On Channel Allocation for Heterogeneous Data Broadcasting

Hsiao-Ping Tsai, *Student Member*, *IEEE*, Hao-Ping Hung, and Ming-Syan Chen, *Fellow*, *IEEE*

Abstract—In recent years, data broadcasting has become a promising technique in designing a mobile information system with power conservation, high scalability, and high bandwidth utilization. However, prior research topics in data broadcasting are mainly based on the assumption that the disseminated data items are of the same size. Motivated by the fact that various kinds of data objects may be disseminated in advanced information systems, we explore in this paper the problem of generating broadcast programs in a *heterogeneous* data broadcasting environment, in which disseminated data items can be of different sizes. Given the broadcast database and the number of channels, we first derive the analytical model of the heterogeneous data broadcasting to obtain the average waiting time of mobile users and prove the allocation problem as an NP-complete problem. In order to solve such problem, we propose a two-phase architecture to perform channel allocation. Algorithm Dimension Reduction Partitioning (DRP) is employed to perform *rough* allocation to derive the satisfactory solutions, whereas mechanism Cost-Diminishing Movement Selection (CDMS) is used for *fine* allocation to achieve local optimum solutions. In addition to the two-phase architecture, we also propose algorithm GA-CDMS according to the concept of *hybrid genetic algorithm* for comparison purposes. GA-CDMS can perform global search more accurately and efficiently than the conventional genetic algorithm GA, and the suboptimum that GA-CDMS achieves will be very close to the optimal solution. In order to validate the two-phase allocation algorithm DRP-CDMS, several experiments are conducted. In these experiments, we consider the important issues such as accuracy, scalability, diversity and efficiency. From the experimental results, we show that the proposed two-phase channel allocation is very practical in performing an effective channel allocation with high efficiency in a heterogeneous broadcasting environment.

Index Terms—Data broadcasting, data dissemination, multiple channels, heterogeneous data size.

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 [1], location-dependent data management [2], pervasive computing [3], and so on.

Data broadcasting is a well-known technique to disseminate data items from an information system to mobile users. In a broadcast-based (i.e., push-based) information system, as shown in Fig. 1, the server generates a broadcast program by collecting the access patterns of mobile users and broadcasts data items periodically via multiple channels. The period of each broadcast channel is viewed as a *broadcast cycle*. To retrieve a data item, users with mobile devices should listen and wait for the data of interest to appear on the broadcast channel. The average waiting time is composed of two components: the *probing time* and the *downloading 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 *probing time*, W_{probe}, is the time that a user should wait until the item of interest appears in the channel, and thus, $W_{probe} = \frac{1}{2}(broadcast\ cycle\ time) = \frac{Nz}{2b}$. The *downloading 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 average waiting time can be formulated as $W_b = W_{probe} + W_{download} = \frac{Nz}{2b} + \frac{z}{b}$.

There are many research topics in generating the broadcast programs to broadcast data items via multiple broadcast channels [4], [5], [6], [7], [8], [9], [10]. A *flat* broadcast program, in which the items are allocated to broadcast channels with equal appearance frequencies, is a straightforward way for broadcasting. However, this approach is ineffective since the expected waiting time of data items with different access probabilities is the same. In order to overcome the effectiveness problem, approaches are proposed in [4], [8], and [9] to generate broadcast programs in which the expected waiting time of popular data items (i.e., with higher access probabilities) is shorter than that of unpopular data items (i.e., with lower access probabilities). In addition to providing basic services for mobile users, there are also many extensions of the broadcast technique. The works in [5] and [7] focus on broadcasting dependent

- *H.-P. Tsai is with the Institute of Information Science at Academia Sinica, 128 Academia Rd., Sec. 2, Nankang, Taipei 115, Taiwan, ROC. E-mail: hptsai@arbor.ee.ntu.edu.tw.*
- *H.-P. Hung is with Cyberlink Corp., 15F, No. 100, Ming-Chiuan Road, Hsin-Tien City, Taipei, Hsien, Taiwan, ROC. E-mail: hphung@arbor.ee.ntu.edu.tw.*
- *M.-S. Chen is with the Research Center for Information Technology Innovation at Academia Sinica, 128 Academia Rd., Sec. 2, Nankang, Taipei 115, and also with the Department of Electrical Engineering, National Taiwan University, No. 1, Sec. 4, Roosevelt Road, Taipei, Taiwan, ROC. E-mail: mschen@cc.ee.ntu.edu.tw.*

Manuscript received 30 Dec. 2007; revised 20 May 2008; accepted 20 Aug. 2008; published online 15 Oct. 2008.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2007-12-0399. Digital Object Identifier no. 10.1109/TMC.2008.148.

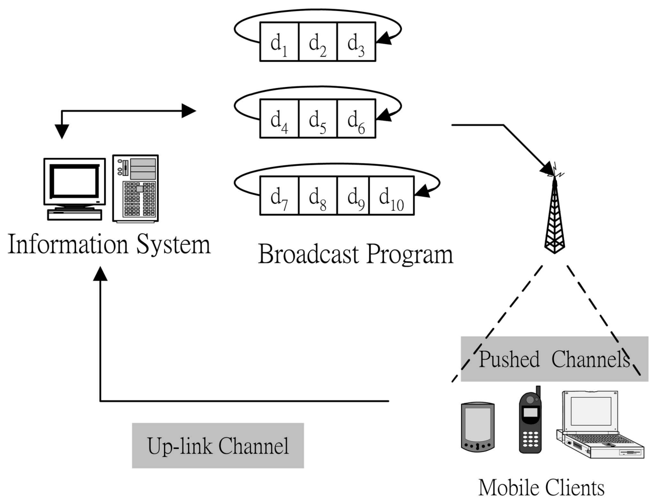


Fig. 1. The architecture of broadcast-based data dissemination.

data for ordered and unordered queries. Moreover, the broadcast program allows a data item to appear in different broadcast channels simultaneously. Such a replication issue is addressed in [11].

Most of the broadcasting schemes in prior research are based on the assumption that each disseminated items are of the same size. However, in the advanced communication environment with larger bandwidth, the mobile users can use their devices with higher capability to access various kinds of information such as still image, video, and audio. Therefore, the data items with different sizes will inevitably be disseminated in a modern information system. Since the effect of item size is neglected, the dissemination policies based on conventional models may suffer from effectiveness issues in a modern information system. To differentiate our work from those in the conventional broadcasting environment, we use the term *heterogeneous* data broadcasting[1] to describe the broadcast environment in which the data items with different sizes are disseminated. In such environments, each data item contains two attributes, the access probability and the item size. In this paper, we focus on generating broadcast programs in a heterogeneous data broadcasting environment. Although the broadcast program may also be generated according to the ready-made techniques proposed for video-on-demand service such as *pyramid broadcasting* [13], [14], [15], this technique neglects the access probability of each video object. Moreover, *pyramid broadcasting* only performs well in playing the video sequence continuously. These intrinsic discrepancies make it difficult to apply *pyramid broadcasting* to disseminating the items in the *heterogeneous data broadcasting* environment. In generating the broadcast program in the heterogeneous environment, there are three major contributions in this paper: First, we derive the analytical model of average waiting time in *heterogeneous* broadcasting environments. Next, we propose an efficient two-phase architecture to generate a local-optimal broadcast program in a heterogeneous environment. Moreover, a *hybrid genetic* algorithm, which achieves global optimization, is also proposed for comparison purposes.

1. The term "heterogeneous" is first used in [12] to indicate the items with different sizes.

In this paper, we first derive the analytical model of heterogeneous data broadcasting in a multiple-channel environment. Given a specific channel allocation, the average waiting time can be determined according to the access probability and the size of each data item. By observing the analytical model, the allocation problem will be transformed to a *grouping problem* with a specific cost function. We next propose an efficient method to perform channel allocation (i.e., grouping) for each data item. This method is named DRP, standing for Dimension Reduction Partitioning. Algorithm DRP is a top-down group-splitting approach, in which the two-dimensional grouping problem is simplified as a one-dimensional partitioning problem. In addition, a Cost-Diminishing Movement Selection (CDMS) mechanism is proposed to refine the effectiveness to the local optimum. In essence, CDMS is executed iteratively. In each iteration, mechanism CDMS checks the possible moving operations for a certain data item from one group to another and determines the best choice. Mechanism CDMS will keep executing until the local optimum is achieved, where no moving operation will lead to the cost reduction. More specifically, what we propose in this paper can be viewed as a two-phase allocation scheme. Algorithm DRP provides *rough* allocation, which achieves satisfactory quality, whereas mechanism CDMS provides *fine* allocation, which achieves the local optimum by refining the results of the rough allocation. Moreover, owing to the iterative property of CDMS, there will be a progressive improvement before the local optimum is reached.

