Theoretical Computer Science 497 (2013) 123–130

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Algebraic data retrieval algorithms for multi-channel wireless data broadcast

Xiaofeng Gao [a,*], Zaixin Lu [b], Weili Wu [b], Bin Fu [c]

[a] *Department of Computer Science & Engineering, Shanghai Jiao Tong University, PR China*

[b] *Department of Computer Science, University of Texas at Dallas, USA*

[c] *Department of Computer Science, University of Texas–Pan American, USA*

ARTICLE INFO

Keywords:
Data broadcast
Data retrieval
Algebraic algorithm
Mobile computing

ABSTRACT

Wireless data broadcast is an important data dissemination method for distributing public information to mobile users. Due to the exponentially increasing number of mobile network users, it is necessary to develop efficient data retrieval protocols for end users to download data items effectively. In this paper, we concentrate on investigating scheduling algorithms for retrieving a set of data items from a multichannel wireless data broadcast system. As we know, the most important issues in mobile computing are energy efficiency and query response efficiency. However, in data broadcast the objectives of reducing access latency and energy cost can be contradictive to each other. Consequently, we define a new problem named *Minimum Constraint Data Retrieval Problem* (MCDR). We prove that MCDR is NP-hard, and then show a fixed parameter tractable algorithm which can balance two factors together. It has computational time $O(2^k(hnt)O^{(1)})$, where n is the number of channels, k is the number of required data items, t is the maximal time slot, and h is the maximal number of channel switches.

1. Introduction

Nowadays, mobile computing technology is developing rapidly and the number of wireless/mobile clients is growing explosively. In the environment, *wireless data broadcast* becomes an efficient data dissemination method for service providers to spread public information to a large number of mobile subscribers. It is a fast, effective, and low-power access to overcome technological limitations of wireless communication. In a typical data broadcast system, data items are sent as data packets to a *base station* (BS), and then the BS will broadcast the packets through multiple RF channels repeatedly. Clients will access channels, get the location of their required data item using indices, and then access the target channel and download the data.

Since the majority of mobile devices have limited battery power and constrained lifetime, the most important issues in wireless data broadcast systems are energy-efficiency and query response time for clients. Correspondingly, two criteria are defined to evaluate the performance of a data broadcast system. They are

1. **access latency**: the time elapsed from when a client starts the query process until it successfully downloads all the required data items;
2. **tuning time**: the time interval a client stays active during its query and downloading process.

* Corresponding author. Tel.: +86 13911712720.
E-mail address: gao-xf@cs.sjtu.edu.cn (X. Gao).

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2011.12.070

Here "active" means the status when a mobile device processes operations. According to the architectural enhancements, each mobile device has an *active mode* and a *doze mode*. It processes operations only in the active mode, while it "sleeps" in the doze mode to save energy. The energy consumed in the active mode is usually 20–30 times higher than that in the doze mode.

Intuitively, a critical problem to improve the performance of a wireless data broadcast system is to reduce the access latency and tuning time. Index technology can help in reducing tuning time significantly. Using a predefined index structure (e.g., Distributed Index [8], Huffman Tree [11], Hash Table [18], Exponential Index [19], Signature Tree [21]), a client can get an estimated waiting offset and channel location for its required datum. Then it tunes off during the offset, and tunes in to the corresponding channel right before the target datum appears. Formally, in an indexed multi-channel wireless data broadcast system, a client needs three steps to retrieve a datum:

1. **Initial probe**: tune in a broadcast channel (usually an arbitrary channel or an index channel) and find the closest index packet;
2. **Index search**: search through indices, find the time offset and channel location of the requested datum;
3. **Data retrieval**: tune in target channel and download the datum when it appears.

Different index techniques may bring different searching processes, but no matter what index technique is used, by the end of the searching process, clients should get the information of time offset and residence channel of requested data. Since in this paper we focus on the data retrieval process, we omit the discussion of the searching process and index constructions, and assume we know the locations of the required data.

A client may require more than one datum. In this case, for a required data set, the client needs to get the location of every datum, order them as a permutation, and download them one by one sequentially. If the order of data items is not appropriate, the client may spend unnecessarily extra time for downloading. Thus, a time efficient schedule for data retrieval is very important to reduce access latency and improve system efficiency. Moreover, during the data retrieval process, the tuning time is always the time needed for downloading the required data set. However, if two data items are located on different channels, a client should "hop" (also called switch) from one channel to another. Say, disconnect from one communication channel, and then construct a connection to another channel. Hop among channels costs additional high energy consumption [7]. Therefore, the total hop count during the data retrieval process also has a notable impact on the energy efficiency of a data retrieval protocol.

Combining the two impacts together, we attempt to determine an optimal permutation pattern for retrieving a required data set on a multi-channel data broadcast system, such that a client can spend minimum time and minimum energy to download all required data items. However, reducing access latency may be contradictary to reducing energy (hop count) at the same time. In Section 3 we provide an example to illustrate this contradiction in detail. To balance the two objectives together, we provide two parameters, n and h, as constraints for access latency and hop count, then define the *Minimum Constraint Data Retrieval Problem* (MCDR). In MCDR, we either bound access latency n and minimize the hop counts, or bound h, and minimize access latency for the data retrieval process.

