On Exploring Channel Allocation in the Diverse Data Broadcasting Environment

Hao-Ping Hung and Ming-Syan Chen
Graduate Institute of Communication Engineering
National Taiwan University
Taipei, Taiwan, ROC
E-mail: mschen@cc.ee.ntu.edu.tw, hphung@arbor.ee.ntu.edu.tw

Abstract

In recent years, data broadcasting becomes a promising technique to design a mobile information system with power conservation, high scalability and high bandwidth utilization. However, prior reseach topics in data broadcasting are mainly based on the assumption that the disseminated data items are of the same size. We explore in this paper the problem of generating broadcast programs in a $diverse$ $data$ $broadcasting$ *environment, in which disseminated data items can be of different sizes. Given the broadcast database and the channel number, we propose algorithms DRP (Dimemsion Reduction Partitioning) to perform the channel allocation for each data item. Moreover, a Cost-Diminishing Selection mechanism is also used to help DRP achieve the local optimum with low complexity. With the capability of generating effective broadcast programs efficiently, the proposed mechanism can be practically used in a* $diverse$ $data$ $broadcasting$ *environment.*

1 Introduction

The advance in wireless communication enables users to access information anytime, anywhere, via laptops, PDAs and smart phones. In addition to conventional text-based information like weather forecast and stock information, an $information\ system$ provides modern information services including Web browsing and multimedia access. In order to provide better services for mobile users, researchers have encountered, and are endeavoring to overcome challenges in various research areas such as mobile data dissemination [4], location-dependent data management [16], pervasive computing [15], and so on.

$Data\ Broadcasting$ is a well-known mechanism to disseminate data items from an information system to mobile users. Such a mechanism is also known as a $push$-$based$ dissemination. In a $push$-$based$ information system, as shown in Figure 1, the server generates a broadcast program by collecting the access patterns of mobile users, and broadcasts data items periodically to mobile users via multiple channels. The period of each broadcast channel is viewed as a $broadcast\ cycle$. To retrieve a data item, an user with a mobile device should listen to the broadcast channel and wait for the data of interest to appear on the broadcast channel. The average waiting time of a broadcast channel is composed of two components: the *probe time* and the *download time*. The analytical model is described as follows. Consider that N data items with size z are broadcast periodically via a broadcast channel with bandwidth b. The *probe time*, W_{probe}, is the time that an user should wait until the data of interest appears in the channel and thus $W_{probe} = \frac{1}{2}(broadcast\ cycle\ time) = \frac{Nz}{2b}$. The *download time*, $W_{download}$, is the time that a user should spend for downloading the data item via the broadcast channel, i.e., $W_{download} = z/b$. Therefore the waiting time, $W_b = W_{probe} + W_{download} = \frac{Nz}{2b} + \frac{z}{b}$.

There are many research topics in generating the broadcast programs to broadcast data items via multiple broadcast channels [1][7][8][9][10][11][14]. A $flat$ broadcast program, which allocates data items within broadcast channels with equal appearance frequencies, is a straight forward way to generate a broadcast program. However, it is ineffective since the waiting time of data items with different access frequencies is the same. In order to overcome the effectiveness problem, approaches are proposed in [7][14] to generate broadcast programs in which the waiting time of popular data items (i.e., with higher access frequencies) is shorter than that of unpopular data items (i.e., with lower access frequencies). The works in [9][10] focus on broadcasting dependent data for ordered and unordered queries. Moreover, the broadcast program allows a data item to appear in different broadcast channels simul-

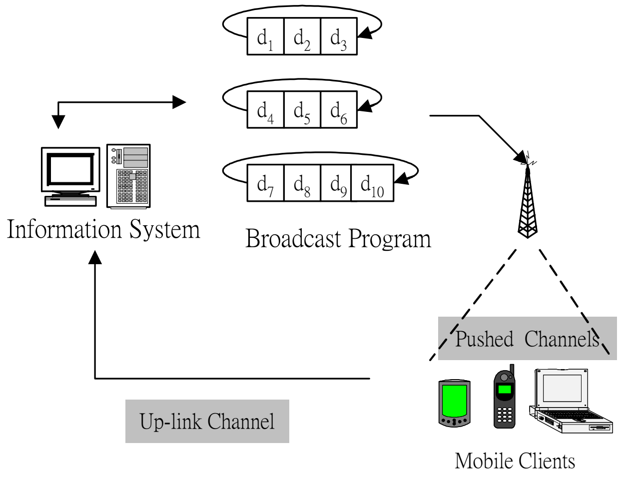

Figure 1. The architecture of broadcast-based data dissemination

taneously. Such a replication issue is addressed in [8].

Since the earlier wireless communication was bounded by the channel bandwidth and the capability of mobile devices, the disseminated data items were usually of text-based information. As a result, the prior research related to mobile data dissemination was mostly based on the assumption that each item is of the same size. However, in the advanced communication environment with lager bandwidth, the mobile users can use their devices with higher capability to access various information such as still image, video and audio. The data items with different sizes are disseminated in a modern information system. The dissemination policies for a modern information system based on conventional models hence suffer from effectiveness issues. To differentiate our work from those in the conventional broadcasting environment, we use a term $diverse\ data\ broadcasting$[1] to describe the broadcast environment in which the data items with different sizes are disseminated. In this paper, we focus on generating broadcast programs in a $diverse\ data\ broadcasting\ environment$. The broadcast program is generated according to the access frequency and the size of each data item. Note that although the broadcasting technique employed in video-on-demand systems in recent years [3][12][13] allows each video object to be of different sizes, the access frequency of each video object is not considered in those areas. To the best of our knowledge, there is no prior research allowing of different data item sizes and access frequencies simultaneously. This feature not only strengthens the practicality of our work but also distinguishes this paper from others.