In addition to the above two-phase channel allocation algorithms, we also propose an algorithm called GA-CDMS for comparison purposes. Algorithm GA-CDMS is a *hybrid genetic algorithm*, which combines the concept of genetic algorithm [16], [17] and the proposed mechanism CDMS. Unlike the two-phase algorithm, GA-CDMS can perform stochastic search globally in optimizing the broadcast program. GA-CDMS solves the heterogeneous broadcasting problem in two levels. In the chromosome level, solutions of channel allocation are encoded as chromosomes with corresponding *fitness* values. By executing the operations such as *selection*, *crossover*, and *mutation*, the chromosomes with higher *fitness* values will have better chances to survive and evolve after each generation. As for the solution level, chromosomes will be decoded as solutions of channel allocation. Mechanism CDMS will refine each solution until the local optimum is reached. The most advantageous feature is that the search space of GA-CDMS is bound by all local optimal solutions. Unlike the conventional genetic algorithm GA in which the searching space contains all possible solutions, GA-CDMS will reach more preferable allocation results within a lower execution time.

To verify the effectiveness and efficiency of the two-phase channel allocation algorithms, several experiments are conducted. First, we compare the performances of the hybrid genetic algorithm with the simple genetic algorithm. By inspecting the expected waiting time and the standard deviation from the outcomes with different initial conditions, we observe that the hybrid genetic algorithm will almost achieve the global optimal solution, whereas the simple genetic algorithm is still bound by a certain local optimum. Next, we also analyze some significant issues such as scalability, diversity, and skewness. During the experiments, in addition to comparing the two-phase channel allocation to the *hybrid genetic algorithm*, the algorithm adopted in the *homogeneous* broadcasting environment is also included for comparison. We will show that the local optimum reached by the two-phase algorithm is very close to the solution quality of the *hybrid genetic algorithm*. In the final experiment, by measuring the execution time, it is shown that the proposed DRP-CDMS is more efficient than GA-CDMS. The experimental results will show that the proposed two-phase allocation algorithms have high quality and high efficiency in the *heterogeneous* data broadcasting environment.

The rest of our paper is outlined as follows: In Section 2, the related works will be reviewed. In Section 3, we will derive the analytical model for heterogeneous data broadcasting and formulate the allocation problem. In Section 4, we will describe the proposed two-phase allocation algorithm DRP-CDMS. In Section 5, a hybrid genetic algorithm GA-CDMS will also be proposed for comparison purposes. The experimental results will be shown in Section 6, and finally, this paper will be concluded with Section 7.

2 Related Works

In the mobile computing environment characterized by the asymmetric communication and the limited power of the client device, the *pushed-based* dissemination [18] provides scalable and preferable information service. By collecting the access patterns, the server can provide multilevel nonuniform dissemination according to the popularity of each data item. Thus, in a *homogeneous* broadcasting environment, popular data will be put in the channel containing fewer items and be broadcast more frequently.

In Peng and Chen's work [8], the problem of generating a broadcast program was modeled as constructing a hierarchical tree. By exploiting the feature of variant-fanout, algorithm VF^K was proposed to achieve near-optimal allocation with low complexity. On the other hand, Hsu et al. also developed an effective algorithm in [4] to generate a near-optimal broadcast program. Different from algorithm VF^K, what Hsu et al. proposed is based on Wong's theorem [19]. In Wong's theorem, considering data items of equal size, the average waiting time can be minimized if each data item is equally spaced and the equation $p_i/p_j = \sqrt{f_i}/\sqrt{f_j}$ is satisfied, where p_i represents the appearance probability of a certain item in the broadcast channel, whereas f_i denotes the access probability of the item. As for the optimal broadcast program in the *homogeneous* environment, it can be generated according to the concept of *dynamic programming* [20], the related detail can be found in [9].

In some applications, the data items that a mobile user requests may have dependency with each other. In order to broadcast dependent data, Martinez et al. designed a framework to generate the broadcast program for a single channel based on the concept of a *hill climbing algorithm* [21]. In Martinez et al.'s work, the numbers of initial random permutations that specify the broadcast sequences are generated. According to the initial permutation, the neighboring permutations will be searched repeatedly until no improvement is made in a certain number of attempts. The authors in [22] proposed a *greedy* algorithm to solve the dependent data broadcasting problem for a single channel. In view of the poor performance of adopting a *greedy* algorithm for multiple channels, Huang and Chen adopted the concept of a *genetic algorithm* to solve the dependent data broadcasting problem for multiple channels [11], [5], [7].

The broadcast techniques mentioned above are based on the assumption that the disseminated data items are of the same size, i.e., *homogeneous* data broadcasting. Compared to *homogeneous* data broadcasting, there is much less prior research in the field of *heterogeneous* data broadcasting. In [23], Su and Tassiulas proposed a scheduling policy for generating a *heterogeneous* broadcast program in a *single push-based* channel. The scheduling decision is based on the *elapsed time* since the last transmission of each item. In [24] and [25], Hammed and Vaidya proposed the square root rule (SSR) as a theoretical guideline in handling heterogeneous data broadcasting in a single channel. According to SSR, in the optimal allocation, every item should be spaced equally with spacing proportional to the square root of its size and inversely proportional to the square root of its access frequency. In addition to *pushed-based* dissemination, there were several scheduling policies in a *pull-based* channel proposed [12], [26], [27], [28]. Different from the *pushed-based* dissemination, the *pull-based* dissemination is also referred to as *on-demand* broadcasting [29]. Such technique was initiated in [26], in which the *homogeneous* environment was considered. In [12] and [28], the scheduling policies of a *heterogeneous* environment were studied based on the metric of *stretch*, which is defined to be the ratio of the response time of a request to its service time. In [27], a simplified scheduling algorithm was proposed to suit the corresponding transcoding proxy system.

3 Model of Heterogeneous Broadcasting

3.1 Analytical Model

Given a database D with its size $|D| = N$ and the number of channels K, a *broadcast program* stands for an allocation of data items in D into K channels. Each channel, denoted by c_i, contains an item set D_i with its size $|D_i| = N_i$, where $\sum_{i=1}^{K} N_i = N$, and $D_i \cap D_j = \{\emptyset\}$ if $i \neq j$. In a heterogeneous broadcasting environment, different from the conventional (i.e., homogeneous) one, the data may have different item sizes. Therefore, a data item $d_j^{(i)}$, which represents the jth data item in c_i, contains two attributes, the access probability $f_j^{(i)}$ 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 average waiting time of each channel. 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 *probing time* of c_i is $(\sum_{j=1}^{N_i} z_j^{(i)})/(2b)$. In addition to the *probing time*, it takes $z_j^{(i)}/b$ to download the data item $d_j^{(i)}$.

TABLE 1
Description of the Symbols

Symbol	Description
K	the number of the broadcast channels ($K \geq 2$)
c_i	the i-th broadcast channel ($0 < i \leq K$)
D	the database of the broadcast data items
N	the number of the broadcast data items
D_i	the item set of the data items allocated to c_i
N_i	the number of the data items allocated to c_i
$d_j^{(i)}$	the j-th data item in c_i ($0 < j \leq N_i$)
$z_j^{(i)}$	the size of the j-th data item in c_i ($0 < j \leq N_i$)
$f_j^{(i)}$	the access probability of the j-th data item in c_i ($0 < j \leq N_i$)
b	tshe bandwidth of each broadcast channel

Therefore, $W_j^{(i)}$, the waiting time of the data item $d_j^{(i)}$ in the channel c_i, can be derived as

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

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

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

Therefore, the average 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,

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

3.2 Problem Formulation

In this paper, given a specific database, we seek 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 (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 *probing time*, while the term $\sum_{i=1}^{K}\sum_{j=1}^{N_i} f_j^{(i)} z_j^{(i)}$ represents the effect of the *downloading time*. The second term can be viewed as the summation of the product value of the access probability and the size of *all* data items 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 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 (2) as follows:

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

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 as the following grouping problem: *given the database D, group the data items in D into K different clusters so that the value of cost can be minimized*.

The grouping problem is polynomial time solvable when the item sizes are homogeneous [9]. In the heterogeneous case, the grouping problem is similar to the bin-packing problem that packs objects of different size into a minimal number of bins of fixed capacity. However, unlike the bin-packing problem, our problem targets to minimize the average waiting time with a fixed channel number and unlimited channel capacity. We prove that the grouping problem is at least an NP-complete problem in Theorem 1.