We prove that MCDR is NP-hard by a reduction from the vertex cover problem. Then we present an algebraic algorithm for the data retrieval problem in multi-channel environments. It can detect whether a given problem has a schedule to download all the requested data before time t and with at most h channel switchings in $O(2^k(nht)^{O(1)})$ time, where n is the number of channels and k is the number of required data items. We also provide a fixed parameter tractable (FPT) algorithm with computational time $O(2^l(nht)^{O(1)})$. It can determine whether there is a scheduling to download l data items from data set D in at most n time slots and at most h channel switches. The service provider can adjust n and h freely to fit their own requirements.

The rest of this paper is organized as follows. In Section 2 we study previous literature related to the MCDR problem. In Section 3 we discuss the system model, problem requirement, and the formal definition of MCDR, and prove the NP-hardness of this problem. In Section 4 we describe our algorithm construction with a correctness proof and theoretical analysis, while in Section 5 we give a conclusion and future work on our research.

2. Previous works

Scheduling is an important issue in the area of wireless data broadcast. Acharya et al. first proposed the scheduling problem for data broadcast [1], and Prabhakara et al. suggested the multi-channel model for data broadcast to improve the data delivery performance [14]. Since then, many works have been done for scheduling data on multiple channels to reduce the expected access time [20,22,2]. Besides, some researches began to study how to allocate dependent data on broadcast channels (see, e.g., [10,19,21,5,6]). With respect to index, many methods have been proposed to improve the search efficiency in data broadcast systems (see, e.g., [8,16,18,19,21]). Furthermore, Jung et al. proposed a tree-structured index algorithm that allocates indices and data on different channels [11]. Lo and Chen designed a parameterized schema for allocating indices and data optimally on multiple channels such that the average expected access latency is minimized [12].

In terms of data retrieval scheduling, Hurson et al. proposed two heuristic algorithms for downloading multiple data items from multiple channels [7]. Shi et al. investigated how to schedule multiple processes to download a set of data items [15]. Both of them investigate the data retrieval problem by assuming that the data are allocated on multiple channels without

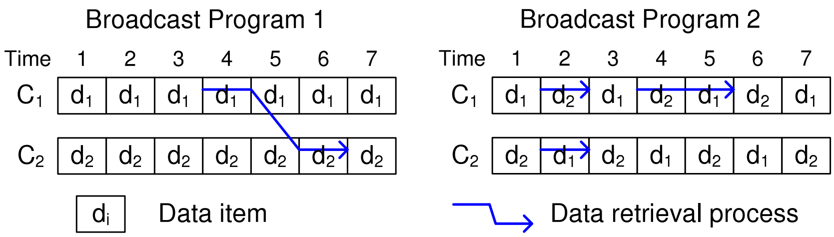

Fig. 1. Two types of broadcasting system.

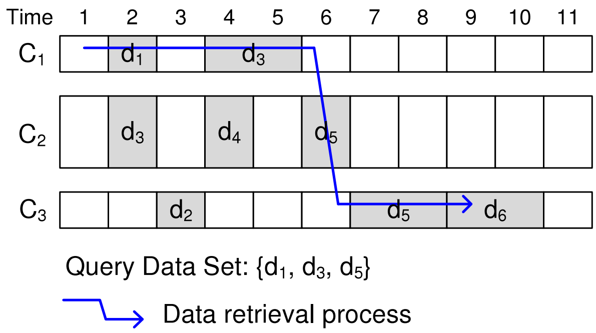

Fig. 2. Switch Constraint.

replication. However, as shown in the prior studies [1,4,3,13], employing data replication in data broadcast programs will reduce the expected access time.

Fig. 1 shows why disseminating replicative data by multiple channels can reduce both access time and energy consumption. The first program allocates data without replication. d_1 and d_2 are separately scheduled on channels c_1 and c_2. we can download d_1 or d_2 in one time slot, but we need at least 3 time slots and 1 channel switching to download both d_1 and d_2 in such a system. If we allocate data on channels in the way of program 2, we can still retrieve each datum in one time slot and we can retrieve both of them in 2 time slots without channel switching.

In this paper, we develop an algebraic algorithm to efficiently retrieve multiple data from multiple channels, in which data items can have different lengths and can be broadcasted multiple times in one broadcast cycle on multiple channels. Each channel may have different bandwidth as well.

3. Problem formulation

In this section, we discuss our system model in detail. Firstly we assume that a client wants to download a group of k data items $D = \{d_1, d_2, \ldots, d_k\}$. Each data item may have different size. Those data items are broadcasted on n different channels repeatedly together with many other data items. The channel set is $C = \{c_1, c_2, \ldots, c_n\}$. Each channel may have different bandwidth and broadcast cycle length. Let the time to download the smallest transmission packet be a unit time, and the length of d_i can be represented as l_i (also referred to as downloading time).

Assume there is s special channel used for indices, and clients firstly access this channel and get the locations (channel id and time offsets) of the required data set. Next, they will access data channels to download the corresponding data items one after another. Assume the time when the client gets all the location information of the data set is the starting point ($t = 0$), then the target becomes how to download k known data from n channels efficiently.