In this paper, we first derive the analytical model of $diverse\ data\ broadcasting$. According to the analytical model, the channel allocating problem is transformed to a $grouping\ probmem$ with a specific $cost\ function$. We next propose a heuristic approach to perform channel allocation (i.e., grouping) for each data item: DRP, which stands for Dimension Reduction Partitioning. Algorithm DRP is a top-down group-splitting approach, in which the two-dimensional grouping problem is simplified as an one-dimensional partitioning problem. In addition, a Cost-Diminishing Selection mechanism, abbreviated as CDS, is also proposed to improve the effectiveness to the local optimum. In essence, CDS checks the possible moving operations for a certain data item from one group to another, and determines the best choice in each iteration. In general, what we propose in this paper can be viewed as a two-step allocation scheme. Algorithm DRP provides $rough\ allocation$, which achieves satisfactory quality, whereas CDS provides $fine\ allocation$, which achieves suboptimum by refining the results of the $rough\ allocation$. In order to verify the effectiveness of DRP and CDS, several experiments are conducted. First, the proposed algorithm is compared to the approach adopted in the conventional broadcasting environment. Moreover, we also compare our channel allocation results to the global optimal ones, which shows that the local optimal results of our approach are in fact very close to the global optimal results. Finally, by comparing the executing complexity with algorithm GOPT, which achieves the global optimal solutions by using the Genetic Algorithm, it is shown that our approach has a much lower complexity than that of GOPT and emerges as a powerful and practical solution to generating broadcast programs in a $diverse\ data\ broadcasting$ environment.

The rest of the paper is organized as follows. Preliminaries are given in Section 2. In Section 3, the broadcast program generating approach DRP is proposed with mechanism CDS. The experimental results are shown in Section 4. Finally, this paper concludes with Section 5.

2 Preliminaries

2.1 Model of Diverse Data Broadcasting

Given a database D with its size $|D| = N$ and the channel number K, the goal of the broadcast program is to allocate each data item to a specific channel c_i, where $1 \leq i \leq K$. Each channel contains an item set D_i with its size $|D_i| = N_i$, where $\sum_{i=1}^{K} N_i = N$, and $D_i \cap D_j = \{\varnothing\}$ for $i \neq j$. In a $diverse\ broadcast$ environment, each data item has different item size. Therefore, a data item $d_j^{(i)}$,

[1]Note that the term *diverse data broadcasting* is also referred to as *heterogeneous data broadcasting [2]*, in which on-demand broadcast mechanism is discussed.

Table 1. Description of the symbols

Description	Symbol
Number of the broadcast channels	K
The i-th broadcast channel	c_i
The database of the broadcast data items	D
Number of the broadcast data items	N
The item set of the data items allocated to c_i	D_i
Number of the data items allocated to c_i	N_i
The j-th data item in c_i	$d_j^{(i)}$
The size of the j-th data item in c_i	$z_j^{(i)}$
The access frequency of the j-th data item in c_i	$f_j^{(i)}$
The bandwidth of each broadcast channel	b

which represents the j-th data item in c_i, contains two features: the access frequency $f_j^{(i)}$, and the item size $z_j^{(i)}$, where $\sum_{i=1}^{K}\sum_{j=1}^{N_i} f_j^{(i)} = 1$. Table 1 shows the description of symbols used in modeling the broadcast program.

Next, we consider the waiting time of each channel. Let $W^{(i)}$ represent the average waiting time of c_i. We derive the analytical model of the diverse data broadcasting. For the channel c_i, the data items in D_i are broadcast periodically. The aggregate size of D_i is $\sum_{j=1}^{N_i} z_j^{(i)}$. Let b represent the bandwidth of the channel. The broadcast cycle of c_i can be derived by $(\sum_{j=1}^{N_i} z_j^{(i)})/b$. The average *probe time* of c_i is $(\sum_{j=1}^{N_i} z_j^{(i)})/(2b)$. In addition to the *probe time*, it also takes $z_j^{(i)}/b$ to download the data item $d_j^{(i)}$. Therefore, $W_j^{(i)}$, the waiting time of the data item $d_j^{(i)}$ in the channel c_i can be derived as follows:

$$W_j^{(i)} = \frac{\sum_{j=1}^{N_i} z_j^{(i)}}{2b} + \frac{z_j^{(i)}}{b}. \quad (1)$$

Also, we can obtain the average waiting time of c_i, denoted as $W^{(i)}$, according to Eq. (1).

$$\begin{aligned} W^{(i)} &= \frac{\sum_{j=1}^{N_i} f_j^{(i)} W_j^{(i)}}{\sum_{j=1}^{N_i} f_j^{(i)}} \\ &= \frac{(\sum_{j=1}^{N_i} f_j^{(i)})(\sum_{j=1}^{N_i} z_j^{(i)})}{2b\sum_{j=1}^{N_i} f_j^{(i)}} + \frac{\sum_{j=1}^{N_i} f_j^{(i)} z_j^{(i)}}{b\sum_{j=1}^{N_i} f_j^{(i)}}. \end{aligned}$$