Theorem 1. *The grouping problem of generating a broadcast program in a heterogeneous data broadcasting environment is NP-complete.*

Proof. First, it is obvious that such grouping problem is NP. Next, consider a special case when we set $f_j^{(i)} = z_j^{(i)} = s_j^{(i)}$. We can rewrite the expression in (3) as

$$cost = \sum_{i=1}^{K}\left[\left(\sum_{j=1}^{N_i} s_j^{(i)}\right)\left(\sum_{j=1}^{N_i} s_j^{(i)}\right)\right] = \sum_{i=1}^{K}\left[\left(\sum_{j=1}^{N_i} s_j^{(i)}\right)^2\right].$$

The special case is exactly equal to the Minimum Sum-of-Squares problem [30]. Since the Minimum Sum-of-Squares problem is an NP-complete problem, the grouping problem in generating a broadcast program in a heterogeneous data broadcasting environment is equivalent to an NP-complete problem. More specifically, given the existing Minimum Sum-of-Squares problem, which is known to be NP-complete, consider the *reduction function* $f(x) = x$. Since each instance of the Minimum Sum-of-Squares problem can be transformed to an instance of the grouping problem, the NP-completeness can thus be proved. □

In this paper, to facilitate the description, we employ a simplified broadcast model. However, via some extensions, our algorithms can still be practical in a real data broadcasting environment. The assumptions and extensions are described as follows:

1. We assume that each mobile user knows which item is broadcast in which channel so that he/she can switch to a specific channel to download the item of interest. This can be achieved by broadcasting the index structure [6], [31], [32] in addition to the data items. Note that the index items can be either broadcast via separate channels or inserted in the time slot between two consecutive items.
2. Similar to the prior research [4], [8], [9] in scheduling mobile data items, we assume that in the broadcast

Algorithm DRP
Input: The number of broadcast channels K
The database D with $|D|=N$. Note that the items $d_1 \sim d_N$ are sorted in descending order according to the benefit ratio br_i
Output: The grouping result $D_1, D_2, ..., D_K$
Create an empty priority queue $MaxPQ$ and insert the element D
while number of elments in $MaxPQ < K$
$D_{\max}$ = ReturnMax($MaxPQ$)
Remove $D_{\max}$ from $MaxPQ$
$\{D', D''\}$ = Partition($D_{\max}$)
Insert D' and D'' into $MaxPQ$
Return all elements in $MaxPQ$

Procedure Partition
Input: The selected itemset D_x with $|D_x| = \text{N}$, where the item $d_{x1} \sim d_{xN}$ have been sorted in a descending order according to br_i, $1 \le i \le N_x$
Output: Two disjoint itemsets D'_x and D''_x, where $D'_x + D''_x = D_x$
Determine the index p, $1 \le p < N_x$, such that $cost(D'_x) + cost(D''_x)$ is minimized, where $D'_x = \{d_{x1}, d_{x2}, ..., d_{xp}\}$ and $D''_x = \{d_{xp+1}, d_{xp+2}, ..., d_{xN}\}$
Return D'_x and D''_x

Fig. 2. Algorithmic form of DRP.

program, the data items are without replication. It is noted that broadcasting the items periodically virtually has the same effect as replicating the items in broadcast channels.

3. The access probabilities of the items are collected by the uplink channels [12], [26]. If the uplink channel is not available, the access probabilities can be estimated via the approaches proposed in [33] and [34].
4. Like [18], we assume that the access probabilities are static, i.e., the access probability of an item will not change over time. For dynamic access probabilities, in addition to regenerating the broadcasting programs once the access probabilities change, we can employ the algorithms in [35] and [36] to generate broadcast programs.

4 Two-Phase Allocation

4.1 Dimension Reduction Partitioning

To allocate the items in the broadcast database D into K channels, we first propose an efficient algorithm DRP. 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 number of groups reaches K. Since each item contains two attributes, item size and access probability, 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 the access probability divided by the item size, i.e., $br_i = \frac{f_i}{z_i}$. The reason for using the benefit ratio to describe the feature of a data item is that in the heterogeneous 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 a 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 priority queue in which each element belongs to a subset of D. When one is to remove an element from $MaxPQ$, it will return the element with the maximal cost. The method of returning the group with the maximal cost, $D_{\max}$, is defined as $ReturnMax(MaxPQ)$.*

The algorithmic form of DRP is outlined in Fig. 2. To generate a broadcast program via algorithm DRP, all the data items in the database D are first sorted according to the br value in descending order. Initially, the database D is viewed as an element stored in a priority queue $MaxPQ$. In each iteration, $MaxPQ$ returns the element with the largest cost, i.e., $D_{\max}$. The returned element will be split by the procedure Partition into two elements, which are disjoint subsets of $D_{\max}$. The two elements are reinserted into $MaxPQ$. At the end of each iteration, the number of the elements in $MaxPQ$ is increased by 1. Note that the Partition procedure determines the most suitable point p to partition the input sequence $d_{x1}, d_{x2}, \ldots, d_{xN_x}$ into two subsequences $d_{x1}, d_{x2}, \ldots, d_{xp}$ and $d_{xp+1}, d_{xp+2}, \ldots, d_{xN_x}$ 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 allocation result will be

TABLE 2
Profile of the Broadcast Database

Item	Prob.	Size	Item	Prob.	size	Item	Prob.	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	$\{d_9 d_2 d_3 d_6 d_5 d_{15} d_1 d_{12}$ $d_{10} d_{13} d_4 d_8 d_{14} d_7 d_{11}\}$	*135.60

(a)

Group	Member	Cost
1	$\{d_9 d_2 d_3 d_6 d_5 d_{15} d_1 d_{12}\}$	*29.04
2	$\{d_{10} d_{13} d_4 d_8 d_{14} d_7 d_{11}\}$	28.62

(b)

Group	Member	Cost
1	$\{d_9 d_2 d_3 d_6 d_5 d_{15}\}$	7.02
2	$\{d_1 d_{12}\}$	6.82
3	$\{d_{10} d_{13} d_4 d_8 d_{14} d_7 d_{11}\}$	*28.62

(c)

Group	Member	Cost
1	$\{d_9 d_2 d_3\}$	2.59
2	$\{d_6 d_5 d_{15}\}$	1.07
3	$\{d_1 d_{12}\}$	6.82
4	$\{d_{10} d_{13} d_4 d_8\}$	7.26
5	$\{d_{14} d_7 d_{11}\}$	6.35

(d)

(a) The initial state of DRP. (b) The first iteration of DRP. (c) The second iteration of DRP. (d) The grouping result of DRP.

obtained from the elements in $MaxPQ$, i.e., each item in the same group is allocated to the same channel.

Lemma 1. *The complexity of DRP can be expressed by* $O(N \log N) + O(K \log K) + O(KN)$.

Proof. We first note that the complexity of sorting all items according to the benefit ratio is $O(N \log N)$. Next, when the DRP algorithm is executed, there are at most K iterations. In each iteration, the complexity of returning $D_{\max}$ is $O(\log K)$, whereas the complexity of finding the most suitable partition point is $O(N)$. Therefore, the overall scheduling cost is $O(K \log K) + O(KN)$. □

Example 1. Consider the broadcast profile shown in Table 2. A database containing 15 items should be broadcast via five channels, i.e., $N = 15$ and $K = 5$. Before algorithm DRP is executed, 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 priority queue $MaxPQ$, as shown in Table 3a. 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 the maximum cost is removed from $MaxPQ$, and two disjoint data sets are inserted into $MaxPQ$. The best partition point is determined by Procedure $\mathrm{Partition}(D_x)$. In Table 3b, 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, $MaxPQ$ removes the data set with $cost = 29.04$ and inserts two disjoint subsets of the removed data set, as shown in Table 3c. Algorithm DRP terminates when the number of the elements in $MaxPQ$ reaches five. Table 3d 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.

4.2 Cost-Diminishing Movement Selection

CDMS is a tuning mechanism, which is used to refine the allocation result of algorithm DRP so that the local optimum can be reached. The basic idea of CDMS is inspired by the observation that the overall cost may either increase or decrease when moving a data item from one group to another in the allocation result of algorithm DRP. By collecting the reduction information of all possible moving operations, we can select the best one, i.e., the moving operation leading to the maximum cost reduction, to perform. Mechanism CDMS will be executed iteratively, and the total cost is guaranteed to diminish after each iteration. CDMS 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. To facilitate the description, several special terms are defined.

Definition 3. *The aggregate probability of an item set D_i, denoted by F_i, is defined as the summation of the access probability 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 probability 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 (3) as

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

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

Mechanism CDMS
Input: The initial grouping result $D_1, D_2, ..., D_K$
Output: The local optimal grouping result $D_{opt1}, D_{opt2}, ..., D_{optK}$
while true
  $D'_{orig} = null$, $D'_{dest} = null$
  $\Delta c_{\max} = 0$, $d'_{orig} = null$
  for $p = 1$ to K
    $D_{orig} = D_p$
    for $x = 1$ to N_p
      $d_{orig} = d_x^{(p)}$
      for $q = 1$ to K, $q \neq p$
        $D_{dest} = D_q$
        $\Delta c = GetReduction(d_{orig}, D_{orig}, D_{dest})$
        if $\Delta c > \Delta c_{\max}$
          $d'_{orig} = d_{orig}$, $\Delta c_{\max} = \Delta c$, $D'_{orig} = D_{orig}$, $D'_{dest} = D_{dest}$
  if $\Delta c_{\max} == 0$
    break while
  Move d'_{orig} from D'_{orig} to D'_{dest}
Return $D_1, D_2, ..., D_K$

Fig. 3. Algorithmic form of the CDMS mechanism.

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

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

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

Using the result in (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 CDMS is outlined in Fig. 3.