3.1. Switch constraint

Note that if a client is downloading a datum from channel c_i at time t_0, then it cannot switch to channel c_j, where $j \neq i$, to download another datum at time $t_0 + 1$. The reason is that switching the channels takes time, and if a client wants to download data from another channel, at least one time unit is needed for channel switching. Fig. 2 gives a typical process of data retrieval in multi-channel broadcast environments. The query data set is $\{d_1, d_3, d_5\}$, and a user can download data objects d_1 and d_3 from channel c_1, and then switch to channel c_3 at time $t = 6$ to download data object d_5 at time $t = 7$. However, after time $t = 5$, the user cannot switch from channel c_1 to c_2 to download datum d_5 at time $t = 6$. From Fig. 2, we also can get that the bandwidths of different channels are not necessarily the same. Actually, the bandwidth of channel c_2 is twice that of c_1 or c_3, thus d_3 or d_5, which take two time slots on c_1 or c_3, can be broadcasted in one time unit by c_2.

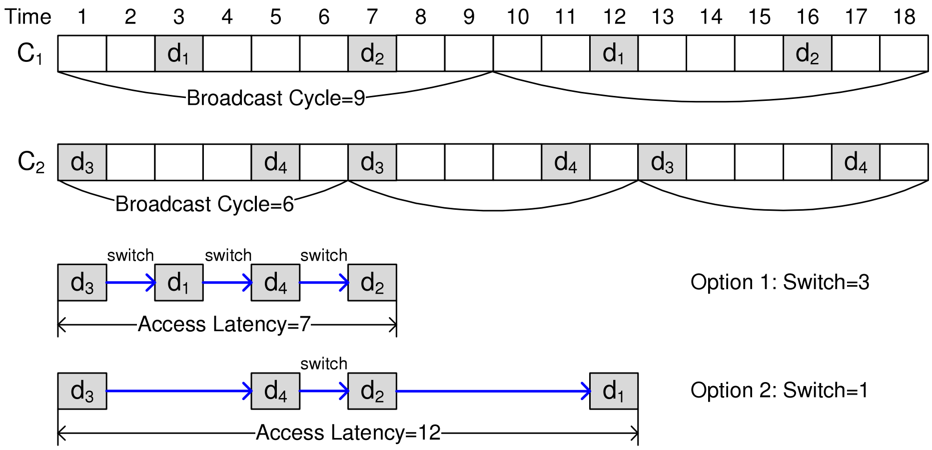

Fig. 3. Example of possible objective contradiction.

3.2. Evaluation criteria

As we mentioned in Section 1, the *access latency*, *tuning time*, and *energy consumption* are the most important criteria to evaluate the efficiency of a downloading process; we want to design a data retrieval protocol with minimum access latency, minimum tuning time, and minimum energy consumption. Let us then further discuss these three impacts.

- **Tuning time**: Since we know the location of each data item, the tuning time in this problem is always the same (the length of all data, $\sum_{i=1}^{k} l_i$). So we can ignore this factor.
- **Energy consumption**: The energy consumption of the data retrieval problem comes from two parts: the energy of data downloading and the energy of channel switching. Since the first part always remains the same, we focus on how to reduce the number of channel switches (hop counts).
- **Access latency**: In this problem we define the access latency from $t = 0$ to the time when all the data have been downloaded successfully.

Therefore, we hope to find an algorithm to download k data items with minimum time duration and minimum switch numbers. However, these two factors are in conflict with one another. The following figure is an example of the conflict.

In the broadcast program shown in Fig. 3, suppose the grey data items $\{1, 2, 3, 4\}$ are the request. The starting point of the retrieval process is at $t = 1$. If we want to minimize the access latency, the request should be retrieved in the order "$3 \to 1 \to 4 \to 2$" which takes only 7 time units but needs 3 switches. However, if we want to minimize the switches, the best retrieval order should be "$3 \to 4 \to 2 \to 1$" which needs only 1 switch but takes 12 time units. This example shows that access latency and number of switches cannot be minimized at the same time.

3.3. Minimum constraint data retrieval problem

As a consequence, we want to fix one factor and minimize another objective, and thus define the *Minimum Constraint Data Retrieval Problem*. The formal definition is shown as follows.

Definition 1 (*Minimum Constraint Data Retrieval Problem*)**.** Given k target data $D = \{d_1, \ldots, d_k\}$ located on n different channels $C = \{c_1, \ldots, c_n\}$. Each d_i has length l_i, and is located at some position on channel c_j. Each channel has different bandwidth and different broadcast cycle length. If we fix a switch (latency) parameter h (t), then the *Minimum Constraint Data Retrieval Problem* is to find a minimum access latency (switch-number) data retrieval schedule, with at most h switches (t access latency).

Let a tuple $s = \{i_s, j_s, t_s, t'_s\}$ denote that the data d_{i_s} can be downloaded from channel c_{j_s} during the time span $[t_s, t'_s]$, then it is clear that a valid data retrieval schedule is a sequence of k intervals $s_1, s_2, \ldots, s_k$, each tuple corresponds to a distinct data item in D and there are no conflicts between any two of the k tuples. To analyze the NP-hardness, we then define the decision problem of MCDR.

Definition 2 (*Decision MCDR*)**.** Given a data set D, a channel set C, a time threshold t and a switching threshold h, find a valid data retrieval schedule to download all the data in D from C before time t with at most h switchings.