Therefore, the waiting time of the broadcast program, denoted as W_b, can be viewed as the average value of the waiting time of each channel c_i. Thus,

$$\begin{aligned} W_b &= E[W^{(i)}] = \sum_{i=1}^{K}(\sum_{j=1}^{N_i} f_j^{(i)})W^{(i)} \\ &= \sum_{i=1}^{K}[\frac{(\sum_{j=1}^{N_i} f_j^{(i)})(\sum_{j=1}^{N_i} z_j^{(i)})}{2b} + \frac{\sum_{j=1}^{N_i} f_j^{(i)} z_j^{(i)}}{b}] \\ &= \frac{1}{2b}\sum_{i=1}^{K}[(\sum_{j=1}^{N_i} f_j^{(i)})(\sum_{j=1}^{N_i} z_j^{(i)})] \\ &\quad + \frac{1}{b}\sum_{i=1}^{K}\sum_{j=1}^{N_i} f_j^{(i)} z_j^{(i)} \end{aligned} \quad (2)$$

2.2 Problem Formulation

The goal of this paper is to generate a broadcast program, which allocates each data item to a specific channel, in such a way that W_b can be minimized. Following Eq. (2), W_b is composed of two terms. The term $\sum_{i=1}^{K}[(\sum_{j=1}^{N_i} f_j^{(i)})(\sum_{j=1}^{N_i} z_j^{(i)})]$ results from the effect of the *probe time*, while the term $\sum_{i=1}^{K}\sum_{j=1}^{N_i} f_j^{(i)} z_j^{(i)}$ represents the effect of downloading the data item. The second term can be viewed as the summation of the product value of the access frequency and the size of *all* data item in the database D. That is, given the database D and the number of channels K, the second term is thus determined regardless of the scheduling schemes employed. Moreover, the channel bandwidth b is also a constant value. Therefore, how the broadcast program is generated only affects the term $\sum_{i=1}^{K}[(\sum_{j=1}^{N_i} f_j^{(i)})(\sum_{j=1}^{N_i} z_j^{(i)})]$.

In order to simplify the problem, we define a cost function to model the first term in Eq. (2) as follows:

$$cost = \sum_{i=1}^{K} cost(i) = \sum_{i=1}^{K}[(\sum_{j=1}^{N_i} f_j^{(i)})(\sum_{j=1}^{N_i} z_j^{(i)})], \quad (3)$$

where $cost(i) = (\sum_{j=1}^{N_i} f_j^{(i)})(\sum_{j=1}^{N_i} z_j^{(i)})$. We can reformulate the broadcast program generating problem to a *grouping problem*. Given the database D, we want to group the data items in D into K different clusters so that the value *cost* can be minimized. Since solving the grouping problem optimally is computationally prohibitive and suffers from scalability issues, a heuristic algorithm DRP is proposed to achieve the satisfactory results with much lower complexity. Moreover, mechanism CDS is proposed to refine the grouping results to the local optimum. As shown in the experimental results in the later sections, the local optimum will be very close to the global optimum and can be achieved with much lower complexity. It will be more suitable and practical to generate the diverse broadcast program by using the proposed approach.

3 Generating Broadcast Programs

3.1 Dimension Reduction Partitioning

Algorithm DRP can be viewed as a top-down group-splitting approach. Initially, there is only one group D. In each iteration of DRP, a group is selected and split into two disjoint subgroups, and the group number is increasing by one. DRP continues until the group number reaches K. Since each item contains two features, item size and access frequency , splitting a group D_i into two subgroups, D_j and D_k, optimally requires huge complexity because $2^{|D_i|}$ possibilities have to be considered. Therefore, in order to reduce complexity, we use the *benefit ratio*, denoted as *br*, to model the features of the data items. The benefit ratio br_i of the data item d_i is defined as access frequency divided by item size. i.e., $br_i = \frac{f_i}{z_i}$. The reason of using the benefit ratio to describe the feature of a data item is that in the *diverse data broadcasting* environment, the access probability corresponds to the *profit*, whereas the item size corresponds to the *cost*. The data item with a higher access probability and smaller item size will tend to be put in the broadcast channel with a shorter broadcast cycle. The intuition of DRP is to consider the ratio *br* instead of the item size and the access probability. Therefore, the two dimensional group-splitting problem can be reduced to a one-dimensional partitioning problem. Before describing algorithm DRP, several definitions are given to facilitate the description.

Definition 1: The cost of the group D_i is defined as $cost(D_i) = (\sum_{j=1}^{N_i} f_j^{(i)})(\sum_{j=1}^{N_i} z_j^{(i)})$, where variables N_i, $f_j^{(i)}$, and $z_j^{(i)}$ are listed in Table 1.

Definition 2: $MaxPQ$ is defined as a max priority queue in which each element belongs to a subset of D. In such a way that when one is to remove an element from $MaxPQ$, it will return the element with the maximal cost. The method of returning the with the maximal cost, $D_{\max}$, is defined as $ReturnMax(MaxPQ)$.

The algorithmic form of DRP is outlined as follows.

Algorithm DRP:
Input: the number of broadcast channels K and the database D with $|D| = N$, where the data item $d_1, d_2, ..., d_N$ have been sorted in descending order according to br_i, $1 \le i \le N$.
Output: The grouping result $D_1, D_2, .., D_K$.
begin
1. Create a max priority queue $MaxPQ$ and insert the element D.
2. **while** (Number of elements in $MaxPQ < K$)
3. $D_{\max} = ReturnMax(MaxPQ)$;
4. remove element $D_{\max}$ from $MaxPQ$;
5. $\{D', D''\}$=Partition($D_{\max}$);
6. insert D' and D'' into $MaxPQ$;
7. **end while**
8. **return** all elements in $MaxPQ$;
end