Given the allocation result, the goal of mechanism CDMS is to find out the best moving operation that can result in the maximum cost reduction. According to (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 moved from D_{orig} to D_{dest}. In each iteration of CDMS, the best moving operation is selected after all possibilities are considered. At the end of the iteration, the best moving operation is performed, and the allocation result is updated. The next iteration is executed according to the updated allocation result. The total cost diminishes after each iteration is executed. Mechanism CDMS 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 CDMS is $O(K^2 N)$, where K and N represent the number of broadcast channels and the number of disseminated items. Mechanism CDMS has two advantageous features. First, from the viewpoint of the complexity, mechanism CDMS can reach the local optimum in polynomial time since Δc must converge to zero within constant iterations. Also, the iterative property, in which the moving operation with the maximum cost reduction is selected, will make mechanism CDMS give a progressive performance.

Example 2. Table 4 illustrates the procedure of mechanism CDMS. Consider the grouping result of the Example 1, as shown in Table 4a. The initial cost, denoted as c_{init}, is 24.09. The goal of each iteration of mechanism CDMS is to find the moving operation for a data item from one group to another, with the maximum cost reduction. In Table 4b, according to the formula in (4), we find that moving d_{10} from group 4 to group 2 will result in $\Delta c_{\max} = 0.95$. 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 4c shows the grouping result in which d_{12} is moved from group 3 to group 2, and the maximum cost reduction $\Delta c_{\max} = 0.45$ is achieved. Mechanism CDMS continues until $\Delta c_{\max} = 0$, which means that no more data item can move from one group to another. Therefore, the local optimum is achieved with cost 22.29.

5 Design of Hybrid Genetic Algorithm

In this section, we develop another algorithm called GA-CDMS to generate broadcast programs in a *heterogeneous* broadcasting environment. Compared to DRP-CDMS, GA-CDMS can overcome the problem of local optimum. GA-CDMS is a *hybrid genetic algorithm*, which combines

TABLE 4
Example of Mechanism CDMS

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)

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	$\{d_{13}, d_4, d_8\}$	$\Delta c_{\max}$
5	$\{d_{14}, d_7, d_{11}\}$	0.95

(b)

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	$\{d_{13}, d_4, d_8\}$	$\Delta c_{\max}$
5	$\{d_{14}, d_7, d_{11}\}$	0.45

(c)

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	$\{d_{13}, d_4, d_8\}$	$\Delta c_{\max}$
5	$\{d_7, d_{11}\}$	0

(d)

(a) The initial state of CDMS. (b) The first iteration of CDMS. (c) The second iteration of CDMS. (d) The grouping result of CDMS.

the conventional genetic algorithm and the proposed mechanism CDMS. Although genetic algorithms have proved to be a versatile and effective approach for solving optimization problems [37], [38], there are still many situations in which the simple genetic algorithm may result in poor performances such as slow convergence and local optimum (or premature) [39], [40], [41]. In order to overcome such problems, the proposed *hybrid genetic algorithm* GA-CDMS incorporates the local optimization (i.e., procedure CDMS) in the loop of recombination and selection. With the hybrid approach, mechanism CDMS is applied to each newly generated chromosome to move it to a local optimum state before being injected into the population. Specifically, we use the genetic algorithm, on one hand, to perform global exploration, and adopt mechanism CDMS, on the other hand, to perform local exploitation around chromosomes. Since the searching space of GA-CDMS is bound by all local optimal solutions instead of all possible solutions, scheme GA-CDMS outperforms the conventional GA in accuracy and efficiency. In the following sections, we will describe the design of GA-CDMS.

5.1 Chromosome Representation and Fitness Evaluation

According to the problem formulation in Section 2, the channel allocation problem can be transformed into a *grouping problem* in which a database D with size N is grouped into K different clusters. In order to satisfy the feasibility, legality, and uniqueness of mapping a solution (i.e., an allocation result) to a chromosome, we represent a chromosome as an integer array with length N. Therefore, a chromosome contains N genes. The *locus* (i.e., position of a gene) represents the specific data item in D. The *allele* (i.e., value of a gene) identifies the group in which the data item is put. It is obvious that each solution corresponds to a distinct chromosome and vice versa. There is a *fitness* value for each chromosome. *Fitness* is the measurement of the quality of a chromosome. A genetic algorithm is designed to search the chromosome with the highest *fitness*. Since the goal of the *grouping problem* is to minimize the value of *cost* in (3), the *fitness* function of a chromosome Ch is defined as $Fitness(Ch) = \frac{1}{cost(S)}$; S is the corresponding broadcast program of Ch.

Example 3. Fig. 4 gives an example of mapping a solution of channel allocation to its corresponding chromosome representation. Consider a broadcast program that allocates data items $d_1 \sim d_3$ to channel c_1, $d_4 \sim d_6$ to c_2, and $d_7 \sim d_{10}$ to c_3. According to the above encoding rule, in the corresponding chromosome, the values of the first three genes that represent the items $d_1 \sim d_3$ are set to be 1, while the next three *alleles* are set to be 2, and the final four are set to be 3.

5.2 Crossover and Mutation

Crossover and *mutation* are the most frequently used operators in the genetic algorithm. The *crossover* operator progressively constructs near-optimal solutions, whereas the *mutation* operator is used to increase population diversity. These operators allow the search process to explore the neighboring regions or to reach further promising regions. The *crossover* operator achieves the recombination of the selected individuals by combining the segments belonging to the two different chromosomes in parents. There are several ways to perform *crossover*. In this

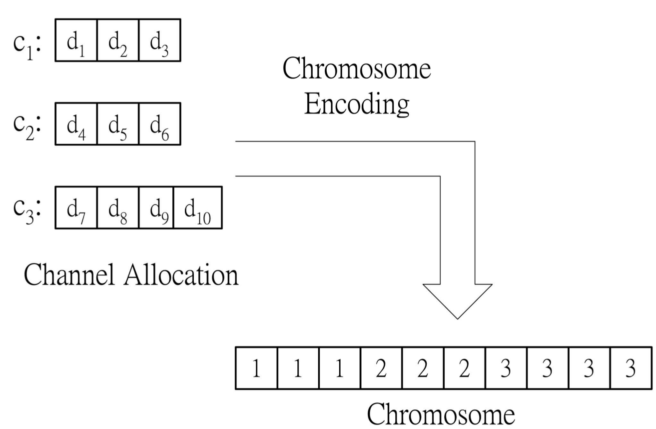


Fig. 4. The mapping of a solution and its corresponding chromosome.

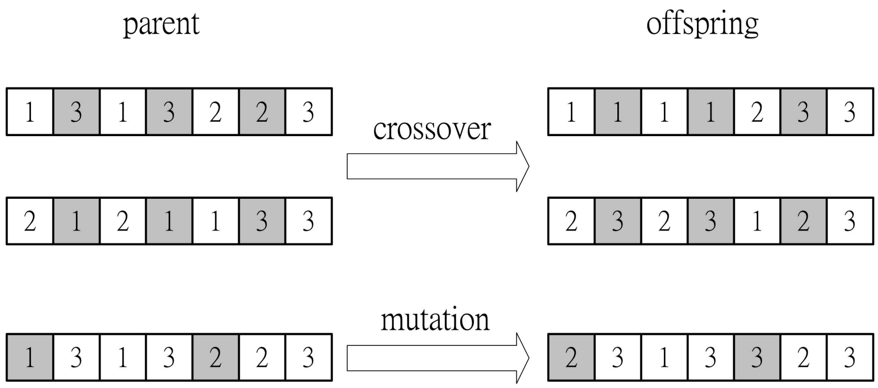


Fig. 5. The illustration of crossover and mutation.

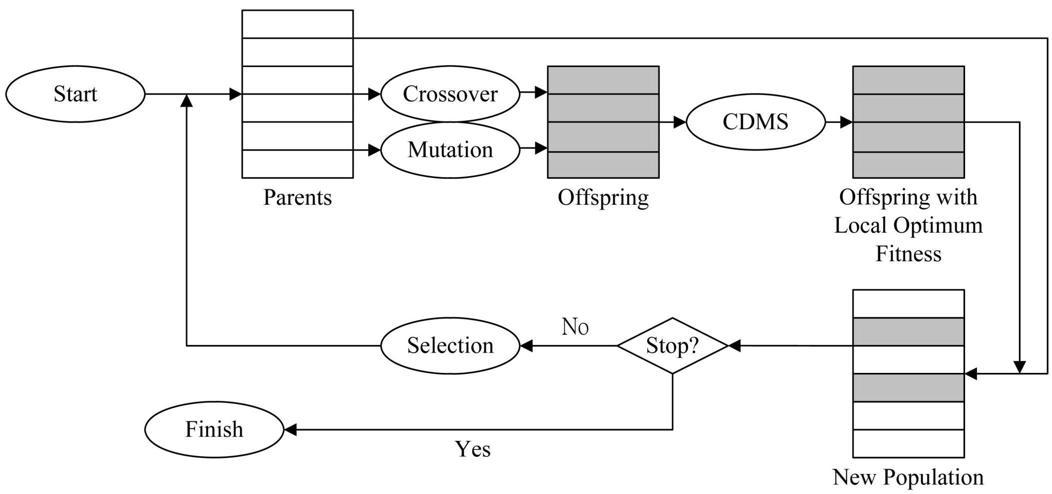


Fig. 6. The architecture of GA-CDMS.

paper, the *uniform crossover* [42] is adopted since it has the advantage of combining features irrespective of their relative position. The *mutation* operator is to change the value of a single gene within a chromosome and assures that the full range of *allele* values is available to the search. In this paper, the *mutation* operator allows multiple genes to randomly change their *alleles* with a uniform distribution [43]. Fig. 5 illustrates the operation of *crossover* and *mutation*.