Theorem 1. *The MCDR problem is NP-hard.*

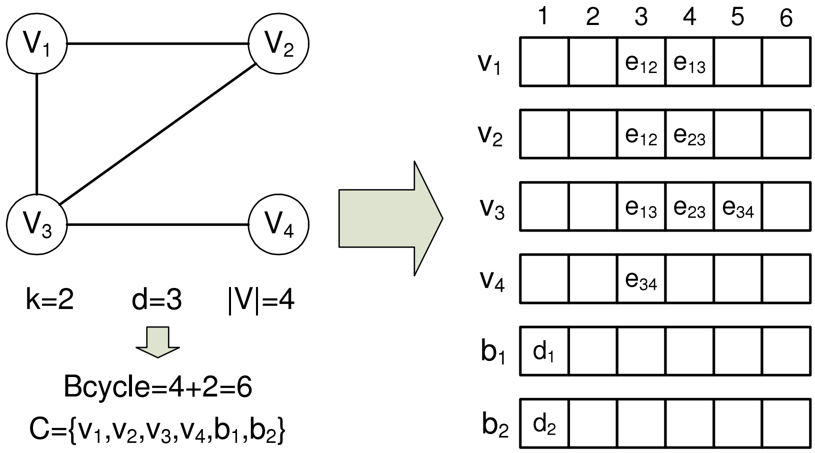

Fig. 4. Example of VC reduction.

Proof. We use $VC \leq_p MCDR$ to prove this theorem. Here VC is the decision problem of Vertex Cover: say, given a graph $G = (V, E)$, we want to find a minimum size vertex subset $VC \subseteq V$ such that for any edge $(v_i, v_j) \in E$, either $v_i \in VC$ or $v_j \in VC$. An instance of Vector Cover is: given a graph $G = (V, E)$ and integer k, does it have a vertex cover VC with size k. Then we construct an instance of MCDR from G and k as follows.

- For each vertex $v_i \in V$, define a channel v_i. Define another k channels $b_1, \ldots, b_k$. Then the channel set is $C = \{v_1, \ldots, v_{|V|}, b_1, \ldots, b_k\}$. In total $|V| + k$ channels. Let δ be the maximum vertex degree in G, then each channel has broadcast cycle length $\delta + 3$.
- For each edge $(v_i, v_j) \in E$, define a unit length data item e_{ij} in data set D_e, and append it on channels c_i and c_j (the order can be arbitrary, and starting from the third time unit).
- For each channel b_i, define a unit length data item d_i in data set D_d, and allocate it on the first time unit of channel b_i.
- The data set $D = D_e \cup D_b$.

Fig. 4 is an example to show how to construct the broadcast system. In this figure, $\delta = 3$, $k = 2$, $|V| = 4$, thus the channel set should be $\{v_1, v_2, v_3, v_4, b_1, b_2\}$, each has broadcast length $\delta + 3 = 6$. Each e_{ij} represents an edge (v_i, v_j), and it is clear that if we download all data items from channel v_i, then it means we cover the edges connecting node v_i.

Next, we prove that G has a vertex cover with size k if and only if there is a valid data retrieval schedule S such that $t = k(\delta + 3)$, and $h = 2k - 1$.

$\Longrightarrow$: If G has a vertex cover VC with size k, then we can select the corresponding k channels in $\{v_i | v_i \in VC\}$ to receive all the data in k cycles. At the beginning of the ith cycle (iteration), the client will visit b_i at $t = 1$, and hop to some $v_i \in VC$ channel, stay on this channel till the last time unit of the broadcast cycle, and then hop to b_{i+1}. There are k b_i's, so each iteration client will download one of them. VC is a vertex cover, so following all $v_i \in VC$ we must download every e_{ij}. The length of each broadcast cycle is $\delta + 3$, so the total access latency is $k(\delta + 3)$. In each broadcast cycle the client will switch twice (except the last cycle), so $h = 2k - 1$.

$\Longleftarrow$: Assume MCDR has a valid schedule S with $t = k(\delta + 3)$ and $h = 2k - 1$. Let us consider D_b first. There are k b_i's located at the first time unit on k different channels. This means we have to switch at least $k - 1$ hops to download D_b, then we only have another k hops for D_e, which means we can visit at most k channels in $\{v_i\}$. At the beginning of each broadcast cycle, we always stay at some channel b_i to download d_i, then we switch to some v_i, and at the end of this cycle we have to switch to channel b_{i+1} for d_{i+1}. This means we cannot switch to two vertex channels within one broadcast cycle, otherwise we cannot download $D = D_e \cup D_b$ in k iterations. Since S is valid, we visit k vertex channels and download all D_e data items, this means these k vertices form a vertex cover with size k.

This reduction can be done in polynomial time, and we can conclude that MCDR is NP-hard. □

4. Algorithm design

To solve the above decision problem, we developed a randomized algebraic algorithm. It can detect whether a given problem has a schedule to download all the requested data before time t and with at most h channel switchings in $O(2^k(nht)^{O(1)})$ time, where n is the number of channels and k is the number of required data items. We also provide a fixed parameter tractable (FPT) algorithm with computational time $O(2^l(nht)^{O(1)})$. It can determine whether there is a scheduling to download l data items from D in at most n time slots and at most h channel switches. The service provider can adjust n and h freely to fit their own requirements. We firstly give some preliminaries and then present our algorithms in detail.