Procedure Partition(D_x)
Input: the selected item set D_x with $|D_x| = N_x$, where the data item $d_{x1}, d_{x2}, ..., d_{xN_x}$ have been sorted in descending order according to br_i, $1 \le i \le N_x$.
Output: two disjoint item sets D'_x and D''_x, where $D'_x + D''_x = D_x$
begin
1. Determine the index p, $1 \le p < N_x$, such that $cost(D'_x) + cost(D''_x)$ is minimized.
Where $D'_x = \{d_{x1}, d_{x2}, ...d_{xp}\}$, $D''_x = \{d_{xp+1}, d_{xp+2}, ..., d_{xN_x}\}$;
2. **return** the corresponding D'_x and D''_x;
end

To generate a broadcast program by algorithm DRP. All the data items in the database D are sorted according to the *br* value in descending order. Initially, the database D is viewed as an element stored in a max priority queue $MaxPQ$. In each iteration, $MaxPQ$ returns the element with the largest cost. The returned element will be split by the procedure Partition(D_x) into two elements, which are disjoint subsets of the original element. The two elements are re-inserted into $MaxPQ$. At the end of each iteration, the number of the elements in $MaxPQ$ increases by 1. Note that the procedure Partition(D_x) determines the most suitable point p to partition the input sequence $d_{x1}, d_{x2}, ..., d_{xN_x}$ into two subsequence $d_{x1}, d_{x2}, ...d_{xp}$, and $d_{xp+1}, d_{xp+2}, ..., d_{xN_x}$, so that the summation of the cost of the two sequences is minimized. Algorithm DRP terminates when the number of elements in $MaxPQ$

reaches K. The channel allocation is performed according to the elements in $MaxPQ$. i.e., each items in the same element is allocated to the same channel.

Lemma 1 *The complexity of DRP can be expressed by* $K \cdot (O(K \log K) + O(N))$*, where* K *is the channel number, and* N *is the size of the broadcast database.*

Proof. *The complexity of returning the element with the largest cost for* $MaxPQ$ *is* $K \cdot O(K \log K)$*, while the complexity of finding the most suitable point to partition is* $K \cdot O(N)$*.* ■

Example 1: Consider the broadcast profile shown in Table 2. A database D containing 15 data items needs to be broadcast via 5 broadcast channels. i.e., $N = 15$, $K = 5$. Before algorithm DRP executes, the data items are sorted according to their *br* values in descending order. In the beginning, there is only one data set D contained in the max priority queue $MaxPQ$, as shown in Table 3(a). The cost of the data set can be calculated from Definition 1, i.e., *cost*$(D) = 135.60$. In each iteration, the data set with maximum cost is removed from $MaxPQ$ and two disjoint data sets are inserted into $MaxPQ$. The best partition point is determined by Procedure Partition(D_x). In Table 3(b), the best partition point lies between d_{12} and d_{10}. The original data set is replaced with two disjoint item sets with their corresponding cost 29.04 and 28.62, respectively. Likewise, in the next iteration, $MaxPQ$ remove the data set with *cost* $=$ 29.04, and inserted two disjoint subsets of the removed data set, as shown in Table 3(c). Algorithm DRP terminates when the number of the elements in $MaxPQ$ reaches 5. Table 3(d) shows the grouping result. Finally, the broadcast program is generated according to the grouping result. i.e., items in the same group will be put in the same channel.

3.2 Cost-Diminishing Selection

CDS (Cost-Diminishing Selection) is a tuning mechanism, which is used to refine a certain grouping result so that the local optimum can be achieved. The intuition of CDS is to consider the amount of cost reduction when moving a data item from one group to another. By collecting the reduction information of all possible moving operations, the best data item with its corresponding movement is determined. The total cost diminishes after each iteration. CDS terminates when the local optimum is achieved. i.e., no data item can be moved from one group to another with the reduction of the cost. Several special terms are also defined before describing mechanism CDS.

Definition 3: The aggregate frequency of an item set D_i, denoted by F_i, is defined as the summation of the access frequency of all data items in D_i. i.e., $F_i = \sum_{j=1}^{N_i} f_j^{(i)}$.

Definition 4: The aggregate size of an item set D_i, denoted by Z_i, is defined as the summation of the item size of all data items in D_i. i.e., $Z_i = \sum_{j=1}^{N_i} z_j^{(i)}$.

Consider a data item d_x with its access frequency f_x and item size z_x. Let d_x be moved from D_p to D_q. The total cost before the moving operation can be derived from Eq. (3) as:

$$c_{before} = \sum_{i=1}^{K} [(\sum_{j=1}^{N_i} f_j^{(i)})(\sum_{j=1}^{N_i} z_j^{(i)})] = \sum_{i=1}^{K} (F_i Z_i).$$

Also, the total cost after the moving operation can be derived as:

$$\begin{aligned} c_{after} &= [\sum_{i=1, i \neq p,q}^{K} (F_i Z_i)] + (F_p - f_x)(Z_p - z_x) \\ &\quad + (F_q + f_x)(Z_q + z_x). \end{aligned}$$

Therefore, the $cost\ reduction$ Δc, which represents the amount of reduced cost after the moving operation is performed, is obtained as:

$$\begin{aligned} \Delta c &= c_{before} - c_{after} \\ &= [F_p Z_p + F_q Z_q] - [(F_p - f_x)(Z_p - z_x) \\ &\quad + (F_q + f_x)(Z_q + z_x)] \\ &= f_x(Z_p - Z_q) + z_x(F_p - F_q) - 2 f_x z_x. \quad (4) \end{aligned}$$

Using the result in Eq. (4), we are able to estimate the cost reduction before a moving operation of a data item is performed. Therefore, we can select the best movement by examining the Δc of all possibilities. The algorithmic form of mechanism CDS is outlined below.