5.3 Architecture of GA-CDMS

The architecture of the hybrid genetic algorithm GA-CDMS is presented in Fig. 6. Initially, several randomly generated solutions (i.e., grouping results) are encoded as the chromosomes of the initial parents. By means of the *crossover* and *mutation* operation, the offspring population, which contains the chromosomes with better characteristics than those in parents, is generated. After that, mechanism CDMS is adopted to refine each chromosome in offspring as a local optimal one. Regarding the chromosomes in both offspring and parents as candidates, a new population is reproduced by 1) picking out the chromosomes with higher fitness and 2) eliminating others so as to keep the population size constant. In the selection phase, well-performed chromosomes will have better chances to survive. We use *roulette wheel* method [44] to determine the distribution of the probability of being selected. Note that each chromosome is allowed to be selected repeatedly. The population of the selected chromosomes will be viewed as the parents in the next generation. GA-CDMS terminates when all of the chromosomes in selecting candidates converge to be identical. The corresponding grouping result will hopefully be a global optimum solution. The major advantage of algorithm GA-CDMS is that the searching space is bound by all local optimal solutions instead of all possible solutions. Therefore, algorithm GA-CDMS will be more accurate and more efficient than GA.

6 Experimental Evaluation

In this section, the performances of our two-phase allocation and hybrid genetic algorithm are inspected through a simulation study. The simulation environment will be introduced in Section 6.1. In Section 6.2, we compare our hybrid genetic algorithm GA-CDMS with a simple genetic algorithm GA by analyzing their statistics. In Section 6.3, we discuss the effectiveness of the relevant algorithms by measuring the average waiting time W_b. Finally, we will also discuss the efficiency of these approaches in Section 6.4.

TABLE 5
Parameter Used in the Simulation

	Small Range Config.		Wide Range Config.	
Parameters	Range	Default Value	Range	Default Value
Number of broadcast items (N)	$60 \sim 160$	100	$200 \sim 1000$	600
Number of channels (K)	$4 \sim 9$	6	$4 \sim 12$	6
Diversity Parameter (Φ)	$0 \sim 3$	2	$0 \sim 3$	2
Skewness Parameter (θ)	$0.4 \sim 1.4$	1.0	$0.4 \sim 1.4$	1.0
Channel bandwidth	–	10 (unit size/sec)	–	10 (unit size/sec)
Size of population	–	100	–	100
Crossover probability	–	0.1	–	0.1
Mutation probability	–	0.1	–	0.1

6.1 Simulation Environment

Table 5 summarizes the definitions for primary simulation parameters. For the small-range configuration,[2] the number of channels is varied from four to nine, whereas the number of broadcast items is varied from 60 to 180. For the wide-range configuration, the number of channels is varied from 4 to 12, whereas the number of broadcast items is varied from 200 to 1,000. To reflect the access skew in the database system, the access probabilities of queries are generated by Zipf distribution [45] $f_i = (\frac{1}{i})^\theta / \sum_{j=1}^{N} (\frac{1}{j})^\theta$, where θ is a *skewness parameter*, and $1 \leq i \leq N$. Note that a large θ indicates the highly skewed access patterns in the mobile computing environment. When $\theta = 0$, the access probabilities of the queries are uniformly distributed. By default, the value of θ is set to be 1 since it is observed in [46] that θ is usually larger than 1 for busy websites. The size of each data item is represented by 10^ϕ units, where the value of ϕ is uniformly distributed over the interval $[0, \Phi]$. The value of 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 zero to three. The case $\Phi = 0$ 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. Note that the diversity of the item size increases as the value of Φ increases. In the design of the *hybrid genetic algorithm* GA-CDMS, the population size (i.e., the number of chromosomes) in each generation is set to be 100. The crossover probability and the mutation probability are set to be 0.1.

During the experiments, we inspect the performances of the following algorithms:

1. DRP, the DRP algorithm without optimization,
2. DRP-CDMS, the DRP algorithm with CDMS as the optimization procedure,
3. GA-CDMS, the proposed hybrid genetic algorithm,
4. VF^K [8], a greedy heuristic algorithm for *homogeneous* data items,
5. DP [4], a near-optimal dynamic programming algorithm for *homogeneous* data items, and
6. Greedy [9], a greedy algorithm for *homogeneous* data items.

2. Since the genetic algorithm for global optimization suffers from extremely high computing complexity, we only compare our algorithms with the genetic algorithm to validate our algorithms on small data sets.

Note that possible extensions for heterogeneous data allocation are mentioned in [4] and [9]. We modify GREEDY and DP accordingly for the *heterogeneous* data broadcasting environment, i.e., ordering items by their br values in descending order before partitioning.

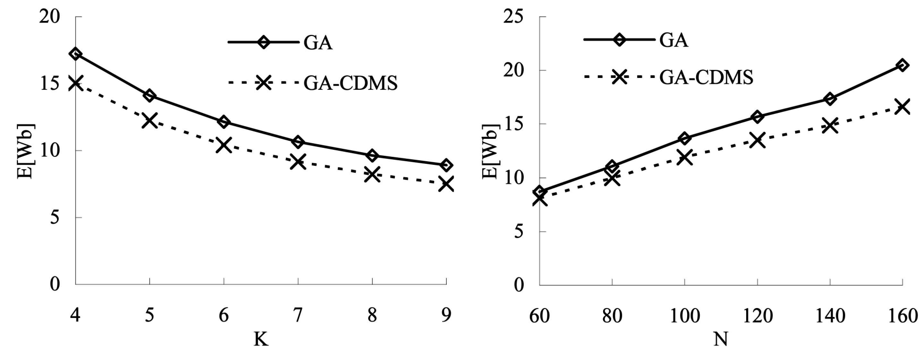
6.2 Performance of Hybrid Genetic Algorithm

We first compare the performance of GA-CDMS to the conventional genetic algorithm GA. Unlike GA-CDMS, algorithm GA inherits the architecture in Fig. 6 except that the block of CDMS is removed. During the experiment, given a specific broadcast profile, algorithms GA and GA-CDMS are executed 20 times with different initial conditions (i.e., different chromosomes in the initial parents in Fig. 6). Since the genetic algorithm performs stochastic search, different initial conditions may result in different solutions. For the broadcast programs generated by these 20 experiments, we measure the expected value and the standard deviation of waiting time W_b. In Fig. 7, which shows the expected value, we can observe that the proposed GA-CDMS can generate the broadcast programs with a lower average waiting time. On the other hand, in Fig. 8, which depicts the standard deviation, the most significant observation is that the standard deviation of GA-CDMS is very close to zero, which is much lower than that of GA. This phenomenon implies that given different initial conditions, algorithm GA-CDMS will almost result in the same channel allocation, while the allocation result of GA will depend on the initial condition. Since GA-CDMS performs global search, the probability that GA-CDMS results in a certain local optimum is quite low. Therefore, it is very possible that the suboptimum generated by GA-CDMS is an optimal solution.

6.3 Effectiveness Analysis

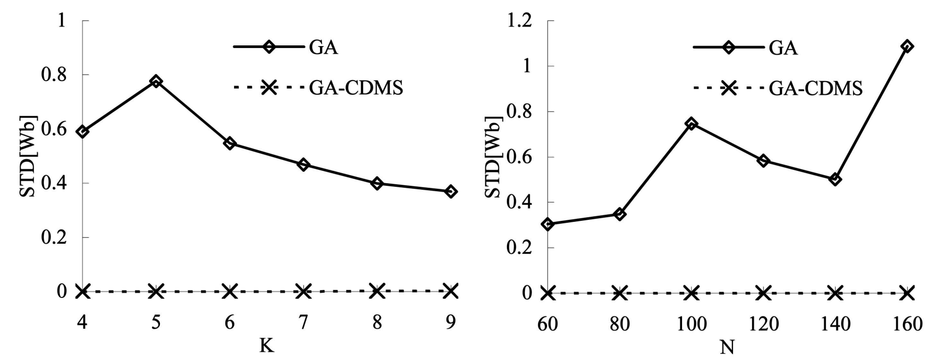
We next discuss the effectiveness of the relevant algorithms. The quality of the broadcast program is determined by four parameters: the diversity parameter (Φ), the skewness parameter (θ), the number of broadcast channels (K), and the size of the broadcast database (N). Also, note that the quality of a broadcast program is reflected by the average waiting time W_b. In the following, we inspect the quality of the broadcast program with different parameters varied.