4.1. Preliminaries

Here we introduce some notions about group algebra which are not often used in algorithm design.

Definition 3. Assume that $x_1, \ldots, x_k$ are variables in group algebra. Then:

1. A *monomial* has format $x_1^{a_1} x_2^{a_2} \cdots x_k^{a_k}$.
2. A *multilinear monomial* is a monomial such that each variable has degree exactly one. For example, $x_3x_5x_6$ is a multilinear monomial, but $x_3x_5^2x_6^3$ is not.
3. For a *polynomial* $p(x_1, \ldots, x_k)$, its *sum of product expansion* is $\sum_j p_j(x_1, \ldots, x_k)$, where each p_j is a monomial, which has a format $c_j x_1^{a_{j_1}} \cdots x_k^{a_{j_k}}$ with c_j with respect to its coefficient.
4. $G_2 = (\{0, 1\}, +, \cdot)$ is a field with two elements $\{0, 1\}$ and two operations $+$ and $\cdot$. The addition operation is under the modular of 2 (mod 2).
5. Z_2^k is the group of binary k-vectors. Let w_0 denote the all-zeros vector, which is the identity of Z_2^k, then for every $v \in Z_2^k$, $v^2 = w_0$, $v \cdot w_0 = v$.

The operations between elements in the group algebra are standard.

4.2. Algorithm

The basic idea of our algebraic algorithm is that for each data item $d_i \in D$, where D is the query data set, we create a variable x_i to represent it. Therefore, given $D = \{d_1, d_2, \ldots, d_k\}$, we construct a variable set $X = \{x_1, x_2, \ldots, x_k\}$. We then design a circuit $H_{t,h,n}$ such that a schedule without conflict will be generated by a multilinear monomial in the sum of product expansion of the circuit. The existence of schedules to download all the data items in D from the multiple channels of C is converted into the existence of multilinear monomials of $H_{t,h,n}$. Replacement of each variable by a specified binary vector can remove all of the non-multilinear monomials by converting them to zero. Thus, the data retrieval problem is transformed into testing whether a multivariate polynomial is zero. It is well known that randomized algorithms can be used to check whether a circuit is identical to zero in polynomial time.

Lemma 1. *There is a polynomial time algorithm such that given a channel c_i, a time interval $[t_1, t_2]$, and an integer m, it constructs a circuit of polynomial $P_{i,t_1,t_2,m}$ such that for any subset $D' = \{d_{i_1}, \ldots, d_{i_m}\} \subseteq D$ which has a size of m and is downloadable in the time interval $[t_1, t_2]$ from channel c_i, the product expansion of $P_{i,t_1,t_2,m}$ contains a multilinear monomial $x_{i_1}x_{i_2} \cdots x_{i_m}$.*

Proof. We can use a recursive way to compute the circuit $P_{i,t_1,t_2,m}$ in polynomial time.

1. $P_{i,t_1,t_2,0} = 0$.
2. $P_{i,t_1,t_2,1} = \sum_j x_j$, $x_j \subseteq X$ and the corresponding data item d_j is entirely in the time interval $[t_1, t_2]$ of channel c_i.
3. $P_{i,t_1,t_2,l+1} = \sum_j x_j \cdot P_{i,t_1,t_2',l} + P_{i,t_1,t_2',l+1}$, d_j starts at time $t_2' + 1$ and ends before time t_2 on channel c_i.

When computing $P_{i,t_1,t_2,l+1}$, x_j multiplies $P_{i,t_1,t_2',l}$ is based on the case that d_j is downloadable from time $t_2' + 1$ to t_2 in the final phase, and the other l data items are downloadable before time t_2'. The term $P_{i,t_1,t_2',l+1}$ is the case that $l + 1$ items are downloaded before time t_2'. Note that the parameter m in $P_{i,t1,t2,m}$ controls the total number of data to be downloaded. □

Definition 4. A subset of data items $D' = \{d_{i_1}, \ldots, d_{i_m}\} \subseteq D$ is (i, t, h)-downloadable if we can download all data items in D' before time t, the total number of channel switches is at most h, and the last downloaded item is from channel c_i.

Lemma 2. *Given two integers t and h, there is a polynomial time algorithm to construct a circuit of polynomial $F_{i,t,h,m}$ such that for any (i, t, h)-downloadable subset $D' = \{d_{i_1}, \ldots, d_{i_m}\} \subseteq D$, the product expansion of $F_{i,t,h,m}$ contains a multilinear monomial $(x_{i_1}, \ldots, x_{i_m})Y$, where Y is a multilinear monomial and does not include any variable in X.*

Proof. We still use a recursive way to construct the circuit. Some additional variables are used as needed. Without loss of generality, we assume the data retrieval process starts at time 0.