Mechanism CDS:
Input: the initial grouping result $D_1, D_2, .., D_K$
Output: the local optimal grouping result: $D_{opt1}, D_{opt2}, .., D_{optK}$
begin
while (true)
  $\Delta c_{\max} = 0, d'_{orig} = Null,$
  $D'_{orig} = Null, D'_{dest} = Null;$
  for (p=1;p≤ K;p++)
    $D_{orig} = D_p;$
    for (x=1;x≤ N_p;x++)
      $d_{orig} = d_x^{(p)};$
      for (q=1;q≤ K and q≠p;q++)
        $D_{dest} = D_q;$
        $\Delta c = GetReduction(d_{orig}, D_{orig}, D_{dest});$
        if ($\Delta c > \Delta c_{\max}$)

Table 2. Profile of the Broadcast Database

Item	Freq.	Size	Item	Freq.	size	Item	Freq.	Size
d_1	0.2374	21.18	d_6	0.0566	2.49	d_{11}	0.0349	30.62
d_2	0.1363	4.77	d_7	0.0500	17.51	d_{12}	0.0325	4.09
d_3	0.0986	3.59	d_8	0.0450	10.86	d_{13}	0.0305	5.33
d_4	0.0783	15.34	d_9	0.0409	1.02	d_{14}	0.0287	7.74
d_5	0.0655	2.91	d_{10}	0.0376	6.41	d_{15}	0.0272	1.74

Table 3. Example of the Algorithm DRP

Group	Member	Cost
1	$\{d_9d_2d_3d_6d_5d_{15}d_1d_{12}$ $d_{10}d_{13}d_4d_8d_{14}d_7d_{11}\}$	*135.60

(a) The initial state of DRP

Group	Member	Cost
1	$\{d_9d_2d_3d_6d_5d_{15}d_1d_{12}\}$	*29.04
2	$\{d_{10}d_{13}d_4d_8d_{14}d_7d_{11}\}$	28.62

(b) The first iteration of DRP

Group	Member	Cost
1	$\{d_9d_2d_3d_6d_5d_{15}\}$	7.02
2	$\{d_1d_{12}\}$	6.82
3	$\{d_{10}d_{13}d_4d_8d_{14}d_7d_{11}\}$	*28.62

(c) The second iteration of DRP

Group	Member	Cost
1	$\{d_9d_2d_3\}$	2.59
2	$\{d_6d_5d_{15}\}$	1.07
3	$\{d_1d_{12}\}$	6.82
4	$\{d_{10}d_{13}d_4d_8\}$	7.26
5	$\{d_{14}d_7d_{11}\}$	6.35

(d) The grouping result of DRP

12. $\Delta c_{\max} = \Delta c, d'_{orig} = d_{orig}$;
13. $D'_{orig} = D_{orig}, D'_{dest} = D_{dest}$;
14. **end if**
15. **end for**
16. **end for**
17. **end for**
18. **if** ($\Delta c_{\max} == 0$)
19. **break**
20. **end if**
21. move d'_{orig} from D'_{orig} to D'_{dest};
22. **end while**
23. **return** $D_1, D_2, ..., D_K$;

end

Given the grouping result, the goal of mechanism CDS is to find out the best moving operation which can result in the maximum cost reduction. According to the Eq. (4), we can estimate the cost reduction Δc of each possible moving operation without performing it. The results of all possible moving operations can be examined without moving the data items back and forth. Each moving operation contains three parameters: the original group D_{orig}, the destination group D_{dest}, and the data item d_{orig} which is moving from D_{orig} to D_{dest}. In each iteration of CDS, from line 4 to line 17, the best moving operation is selected after all possibilities are considered. At the end of the iteration, as shown in line 21, the, best moving operation is performed and the grouping result is updated. The next iteration is executed according to the updated grouping result. The total cost diminishes after each iteration is executed. Mechanism CDS terminates when no moving operation can result in the cost reduction. The local optimum is thus achieved. Note that in each iteration, the complexity of mechanism CDS is $O(K^2N)$, where K and N represent the number of broadcast channel and the number of disseminated items. Mechanism CDS has two advantageous features. First, from the viewpoint of the complexity, mechanism CDS can reach the local optimum in polynomial time. Moreover, the iterative property makes mechanism CDS give a progressive performance since the moving operation with maximum cost reduction is selected at the end of each iteration.

Example 2: Table 4 illustrates the procedure of mechanism CDS. Consider the grouping results of the example 1, as shown in Table 4(a). The initial cost, denoted as c_{init}, is 24.09. The goal of each iteration of mechanism CDS is to find the moving operation for a data item from one group to another, with the maximum cost reduction. In Table 4(b), according to the formula in Eq. (4), we find that moving d_{10} from group 4 to group 2 will result in the $\Delta c_{\max} = 0.95$. At the end of the iteration, such a moving operation is performed. After that, in the next iteration, the grouping result in the previous iteration is considered. Table 4(c) shows the grouping result in which d_{12} is moved

Table 4. Example of mechanism CDS

Group	Member	c_{init}
1	$\{d_9, d_2, d_3\}$	24.09
2	$\{d_6, d_5, d_{15}\}$	
3	$\{d_1, d_{12}\}$	
4	$\{d_{10}, d_{13}, d_4, d_8\}$	
5	$\{d_{14}, d_7, d_{11}\}$	

(a) The initial state of CDS

Group	Member	
1	$\{d_9, d_2, d_3\}$	c_{before} 24.09
2	$\{d_6, d_5, d_{15}, d_{10}\}$	c_{after}
3	$\{d_1, d_{12}\}$	23.13
4	$\{d_{13}, d_4, d_8\}$	Δc
5	$\{d_{14}, d_7, d_{11}\}$	0.95