Fig. 9a depicts the effect of the parameters K and N on the average waiting time of algorithm DRP-CDMS. Intuitively, the average waiting time increases as the size of the broadcast database increases and decreases as the



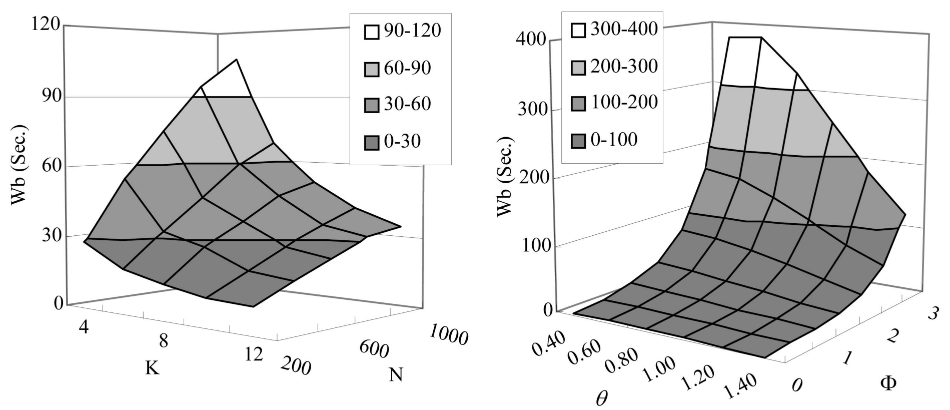
(a) (b)

Fig. 7. The expected value of waiting time (a) as K varied and (b) as N varied (small-range configuration). (a) Expected Wb with K varied. (b) Expected Wb with N varied.



(a) (b)

Fig. 8. The standard deviation of waiting time (a) as K varied and (b) as N varied (small-range configuration). (a) Standard deviation with K varied. (b) Standard deviation with N varied.

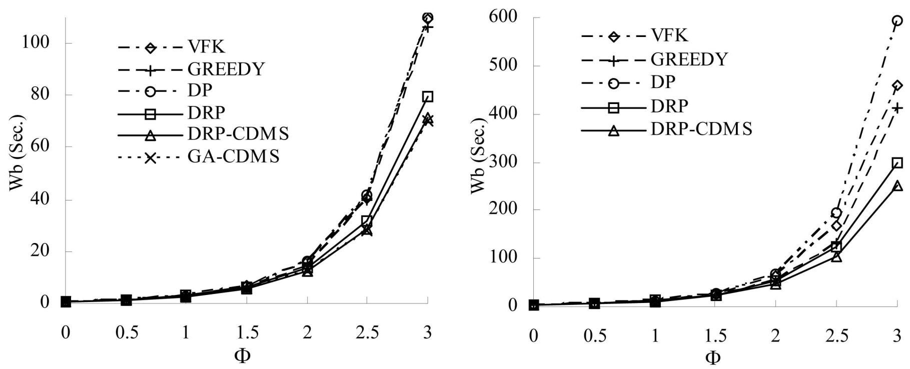


(a) (b)

Fig. 9. The expected value of waiting time (a) as N and K varied and (b) as θ and Φ varied. (DRP-CDMS, wide-range configuration.) (a) Wb versus K and N. (b) Wb versus θ and Φ.

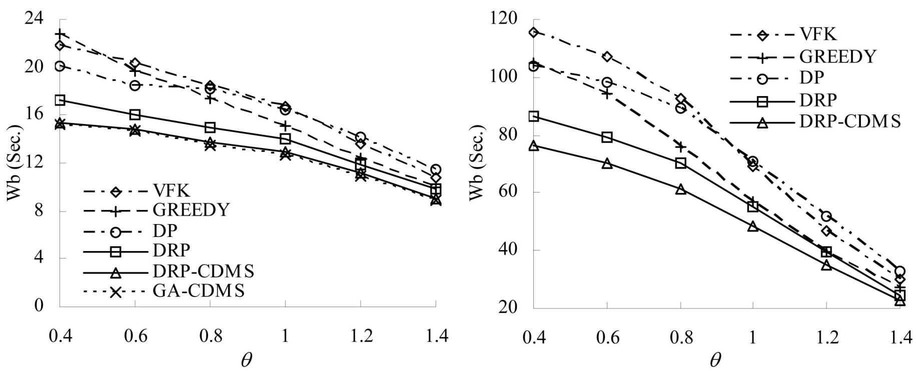
number of broadcast channels increases. Fig. 9b depicts the effect of the parameters θ and Φ on the average waiting time. In the case of high diversity, the average waiting time still decreases quickly when data skewness grows.

Next, we discuss the diversity issue of the proposed algorithms. As shown in Fig. 10, when the diversity increases, the average waiting time of each approach increases drastically. The reason is that the average size of a data item increases in highly diverse environments. Since the bandwidth still remains the same, it takes more time to disseminate each data item. In Fig. 10, the performances of VF^K, GREEDY, DP, DRP, and DRP-CDMS are very close to GA-CDMS when the value of Φ is low. Algorithm VF^K, which is an algorithm suitable for the *homogeneous* broadcasting environment, only considers the access probability of each data item. DP and GREEDY are originally designed for the *homogeneous* broadcasting environment. Possible extensions for heterogeneous data allocation are mentioned without experiments. We modify their algorithms accordingly in our experiments, i.e.,



(a) (b)

Fig. 10. The expected value of waiting time as Φ varied. (a) Small-range configuration. (b) Wide-range configuration.



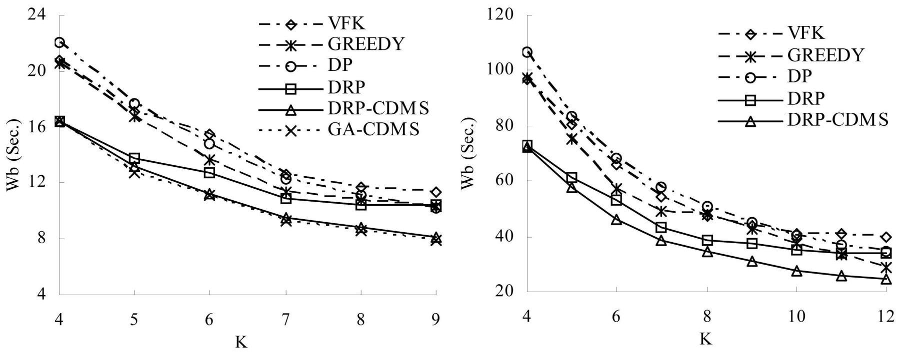
(a) (b)

Fig. 11. The expected value of waiting time as θ varied. (a) Small-range configuration. (b) Wide-range configuration.

ordering items by their *br* values instead of access probability. In the *heterogeneous* broadcast environment, VF^K, DP, and GREEDY suffer from effectiveness issues and result in poor performance. In the case of a high Φ, it is obvious that our approaches outperform algorithm VF^K, GREEDY, and DP. Moreover, when the diversity of page sizes is large, e.g., $\Phi = 3$, the average waiting time of DRP-CDMS is about 10 percent and 15 percent lower than that of DRP for the small-range and wide-range configurations, respectively. As shown in Fig. 10a, the local optimal points that DRP-CDMS achieves are still very close to the qualities that GA-CDMS achieves. This experiment shows the necessity of developing algorithms suitable for the *heterogeneous* data broadcasting environment because the algorithm used in the *homogeneous* environment is no longer suitable in this environment.

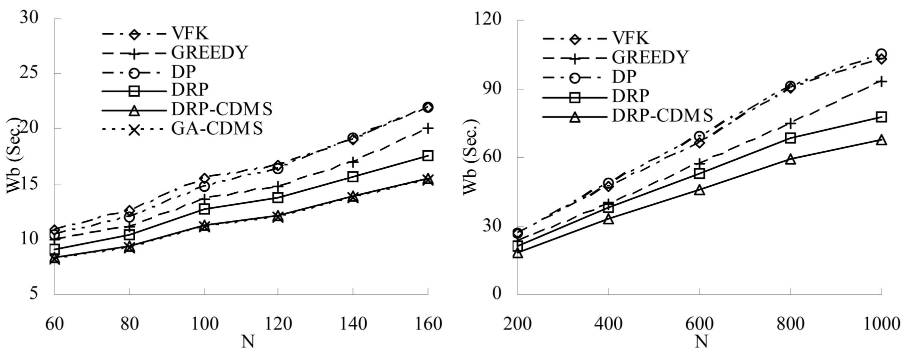
As depicted in Fig. 11, this experiment shows the average waiting time of each approach as the skewness parameter θ varies. A larger value of θ implies the more skewed access probabilities of the data items. The average waiting time of GREEDY is close to DRP in the case of a high θ. In addition, although DP achieves a near-optimal allocation in a homogeneous environment, Greedy outperforms DP in the heterogeneous environment when the access probability is more skewed. There are several observations made. First, the average waiting time of each approach decreases as the skewness parameter increases. This is because of the fact that the degree of request locality is high when the access probability is highly skewed. The system can put the data items with higher access probabilities together into a channel with fewer items in order to reduce the average waiting time. Second, the discrepancy of the proposed approaches compared to GA-CDMS becomes subtle. The reason is that under the same diversity, the increase of the skewness will make the access probabilities of data items dominate the channel allocation. The channel allocation will be more precise when one feature (i.e., access probability) of each item is more important than the other (i.e., item size).