1. $F_{i,t,0,0} = 0$.
2. $F_{i,t,0,1} = P_{i,1,t,1} \cdot y_{i,t,0,1}$.
3. $F_{i,t,h'+1,m'+1} = y_{i,t,h'+1,m'+1,0}(\sum_{t'<t} F_{i,t',h'+1,m'} \cdot P_{i,t'+1,t,1}) + y_{i,t,h'+1,m'+1,1}(\sum_{j \neq i} \sum_{t'<t} F_{i,t'-1,h',m'} \cdot P_{i,t'+1,t,1})$. □

The computing of $F_{i,t,h'+1,m'+1}$ is based on two cases, and we use two variables, $y_{i,t,h'+1,m'+1,0}$ and $y_{i,t,h'+1,m'+1,1}$, to mark them respectively. We now present an algorithm that involves one layer randomization to determine whether there is a schedule to download all the data items in D before time t and with at most h channel switchings.

Theorem 2. *There is an $O(2^k(hnt)^{O(1)})$ time randomized algorithm to determine whether there is a scheduling to download $k = |D|$ data items before time t and the number of channel switches is at most h, where n is the total number of channels.*

Proof. By Lemma 2, we can construct a circuit $H_{t,h,n} = \sum_{i=1}^{n} F_{i,t,h,k}$ in polynomial time. It is easy to see that there is a scheduling for downloading the k data items before time t and with h channel switches, if and only if the sum product expansion of $H_{t,h,n}$ has a multilinear monomial $(x_1, \ldots, x_k)Y$.

Replace each x_i by a vector $w_i = w_0^T + v_i^T$, where w_0 is the all-zeros vector of dimension k, and v_i is a binary vector of dimension k with its ith element 1 and all other elements 0. Assuming $k = 3$, we define the following operations:

$$v_a \cdot v_b = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \cdot \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} (a_1 + b_1)(\text{mod } 2) \\ (a_1 + b_2)(\text{mod } 2) \\ (a_1 + b_3)(\text{mod } 2) \end{pmatrix} \tag{1}$$

$$(v_a + v_b) \cdot v_c = v_a \cdot v_c + v_b \cdot v_c. \tag{2}$$

By Eqs. (1) and (2), for any k-dimensional binary vector $w' = w_0 + v$, we have $w'^2 = w_0^2 + 2w_0 \cdot v + v^2 = w_0 + 2(w_0 \cdot v) + w_0 = 2(w_0 \cdot v) + 2w_0 = 0$, because the coefficients are in the field of G_2. The replacement $x_i = w_i (i = 1, \dots, m)$ makes all the non-multilinear monomials become zero. Meanwhile, all the multilinear monomials remain non-zero. Hence, it is clear that there is a scheduling to download all the data items in D before time t and with at most h channel switchings if and only if $H_{t,h,n|x_i=w_i(i=1,\dots,k)}$ is a non-zero polynomial. The variables in Y make it impossible to have cancelation when adding two identical multilinear monomials, which can be generated from different paths with variables in $\{x_1, \dots, x_k\}$. It is well known that randomized algorithms can be used to check whether a circuit is identical to zero in polynomial time [17,9].

The algorithm generates less than 2^k terms during the computing process since there are at most 2^k distinct binary vectors. Therefore, the computational time is $O(2^k(nht)^{O(1)})$. $\square$

Example. Let $H_1 = x_1x_2y_1 + x_2^2y_2$ and $H_2 = x_1^2y_1 + x_2^2y_2$. Consider the replacement $x_1 = w_1 = \binom{0}{0} + \binom{1}{0}$ and $x_2 = w_2 = \binom{0}{0} + \binom{0}{1}$. We have the following steps of operations.

