen, Index and data allocation on multiple broadcast channels considering data access frequencies, *IEEE MDM*, pp. 87-93, 2002.
3. C. Hu and M.S. Chen, Adaptive balanced hybrid data delivery for multi-channel data broadcast, *IEEE ICC*, pp. 960-964, 2002.
4. D.A. Tran, K.A. Hua and N. Jiang, A generalized air-cache design for efficiently broadcasting on multiple channels, *ACM SAC*, 2001.
5. W.C. Peng and M.S. Chen, Dynamic generation of data broadcasting programs for a broadcast array in a mobile computing environment, *ACM CIKM*, pp. 35-45, 2000.
6. J.W. Wong, Broadcast delivery, *Proc. IEEE*, **76**(12), 1566-1577, 1988.
7. C. Hsu, G. Lee, and A.L.P. Chen, A near optimal algorithm for generating broadcast programs on multiple channels, *ACM CIKM*, pp. 303-309, 2001.
8. W.G. Yee, S.B. Navathe, E. Omiecinski and C. Jermaine, Efficient data allocation over multiple channels at broadcast servers, *IEEE Trans. Comp*, **51**(10),1231-1236, 2002.
9. D.E. Knuth, *The Art of Computer Programming*, vol. 3, Addison-Wesley, 2nd, 1981.