(b) The first iteration of CDS

Group	Member	
1	$\{d_9, d_2, d_3\}$	c_{before} 23.13
2	$\{d_6, d_5, d_{15}, d_{10}, d_{12}\}$	c_{after}
3	$\{d_1\}$	22.68
4	$\{d_{13}, d_4, d_8\}$	Δc
5	$\{d_{14}, d_7, d_{11}\}$	0.45

(c) The second iteration of CDS

Group	Member	
1	$\{d_9, d_2, d_3, d_6\}$	c_{before} 22.29
2	$\{d_5, d_{15}, d_{10}, d_{12}, d_{14}\}$	c_{after}
3	$\{d_1\}$	22.29
4	$\{d_{13}, d_4, d_8\}$	Δc
5	$\{d_7, d_{11}\}$	0

(d) The grouping result of CDS

from group 3 to group 2, and the maximum cost reduction $\Delta c_{\max} = 0.45$ is achieved. Mechanism CDS continues until $\Delta c_{\max} = 0$, which means no more data item can move from one group to another. Therefore, the local optimum is achieved with cost 22.29.

4 Experimental Results

To verify the quality and the practicability of the proposed algorithm DRP and mechanism CDS, several experiments are conducted to observe the performances. In each experiment, we investigate the result of DRP-CDS which represents algorithm DRP combined with CDS tuning mechanism. For the comparing purposes, we implement an algorithm GOPT, which can reach the global optimum, by using the concept of Genetic Algorithm[2] [5][6]. The detail of the algorithm GOPT is omitted for interest of space. Moreover, algorithm VF^K [14], which is proposed for the conventional broadcast environment, is also used for the comparison purposes.

4.1 Simulation Environment

Table 5 lists the simulation parameters. The access frequencies of the data items are generated by Zipf distribution [17] $f_i = (\frac{1}{i})^\theta / \sum_{j=1}^{N} (\frac{1}{j})^\theta$, where θ is a skewness parameter and $1 \le i \le N$. The size of each data item is represented by 10^ϕ units, where the value ϕ is uniformly distributed over the interval $[0, \Phi]$. The value Φ determines the exponent range of the item sizes. We name it the $diversity\ parameter$. More specifically, in Table 5, the value of Φ varies from 0 to 3. The case of $\Phi = 0$ implies that all data items are of the same size (i.e., 1 unit). When $\Phi = 3$, the size of each data item is located in the interval $[10^0, 10^3]$ units. Therefore, the diversity of the item size increases as the value Φ increases.

[2] Although GOPT has a very good performance, since it is based on Genetic Algorithm, the value derived is still viewed as a suboptimum.

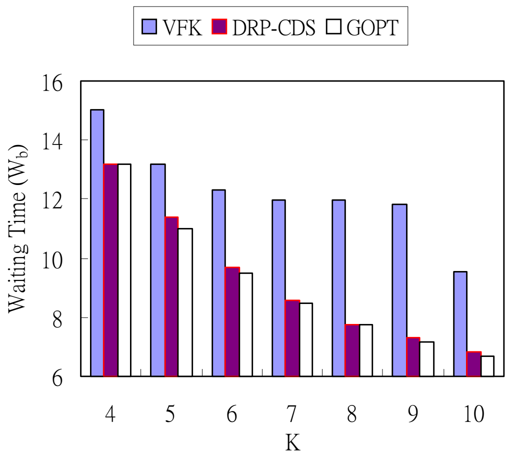

Figure 2. The channel number v.s. the average waiting time

4.2 Scalability Analysis

The first experiment discusses the scalability issue of the proposed approach. There are two scalability parameters, the number of channels, K, and the number of the broadcast items, N. The amount of waiting time W_b is used to measure the effectiveness of broadcast program

Table 5. Parameter used in the simulation

Parameters	Values
Number of broadcast items (N)	$60 \sim 180$
Number of channels (K)	$4 \sim 10$
Diversity Parameter (Φ)	$0 \sim 3$
Skewness Parameter (θ)	$0.4 \sim 1.6$
Channel bandwidth	10 (unit size/sec)