$$\begin{aligned}
H_1|x_1 = w_1, x_2 = w_2 &= \left(\begin{pmatrix}0\\0\end{pmatrix} + \begin{pmatrix}1\\0\end{pmatrix}\right)\left(\begin{pmatrix}0\\0\end{pmatrix} + \begin{pmatrix}0\\1\end{pmatrix}\right)y_1 + \left(\begin{pmatrix}0\\0\end{pmatrix} + \begin{pmatrix}0\\1\end{pmatrix}\right)^2 y_2 \\
&= \left(\begin{pmatrix}0\\0\end{pmatrix} + \begin{pmatrix}0\\1\end{pmatrix} + \begin{pmatrix}1\\0\end{pmatrix} + \begin{pmatrix}1\\1\end{pmatrix}\right)y_1 + \left(\begin{pmatrix}0\\0\end{pmatrix} + \begin{pmatrix}0\\1\end{pmatrix} + \begin{pmatrix}0\\1\end{pmatrix} + \begin{pmatrix}0\\0\end{pmatrix}\right)y_2 \\
&= \left(\begin{pmatrix}0\\0\end{pmatrix} + \begin{pmatrix}0\\1\end{pmatrix} + \begin{pmatrix}1\\0\end{pmatrix} + \begin{pmatrix}1\\1\end{pmatrix}\right)y_1 + \left(2\begin{pmatrix}0\\0\end{pmatrix} + 2\begin{pmatrix}0\\1\end{pmatrix}\right)y_2 \\
&= \left(\begin{pmatrix}0\\0\end{pmatrix} + \begin{pmatrix}0\\1\end{pmatrix} + \begin{pmatrix}1\\0\end{pmatrix} + \begin{pmatrix}1\\1\end{pmatrix}\right)y_1 + (0+0)y_2 \\
&= \left(\begin{pmatrix}0\\0\end{pmatrix} + \begin{pmatrix}0\\1\end{pmatrix} + \begin{pmatrix}1\\0\end{pmatrix} + \begin{pmatrix}1\\1\end{pmatrix}\right)y_1 + 0 \\
&= \left(\begin{pmatrix}0\\0\end{pmatrix} + \begin{pmatrix}0\\1\end{pmatrix} + \begin{pmatrix}1\\0\end{pmatrix} + \begin{pmatrix}1\\1\end{pmatrix}\right)y_1 \\
&\neq 0 \\
H_2|x_1 = w_1, x_2 = w_2 &= \left(\begin{pmatrix}0\\0\end{pmatrix} + \begin{pmatrix}1\\0\end{pmatrix}\right)^2 y_1 + \left(\begin{pmatrix}0\\0\end{pmatrix} + \begin{pmatrix}0\\1\end{pmatrix}\right)^2 y_2 \\
&= \left(\begin{pmatrix}0\\0\end{pmatrix} + \begin{pmatrix}1\\0\end{pmatrix} + \begin{pmatrix}1\\0\end{pmatrix} + \begin{pmatrix}0\\0\end{pmatrix}\right)y_1 + \left(\begin{pmatrix}0\\0\end{pmatrix} + \begin{pmatrix}0\\1\end{pmatrix} + \begin{pmatrix}0\\1\end{pmatrix} + \begin{pmatrix}0\\0\end{pmatrix}\right)y_2 \\
&= \left(2\begin{pmatrix}0\\0\end{pmatrix} + 2\begin{pmatrix}1\\0\end{pmatrix}\right)y_1 + \left(2\begin{pmatrix}0\\0\end{pmatrix} + 2\begin{pmatrix}0\\1\end{pmatrix}\right)y_2 \\
&= (0+0)y_1 + (0+0)y_2 \\
&= 0.
\end{aligned}$$

H_1 is a polynomial that contains a multilinear monomial. It becomes nonzero after replacement. H_2 is a polynomial without multilinear monomials. It becomes zero after the replacement. If we just download a subset of l data items from set D, we have the following theorem that involves two layers of randomization.

Theorem 3. *There is an $O(2^l(hnt)^{O(1)})$ time randomized algorithm to determine whether there is a scheduling to download l data items from D in at most t time units and at most h channel switches.*

Proof. By Lemma 2, we can construct a polynomial $H_{t,h,l} = \sum_{i=1}^{l} F_{i,t,h,l}$ in polynomial time. Replace each x_i by a vector $w_i = w_0^T + v_i^T$, where w_0 is the all-zeros vector of dimension l, and v_i is a random distinct vector of dimension l. The replacement $x_i = w_i$ $(i = 1, \dots, k)$ makes all monomials which have non-multilinear monomial in the x part become zero.

Therefore, there is a scheduling to download l data items from D before time t and with the number of switches no more than h if and only if $H_{t,h,k|x_i=w_i\ (i=1,\dots,k)}$ is not a zero polynomial in the field of G_2. Assume that the product expansion of $H_{t,h,l}$ has a multilinear monomial $(x_{i_1}, \dots, x_{i_l})Y$, where Y is a multilinear monomial with variables not in $x_1, \dots, x_k$. For a series of randomly assigned vectors with dimension l: $v_{j_1}, \ldots, v_{j_l}$, the probability that v_{j_i} is a linear combination of $v_{j_1}, \ldots, v_{j_{i-1}}$ is at most $\frac{2^{i-1}}{2^l} = \frac{1}{2^{l-i+1}}$. Therefore, with probability at most $\sum_{i=1}^{l} \frac{1}{2^{l-i+1}}$, v_{j_i} is a linear combination of $v_{j_1}, \ldots, v_{j_{i-1}}$ for some $i \le l$. When $v_{j_i}, \ldots, v_{j_l}$ are linearly independent, the product of $v_{j_1}, \ldots, v_{j_l}$ is nonzero. Every multilinear monomial in the product expansion has different variables to form Y since it is determined by a unique path to generate the polynomial. Therefore, for those random vectors v_i, every multilinear monomial has a chance of at least $1 - \frac{3}{4} = \frac{1}{4}$ be nonzero. Therefore, if there is a solution $H_{t,h,k|x_i=w_i(i,1,\ldots,l)}$ with random assignment Y is not zero in the field of G_2 with probability at least $\frac{1}{4}$.

After the replacements, less than $2l$ terms are generated since there are at most $2k$ different vectors for a group of Z_2^l. The coefficient of each vector is kept as a polynomial size circuit. Therefore, the computational time of our algorithm is $O(2^l(hnt)^{O(1)})$, and if we run it 30 times, the error rate is $(\frac{3}{4})^{30} < 0.0002$. $\square$

5. Conclusions

In this paper, we take both access time and channel switchings into consideration to investigate the minimum constraint data retrieval problem in multi-channel data broadcast environments. We prove that this problem is NP-hard, and design an algebraic algorithm. The algorithm proposed can detect whether a given data retrieval problem has a solution with access time t and number of switchings h in $O(2^k(hnt)^{O(1)})$ time, where n is the number of channels and k is the number of requested data items. It can be used in almost any broadcast environment, in which the data access frequencies, data sizes, and channel bandwidths can all be non-uniform. We also discuss a randomized algorithm to determine whether there is a scheduling to download l data items from D in at most t time and at most h switches.