Fig. 12 depicts the effect of the parameter K. In this experiment, we observe that the average waiting time decreases as the value of K increases for all listed approaches. It is found that VF^K suffers from scalability issues since the discrepancy of VF^K compared to GA-CDMS increases when K increases. Compared to VF^K, the DRP algorithm can achieve satisfactory quality. The performance of DRP can be refined to the local optimum by employing the CDMS mechanism. By observing the average waiting time, the error between DRP-CDMS and GA-CDMS is less than 3 percent in most of the situations. The error will become even insignificant as the value of K increases because the increase of the number of channels is helpful for distributing the data items. There is another interesting observation. DRP has excellent performance



(a) (b)

Fig. 12. The expected value of waiting time as K varied. (a) Small-range configuration. (b) Wide-range configuration.



(a) (b)

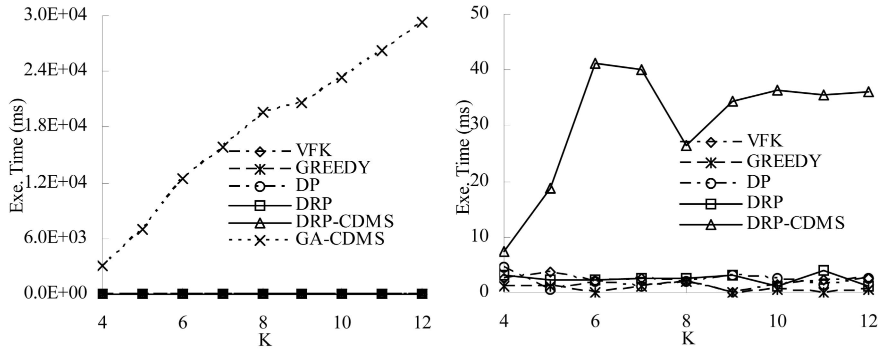
Fig. 13. The expected value of waiting time as N varied. (a) Small-range configuration. (b) Wide-range configuration.

without adopting CDMS when $K = 4$ and $K = 8$. That is, the improvement of DRP-CDMS 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 broadcast items, the observed performances is depicted in Fig. 13. When the number of broadcast items increases, the average waiting time for each approach increases. This phenomenon also agrees with our intuition since each channel should disseminate more items. Here, the proposed DRP and DRP-CDMS still result in better qualities than VF^K, GREEDY, and DP consistently. By observing the performances of DRP-CDMS as N varied in Fig. 13a, we can see that the results are still very close to the quality of GA-CDMS. The qualities of DRP and DRP-CDMS are not affected as the value of N increases. Therefore, it is shown that mechanism CDMS is scalable so that the quality can be maintained when larger numbers of data items are broadcast.

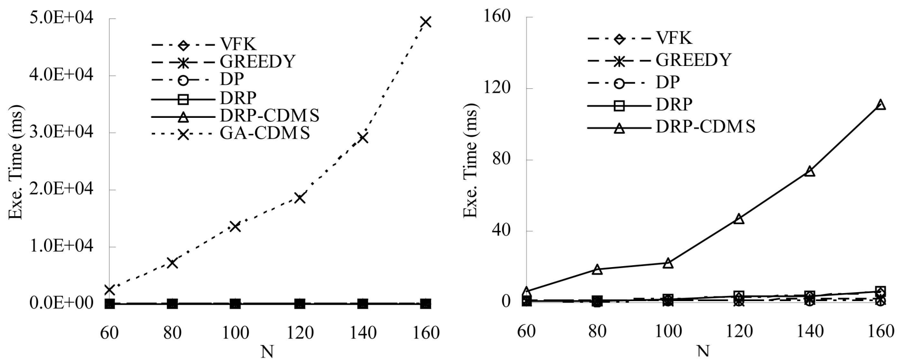
6.4 Efficiency Analysis

In the final section, we discuss the efficiency among all 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 efficiency. We use millisecond as the unit of execution time. Since the parameters θ and Φ do not affect the complexity, in this experiment, we only consider the parameters K and N. Fig. 14 shows the execution time of each approach as the number of channels K varies, while Fig. 15 depicts the execution times as N increases. Compared to GA-CDMS, algorithm DRP-CDMS spends much less time generating broadcast programs. The execution time of the GA-CDMS increases as K or N increases. Note that the execution time of GA-CDMS is more sensitive to N than to K. The reason is given as follows: The GA-CDMS is implemented based on the *hybrid* genetic algorithm. The increase of N will increase the length of each chromosome, while the increase of K only changes the variety of the gene value in a chromosome. Moreover, the execution time of DRP-CDMS is about 2.5 to 35.5 times higher than that of DRP as K is varied from four to nine, whereas the execution time of DRP-CDMS is about 12.5 to 20 times higher than that of DRP as N is varied from 60 to 160. Without CDMS, the execution cost of DRP is similar to VF^K. Obviously, DRP-CDMS provides a broadcast program with a higher quality at higher execution cost. By observing the above two figures, we find that although the GA-CDMS can achieve the best solutions, it is computationally prohibitive. Compared to GA-CDMS, the proposed algorithms are still very efficient whether the CDMS optimization procedure is employed or not. Therefore, the proposed DRP and DRP-CDMS are very suitable for generating broadcast programs practically. As mentioned earlier, since DRP-CDMS achieves the local optimal solution, which is very close to the global optimal, to generate the broadcast program with high quality, we suggest that the DRP-CDMS approach should be employed.



(a) (b)

Fig. 14. The number of channels versus the execution time (small-range configuration). (a) With GA-CDMS. (b) Without GA-CDMS.



(a) (b)

Fig. 15. The number of broadcast items versus the execution time (small-range configuration). (a) With GA-CDMS. (b) Without GA-CDMS.

7 Conclusion

In this paper, we focus on generating broadcast programs in a *heterogeneous* data broadcasting environment. The two-phase channel allocation approach is proposed in the paper. First, we propose algorithm DRP to perform the *rough* allocation. After that, we also use a mechanism called CDMS to refine the result of DRP to the local optimum. Moreover, a *hybrid genetic algorithm* GA-CDMS is also proposed for comparison purposes. In order to verify the performance, several experiments are conducted. In these experiments, we consider the important issues such as accuracy, scalability, diversity, and complexity. From the experimental results, we show that the proposed two-phase channel allocation is very practical in performing an effective channel allocation efficiently in a heterogeneous broadcasting environment.

Acknowledgments

This work was supported in part by the National Science Council of Taiwan, R.O.C., under Contracts NSC97-2221-E-002-172-MY3.

References

[1] D. Barbará, "Mobile Computing and Database—A Survey," *IEEE Trans. Knowledge and Data Eng.*, vol. 11, no. 1, pp. 108-117, Jan./Feb. 1999.

[2] J. Xu, X. Tang, and D. Lee, "Performance Analysis of Location-Dependent Cache Invalidation Schemes for Mobile Environments," *IEEE Trans. Knowledge and Data Eng.*, vol. 15, no. 2, pp. 474-488, Mar./Apr. 2003.

[3] M. Satyanarayanan, "Pervasive Computing: Vision and Challenges," *IEEE Personal Comm.*, vol. 8, no. 4, pp. 10-17, Aug. 2001.

[4] C.H. Hsu, G. Lee, and A.L.P. Chen, "An Efficient Algorithm for Near Optimal Data Allocation on Multiple Broadcast Channels," *Distributed and Parallel Databases*, vol. 18, no. 3, 2005.

[5] J.-L. Huang and M.-S. Chen, "Broadcasting Dependent Data for Ordered Queries without Replication in a Multi-Channel Mobile Environment," *Proc. 19th IEEE Int'l Conf. Data Eng. (ICDE '03)*, Mar. 2003.

[6] T. Imielinski, S. Viswanathan, and B.R. Badrinath, "Data on Air: Organization and Access," *IEEE Trans. Knowledge and Data Eng.*, vol. 9, no. 3, pp. 353-372, May/June 1997.

[7] J.-L. Huang and M.-S. Chenn, "Dependent Data Broadcasting for Unordered Queries in a Multiple Channel Mobile Environment," *IEEE Trans. Knowledge and Data Eng.*, vol. 16, no. 6, June 2004.

[8] W.-C. Peng and M.-S. Chen, "Efficient Channel Allocation Tree Generation for Data Broadcasting in a Mobile Computing Environment," *Wireless Networks*, vol. 9, no. 2, pp. 117-129, 2003.

[9] W.-G. Yee, S.B. Navathe, E. Omiecinski, and C. Jermaine, "Efficient Data Allocation over Multiple Channels at Broadcast Servers," *IEEE Trans. Computers*, vol. 51, no. 10, pp. 1231-1236, Oct. 2002.

[10] B. Zheng, X. Wu, X. Jin, and D.L. Lee, "TOSA: A Near-Optimal Scheduling Algorithm for Multi-Channel Data Broadcast," *Proc. Sixth Int'l Conf. Mobile Data Management (MDM '05)*, 2005.

[11] J.-L. Huang and M.-S. Chen, "Broadcast Program Generation for Unordered Queries with Data Replication," *Proc. ACM Symp. Applied Computing (SAC '03)*, Mar. 2003.