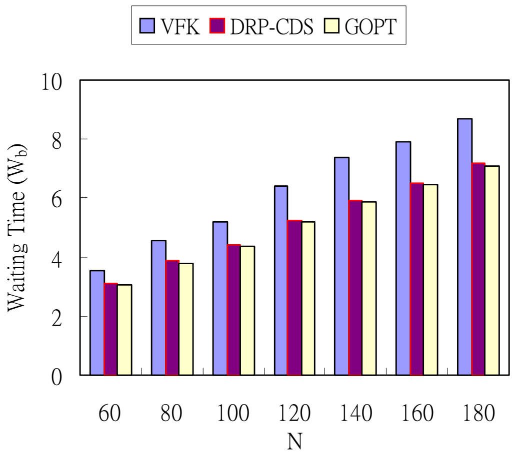

Figure 3. The number of broadcast items v.s. the average waiting time

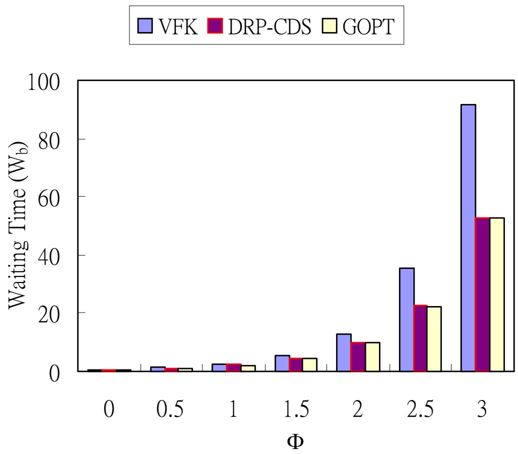

Figure 4. The diversity v.s. the average waiting time

generation. First, in order to evaluate the impact of channel numbers, the value K varies from 4 to 10, as can be shown in Figure 2. We observe that the average waiting time decreases as the value K increases for all listed approaches. It is found that VF^K suffers from the scalability issues since the discrepancy of VF^K compared to GOPT increases when K increases. The performance of DRP can be improved to local optimal by adopting mechanism CDS. By observing the bars of DRP-CDS, the error compared to the optimal waiting time is about 3% in most of the situations. The error will be even lower as the increase of the value K because the increase of the channel numbers is helpful of distributing the data items. There is an another interesting observation. DRP has excellent performance without adopting CDS when $K = 4$ and $K = 8$. That is, the improvement of DRP-CDS is subtle compared to DRP when $K = 2^n$, where n belongs to integers. It is because DRP partitions one channel into two to minimize the average waiting time of these two channels. When the channel number can be expressed as $K = 2^n$, where n belongs to integers, the data items can be evenly distributed into K groups. Next, if we fix the value of K and vary the number of N, the observed performances is depicted in Figure 3. When the number of the broadcast items increases, the average waiting time for each approach increases because each channel should disseminate more items. Here, the proposed approach still results in better qualities than algorithm VF^K. However, the increasing value N degrades the performance of DRP if CDS is not adopted by the previous two algorithms. When the system needs to broadcast larger numbers of data items, it is necessary to adopt mechanism CDS to achieve the effectiveness. By observing the performances of DRP-CDS, it is still very close to the optimum. The qualities of DRP-CDS is not affected as the value N increases. Therefore, it is shown that mechanism CDS is scalable so that the quality can be maintained when larger numbers of data items are broadcast.

4.3 Diversity Analysis

In this experiment, we discuss the diversity issue of the proposed DRP-CDS. As shown in Figure 4, when the

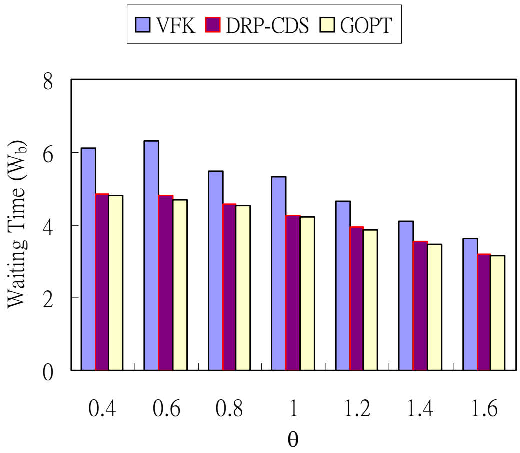

Figure 5. The skewness v.s. the average waiting time

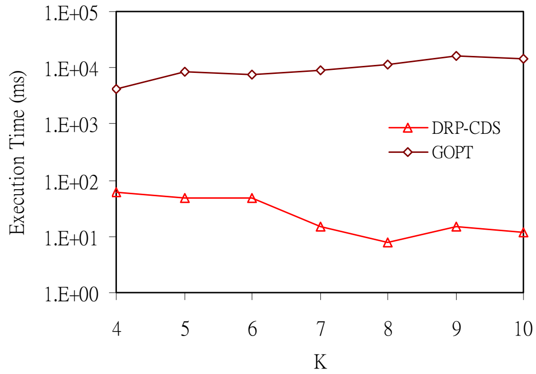

Figure 6. The channel number v.s. the execution time

diversity increases, the average waiting time of each approach increases drastically. The reason is that the average size of data item increases in highly diverse environment. Since the bandwidth still remains the same, it takes more time to disseminate each data item. In Figure 4, the performances of VF^K and DRP-CDS are very close to the optimum when the value of Φ is low. Algorithm VF^K, which is an algorithm suitable for the conventional broadcast environment, only considers the access frequency of each data item. In the diverse broadcast environment, VF^K suffers from the the effectiveness issue and results in poor performance. In the case of high Φ, it is obvious that our approach outperforms algorithm VF^K. The local optimal points that DRP-CDS achieves are still very close to the global optimal points that GOPT achieves. This experiment shows the necessity of developing algorithms suitable for the diverse broadcast environment because the algorithm used in the conventional environment is no longer suitable in this environment.

4.4 Skewness Analysis

As depicted in Figure 5, this experiment shows the waiting time of each approach as the value of skewness parameter θ varies. A larger value of θ implies the more skewed access frequencies of the data items. There are several observations made. First, the average waiting time of each approach decreases as the increase of the skewness parameter. This is because that the degree of request locality is high when the access frequency is highly skewed. The system can put the data items with higher access frequencies together into a channel with fewer items in order to reduce the average waiting time. Second, the discrepancy of the proposed approach compared to GOPT becomes subtle. For example, there is an error of 0.04 between the DRP-CDS and the GOPT when $\theta = 0.4$. The error is reduced to 0.005 when $\theta = 1.6$. The reason is that under the same diversity, the increase of the skewness will make the access frequencies of data items dominate the channel allocation. The channel allocation will be more precise when one feature (i.e., access frequency) of each item is more important than the other (i.e., item size).