Acknowledgement

This work has been supported by the National Natural Science Foundation of China (grant number 61033002).

References

[1] S. Acharya, R. Alonso, M. Franklin, S. Zdonik, Broadcast disks: data management for asymmetric communication environments, in: The 1995 ACM Special Interest Group on Management of Data Conference, 1995, pp. 199–210.
[2] E. Ardizzoni, A.A. Bertossi, S. Ramaprasad, R. Rizzi, M.V.S. Shashanka, Optimal skewed data allocation on multiple channels with flat broadcast per channel, IEEE Transactions on Computers 54 (5) (2005) 558–572.
[3] K. Foltz, L. Xu, J. Bruck, Scheduling for efficient data broadcast over two channels, in: The 2004 International Symposium on Information Theory, 2004, pp. 113–116.
[4] J.L. Huang, M.S. Chen, Broadcast program generation for unordered queries with data replication, in: The 2003 ACM Symposium on Applied Computing, 2003, pp. 866–870.
[5] J.L. Huang, M.S. Chen, W.C. Peng, Broadcasting dependent data for ordered queries without replication in a multi-channel mobile environment, in: The 2003 International Conference on Data Engineering, 2003, pp. 692–694.
[6] J.L. Huang, M.S. Chen, Dependent data broadcasting for unordered queries in a multiple channel mobile environment, IEEE Transactions on Knowledge and Data Engineering 16 (9) (2004) 1143–1156.
[7] A.R. Hurson, A.M. Munoz-Avila, N. Orchowski, B. Shirazi, Y. Jiao, Power aware data retrieval protocols for indexed broadcast parallel channels, Pervasive and Mobile Computing 2 (1) (2006) 85–107.
[8] T. Imielinski, S. Viswanathan, B.R. Badrinath, Data on air: organization and access, IEEE Transactions on Knowledge and Data Engineering 9 (1996) 353–372.
[9] I. Koutis, Faster algebraic algorithms for path and packing problems, in: The 2008 International Colloquium on Automata, Languages and Programming, in: LNCS, vol. 5125, Springer, 2008, pp. 575–586.
[10] G. Lee, M.S. Yeh, S.C. Lo, A. Chen, A strategy for efficient access of multiple data items in mobile environments. in: The 2002 International Conference on Mobile Data Management, 2002, pp. 71–78.
[11] S. Jung, B. Lee, S. Pramanik, A tree-structured index allocation method with replication over multiple broadcast channels in wireless environment, IEEE Transaction on Knowledge and Data Engineering 17 (3) (2005) 311–325.
[12] S.C Lo, A.L.P. Chen, Optimal index and data allocation in multiple broadcast channels, in: The 2000 International Conference on Data Engineering, 2000, pp. 293–302.
[13] Z. Lu, Y. Shi, W. Wu, B. Fu, Efficient data retrieval scheduling for multi-channel wireless data broadcast, in: The 31st Annual IEEE International Conference on Computer Communications (IEEE INFOCOM 2012) (in press).
[14] K. Prabhakara, K.A. Hua, J. Oh, Multi-level multi-channel air cache designs for broadcasting in a mobile environment, in: The 2000 International Conference on Data Engineering, 2000, pp. 167–176.
[15] Y. Shi, X. Gao, J. Zhong, W. Wu, Efficient parallel data retrieval protocols with MIMO antennae for data broadcast in 4G wireless communications, in: The 2010 International Conference on Database and Expert Systems Applications, 2010.
[16] N. Shivakumar, S. Venkatasubramanian, Efficient indexing for broadcast based wireless systems, ACM/Baltzer Mobile Network and Application 1 (4) (1996) 433–446.
[17] R. Williams, Finding paths of length k in $O*(2^k)$ time, Information Processing Letters 109 (6) (2009) 315–318.
[18] J. Xu, W.C. Lee, X. Tang, Q. Gao, S. Li, An error-resilient and tunable distributed indexing scheme for wireless data broadcast, IEEE Transactions on Knowledge and Data Engineering 18 (3) (2006) 392–404.
[19] Y. Yao, X. Tang, E.P. Lim, A. Sun, An energy-efficient and access latency optimized indexing scheme for wireless data broadcast, IEEE Transactions on Knowledge and Data Engineering 18 (8) (2006) 1111–1124.
[20] W.G. Yee, S.B. Navathe, E. Omiecinski, C. Jermaine, Efficient data allocation over multiple channels at broadcast servers, IEEE Transactions on Computers 51 (10) (2002) 1231–1236.
[21] B. Zheng, W.C. Lee, P. Liu, D.L. Lee, X. Ding, Tuning on-air signatures for balancing performance and confidentiality, IEEE Transactions on Knowledge and Data Engineering 21 (12) (2009) 1783–1797.
[22] B. Zheng, X. Wu, X. Jin, D.L. Lee, Tosa: a near-optimal scheduling algorithm for multi-channel data broadcast, in: The 2005 International Conference on Mobile Data Management, 2005, pp. 29–37.