[12] S. Acharya and S. Muthukrishuan, "Scheduling On-Demand Broadcasts: New Metrics and Algorithm," *Proc. ACM MobiCom*, 1998.

[13] C.C. Aggarwal, J.L. Wolf, and P.S. Yu, "A Permutation-Based Pyramid Broadcasting Scheme for Video-on-Demand Systems," *Proc. Int'l Conf. Multimedia Computing and Systems (ICMCS '96),* 1996.
[14] L.-S. Juhn and L.-M. Tseng, "Fast Data Broadcasting and Receiving Scheme for Popular Video Service," *IEEE Trans. Broadcasting,* vol. 44, no. 1, Mar. 1998.
[15] S. Viswanathan and T. Imielinski, "Pyramid Broadcasting for Video-on-Demand Service," *Proc. SPIE Multimedia Computing and Networking Conf. (MMCN '95),* 1995.
[16] D.E. Goldberg, *Genetic Algorithm in Search, Optimization and Machine Learning.* Addison-Wesley, 1989.
[17] J.H. Holland, *Adaptation in Natural and Artificial Systems.* Univ. of Michigan Press, 1975.
[18] S. Acharya, R. Alonso, M.J. Franklin, and S.B. Zdonik, "Broadcast Disks: Data Management for Asymmetric Communications Environments," *Proc. ACM SIGMOD,* pp. 199-210, May 1995.
[19] J.W. Wong, "Broadcast Delivery," *Proc. IEEE,* vol. 76, no. 12, pp. 1566-1577, Dec. 1988.
[20] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms.* McGraw-Hill.
[21] F. Martinez, J. Gonzalez, and I. Stojmenovic, "A Parallel Hill Climbing Algorithm for Pushing Dependent Data in Clients-Providers-Servers Systems," *Proc. Seventh IEEE Symp. Computers and Comm. (ISCC '02),* 2002.
[22] Y. Chung and M. Kim, "Efficient Data Placement for Wireless Broadcast," *Distributed and Parallel Database,* vol. 9, no. 2, Mar. 2001.
[23] C.J. Su and L. Tassiulas, "Broadcast Scheduling for Information Distribution," *Proc. IEEE INFOCOM,* 1997.
[24] N.H. Vaidya and S. Hameed, "Scheduling Data Broadcast in Asymmetric Communication Environments," *ACM/Baltzer Wireless Networks,* vol. 5, no. 3, pp. 171-182, 1999.
[25] S. Hameed and N.H. Vaidya, "Efficient Algorithms for Scheduling Data Broadcast," *ACM/Baltzer Wireless Networks,* vol. 5, no. 3, pp. 183-193, 1999.
[26] D. Aksoy and M.J. Franklin, "Scheduling for Large Scale On-Demand Data Broadcasting," *Proc. IEEE INFOCOM,* 1998.
[27] J.-L. Huang, M.-S. Chen, and H.-P. Hung, "A QoS-Aware Transcoding Proxy Using On-Demand Data Broadcasting," *Proc. IEEE INFOCOM,* Mar. 2004.
[28] Y. Wu and G. Cao, "Stretch-Optimal Scheduling for On-Demand Data Broadcasts," *Proc. 10th Int'l Conf. Computer Comm. and Networks (ICCCN '97),* 1997.
[29] D. Aksoy and M.S. Leung, "Pull versus Push: A Quantitative Comparison for Data Broadcast," *Proc. IEEE Global Telecomm. Conf. (GLOBECOM '04),* 2004.
[30] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman, 1979.
[31] M.-S. Chen, P.S. Yu, and K.-L. Wu, "Optimizing Index Allocation for Sequential Data Broadcasting in Wireless Mobile Computing," *IEEE Trans. Knowledge and Data Eng.,* vol. 15, no. 1, Jan./Feb. 2003.
[32] S.-C. Lo and A.L. Chen, "Optimal Index and Data Allocation in Multiple Broadcast Channels," *Proc. 16th IEEE Int'l Conf. Data Eng. (ICDE '00),* pp. 293-302, 2000.
[33] C.-L. Hu and M.-S. Chen, "Adaptive Multi-Channel Data Dissemination: Support of Dynamic Traffic Awareness and Push-Pull Time Balance," *IEEE Trans. Vehicular Technology,* vol. 54, no. 2, 2005.
[34] T. Imielinski and B.R. Badrinath, "Wireless Mobile Computing: Challenges in Data Management," *Comm. ACM,* vol. 37, no. 10, 1994.
[35] H.P. Hung and M.S. Chen, "Efficient Data Broadcasting by Progressively Merging and Splitting," *Proc. IEEE Global Telecomm. Conf. (GLOBECOM '06),* 2006.
[36] W.-C. Peng, J.-L. Huang, and M.S. Chen, "Dynamic Leveling: Adaptive Data Broadcasting in a Mobile Computing Environment," *Mobile Networks and Applications,* vol. 8, no. 4, pp. 355-364, 2003.
[37] J.M.R. Dondo and R.G. Valle, "An Optimal Power System Model Order Reduction Technique," *Int'l J. Electrical Power and Energy Systems,* vol. 26, no. 7, pp. 493-500, Sept. 2004.
[38] Y.B. Park, "A Hybrid Genetic Algorithm for the Vehicle Scheduling Problem with Due Times and Time Deadlines," *Int'l J. Production Economics,* vol. 73, no. 2, pp. 175-188, 2001.
[39] J. Andre, P. Siarry, and T. Dognon, "An Improvement of the Standard Genetic Algorithm Fighting Premature Convergence in Continuous Optimization," *Advances in Eng. Software,* vol. 32, no. 12, pp. 49-60, Jan. 2001.
[40] A.G. Williamson, "Refining a Neural Network Credit Application Vetting System with a Genetic Algorithm," *J. Microcomputer Applications,* vol. 18, no. 3, pp. 261-277, July 1995.
[41] D.B. Fogel, "An Introduction to Simulated Evolutionary Optimization," *IEEE Trans. Neural Networks,* vol. 5, no. 1, pp. 3-14, Jan. 1994.
[42] G. Syswerda, "Uniform Crossover in Genetic Algorithms," *Proc. Third Int'l Conf. Genetic Algorithms (ICGA '89),* 1989.
[43] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs,* second extended ed. Springer, 1994.
[44] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programming,* second ed. Springer, 1994.
[45] G.K. Zipf, *Human Behaviour and the Principle of Least Effort.* Addison-Wesley, 1949.
[46] V. Padmanabhan and L. Qiu, "The Content and Access Dynamics of a Busy Web Site: Findings and Implications," *Proc. ACM SIGCOMM,* 2000.

Hsiao-Ping Tsai received the BS and MS degrees in computer science and information engineering from National Chiao Tung University in 1996 and 1998, respectively, and the PhD degree in electrical engineering from National Taiwan University in January 2009. She joined the Institute of Information Science at Academia Sinica as a postdoctoral fellow in February 2009. Her research interests include mobile computing, data mining, sensor data management, and object tracking sensor networks. She is a student member of the IEEE.

Hao-Ping Hung received the BS and PhD degrees from the National Taiwan University, Taipei, in 2001 and 2007, respectively. Now, he serves as a senior engineer at CyberLink Corporation. His research interests include mobile computing, resource allocation in the wireless environment, multimedia networking, and data streams.

Ming-Syan Chen received the BS degree in electrical engineering from the National Taiwan University, Taipei, and the MS and PhD degrees in computer, information, and control engineering from the University of Michigan, Ann Arbor, in 1985 and 1988, respectively. He is now the director and a distinguished research fellow of the Research Center for Information Technology Innovation at Academia Sinica, Taiwan, and also a distinguished professor in the Department of Electrical Engineering, National Taiwan University. He was a research staff member at IBM T.J. Watson Research Center, Yorktown Heights, New York, from 1988 to 1996, the director of GICE from 2003 to 2006, and the president/CEO of the Institute for Information Industry (III), which is one of the largest organizations for information technology in Taiwan, from 2007 to 2008. His research interests include databases, data mining, mobile computing systems, and multimedia networking. He has published more than 270 papers in his research areas. In addition to serving as a program chair/vice chair and a keynote/tutorial speaker in many international conferences, he was an associate editor of the *IEEE Transactions on Knowledge and Data Engineering* and the *Journal of Information Science and Engineering*, is currently on the editorial board of *The VLDB Journal*, *Knowledge and Information Systems*, and the *International Journal of Electrical Engineering*, and was a distinguished visitor of the IEEE Computer Society for Asia-Pacific from 1998 to 2000 and from 2005 to 2007. He holds or has applied for 18 US patents and seven ROC patents in his research areas. He is a recipient of the National Science Council (NSC) Distinguished Research Award, the Pan Wen Yuan Distinguished Research Award, the Teco Award, the Honorary Medal of Information, and the K.-T. Li Research Breakthrough Award for his research work, and also the Outstanding Innovation Award from IBM Corporate for his contribution to a major database product. He also received numerous awards for his research, teaching, inventions, and patent applications. He is a fellow of the ACM, the IEEE, and the IEEE Computer Society.