4.5 Complexity Analysis

The final experiment discusses the complexity among both approaches. Since we implement these approaches by Java language and execute the programs under the same system platform, the execution time of the program will reflect the relative complexity. We use mini-second as the unit of execution time. Since the parameter θ and Φ do not affect the complexity, in this experiment, we only consider the parameter K and N. Figure 6 shows the execution time of each approach as the number of channel K varies, while Figure 7 depicts the execution times as the increase of N. Compared to GOPT, algorithm DRP-CDS spends much less time generating broadcast programs. The execution time of the GOPT increases as K or N increase. Note that the execution time of GOPT is more sensitive to N than to K. The reason is as follows. The GOPT is implemented by the Genetic Algorithm. The increase of N will increase the length of each chromosome, while the the increase of K only changes the variety of the gene value in a chromosome. By observing the above two figures, we find that although the GOPT can achieve the global optimal solutions, it is computationally prohibitive. On the

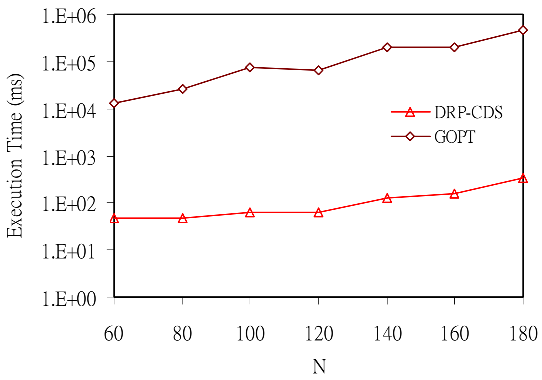

Figure 7. The number of broadcast items v.s. the execution time

other hand, our approach DRP-CDS can result in quality close to the optimum with significantly shorter execution time. Therefore, the proposed DRP-CDS is very suitable for generating broadcast programs practically.

5 Conclusion

In this paper, we focus on generating broadcast programs in a *diverse data broadcasting* environment. First, we propose an effective algorithm DRP. After that, we also use a mechanism called CDS to refine the result of DRP to the local optimum. In order to verify the performance, several experiments are conducted. These experiments consider the important issues such as scalability, diversity and the complexity. From the experimental results, we prove that the proposed DRP-CDS is very practical in performing an effective channel allocation efficiently in a *diverse broadcast* environment.

Acknowledgements

The work was supported in part by the National Science Council of Taiwan, R.O.C., under Contracts NSC93-2752-E-002-006-PAE.

References

[1] S. Acharya, R. Alonso, M. J. Franklin, and S. B. Zdonik. Broadcast disks: Data management for asymmetric communications environments. In *Proceedings of the 1995 ACM International Conference on Management of Data*, pages 199–210, May 1995.

[2] S. Acharya and S. Muthukrishuan. Scheduling on-demand broadcasts: New metrics and algorithm. In *ACM MOBICOM*, 1998.

[3] C. C. Aggarwal, J. L. Wolf, and P. S. Yu. A permutation-based pyramid broadcasting scheme for video-on-demand systems. In *IEEE Proceedings of Multimedia*, 1996.

[4] D. Barbará. Mobile computing and database - a survey. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):108–117, 1999.

[5] D. E. Goldberg. Genetic algorithm in search, optimization and machine learning. *Addison-Wesley Publishing*, 1989.

[6] J. H. Holland. Adaption in natural and artificial systems. *University of Michigan Press, Ann Arbor*, 1975.

[7] C.-H. Hsu, G. Lee, and A. L. P. Chen. A near optimal algorithm for generating broadcast programs on multiple channels. In *Proceedings of the 10th ACM International Conference on Information and Knowledge Management*, pages 303–309, November 2001.

[8] J.-L. Huang and M.-S. Chen. Broadcast program generation for unordered queries with data replication. In *Proceedings of ACM SAC'03*, March 2003.

[9] J.-L. Huang and M.-S. Chen. Broadcasting dependent data for ordered queries without replication in a multi-channel mobile environment. In *Proceedings of the 19th IEEE International Conference on Data Engineering*, March 2003.

[10] J.-L. Huang and M.-S. Chen. Dependent data broadcasting for unordered queries in a multiple channel mobile environment. *IEEE Trans. on Knowledge and Data Engineering*, 16(6), Jun. 2004.

[11] T. Imielinski, S. Viswanathan, and B. R. Badrinath. Data on air: Organization and access. *IEEE Transactions on Knowledge and Data Engineering*, 9(3):353–372, May/June 1997.

[12] L.-S. Juhn and L.-M. Tseng. Fast data broadcasting and receiving scheme for popular video service. *IEEE Tran. on Broadcasting*, 44(1), March 1998.

[13] W. Liu and J. Y. B. Lee. Constrained consonant-a generalized periodic broadcasting scheme for large scale video streaming. In *IEEE Intern'l Conf. on Multimedia and Expo (ICME-03)*, June 2003.

[14] W.-C. Peng and M.-S. Chen. Efficient channel allocation tree generation for data broadcasting in a mobile computing environment. *Wireless Networks*, 9(2):117–129, 2003.

[15] M. Satyanarayanan. Pervasive computing: Vision and challenges. *IEEE Personal Communications*, 8(4):10–17, Aug. 2001.

[16] J. Xu, X. Tang, and D. Lee. Performance analysis of location-dependent cache invalidation schemes for mobile environments. *IEEE Transactions on Knowledge and Data Engineering*, 15(2):474–488, Mar/Apr 2003.

[17] G. K. Zipf. *Human Behaviour and the Principle of Least Effort*. Addison-Wesley, Reading, MA, 1949.