Efficient Approximation Algorithms for the Largest Weight Data Retrieval Problem via Maximum Sectionalized Coverage *

Longkun Guo¹, Wenxing Zhu¹, Hong Shen²,³

¹College of Mathematics and Computer Science, Fuzhou University, China
²School of Information Science and Technology, Sun Yat-Sen University, China
³School of Computer Science, University of Adelaide, Australia

Abstract. In mobile networks, wireless data broadcast is a powerful approach for disseminating a large number of data copies to a great number of clients. The largest weight data retrieval (LWDR) problem, first addressed by [15], is to schedule the download of a specified subset of data items in a given time interval, such that the weight of the downloaded items will be maximized.

In this paper, we present an approximation algorithm with ratio 0.632 for LWDR via approximating the maximum sectionalized coverage (MSC) problem, a generalization of the maximum coverage problem which is one of the most famous NP-complete problems. Let \( F = \{ F_1, \ldots, F_N \} \), in which \( F_i = \{ S_{i,j} \mid j = 1, 2, \ldots \} \) is a collection of subsets of \( S = \{ u_1, u_2, \ldots, u_n \} \), and \( w(u_i) \) be the weight of \( u_i \). MSC is to select at most one \( S_{i,j} \) from each \( F_i \), such that \( \sum_{u_i \in S'} w(u_i) \) is maximized, where \( S' = \bigcup_{i=1}^N S_{i,j} \). First, the paper presents a factor-0.632 approximation algorithm for MSC by giving a novel linear programming (LP) formula and employing the randomized LP rounding technique. By reducing from the maximum 3 dimensional matching problem, the paper then shows that MSC is NP-complete even when every \( S \in F_i \) is with cardinality 2, i.e. \( |S| = 2 \). Last but not the least, the paper gives a method transforming any instance of LWDR to an instance of MSC, and shows that an approximation for MSC can be applied to LWDR almost preserving the approximation ratio. That is a factor-0.632 approximation for LWDR, improving the currently best ratio of 0.5 in [15].

Keywords: Maximum sectionalized coverage, the largest weight data retrieval problem, approximation algorithm, randomized linear programming rounding.

⋆ This research work is supported by Natural Science Foundation of China #61300025, Natural Science Foundation of Fujian Province #2012J05115, Doctoral Fund of Ministry of Education of China for Young Scholars #20123514120013 and Fuzhou University Development Fund #2012-XQ-26.
1 Introductions

In recent years, wireless data broadcast has been gradually considered as an attractive data propagation scheme for transferring public information to a large number of specified mobile devices, in applications ranging from satellite communications to wireless mobile ad hoc networks. Most wireless broadcast is between base stations and battery-limited mobile devices, where a base station emits public information (such as stock marketing, weather, and traffic) via a number of parallel channels, and mobile devices within a limited area, using an antenna (or multiple antennae), listen to the channels and obtain required data packages. The base stations can coordinate information propagation to cover a larger scope; Within the covered scope, the mobile client can move freely among different areas while keep listening to the channels for downloading the required data packages; An antenna, equipped in the mobile client, can only listen to a channel at one time, but it can switch between the channels by adjusting its frequency.

Wireless data broadcast has already shown its advantages in wireless networks: Possible power saving, throughput improvement, the communication efficiency that every transmission by a base station can be received by all nodes which lie within its communication range, and so on. At the same time, it brings a number of challenges for data propagation technologies, which have become popular research topics in recent years, such as indexing technique, data scheduling, and data retrieval. Indexing technique and data scheduling are mainly based on the server side. The former topic investigates the structure of the indexing information that emitted by the server to boost clients on finding the locations of requested data items among the channels. The latter investigates that how the server allocates the data items in proper channels and at proper time slots, such that clients can quickly accomplish download tasks. Differently, data retrieval is based on the client side. The goal is to find a data retrieval sequence retrieving all requested data items among the channels such that the total access latency is minimized, where the access latency is the length of the period from the starting time when the client knows the offset of each requested data item (by index techniques) to the ending time when the client downloads all the requested data items.

Let $D = \{d_1, d_2, \ldots, d_N\}$ be a set of data items to be downloaded, with weights $w_1, \ldots, w_N$ respectively. Assume the items of $D$ are broadcast in channels $c_1, c_2, \ldots, c_m$ in a given time interval, which is separated into time slots $t_1, t_2, \ldots, t_T$. Given the index information of the data items in a time interval (as above by the indexing technique), the largest weight data retrieval (LWDR) problem is to schedule to download the data items of $D$ in the given time interval, such that the weight of the downloaded items will be maximized [15]. When the items are with equal weight, LWDR will reduce to the Largest Number Data Retrieval (LNDR) problem, which is to maximize the number of the downloaded items in the given time interval. Through this paper, we consider only LWDR with one antenna. We say there exist conflicts between two data items iff it is impossible to retrieve both of them in the same broadcast cycle. There exist
two well-known conflicts for single-antenna data retrieval problems (including LWDR): (1) two requested data items at two same time slots; (2) two adjacent time slots of different channels. The first conflict is because one antenna can retrieve one channel in one time slot. The reason for the second conflict is that (the antenna) switching between different channels takes typically one time slot. Paper [15] has proposed a polynomial time exact algorithm to solve LWDR with the first conflict only. For LWDR with both the two conflicts, the same authors have proposed a factor $-0.5$ approximation. This paper designs a factor $-0.632$ approximation algorithm for LWDR with both the two conflicts via sectionalized maximum coverage. Below is the definition of the maximum sectionalized coverage (MSC):

**Definition 1** Given $\mathbb{F} = \{F_1, \ldots, F_N\}$, in which $F_i = \{S_{i,j}\}$ is a collection of subsets of $S = \{u_1, u_2, \ldots, u_n\}$, and $w(u_i)$ is the weight of $u_i$, MSC is to select at most $k$ element $S_{i,j}$ from each $F_i$, such that $\sum_{u_i \in S'} w(u_i)$ is maximized, where $S' = \bigcup_{i=1}^{\#} S_{i,j}$.

The classical maximum coverage problem (or namely, max $k$-cover) is, given an integer $k$ and a collection $\mathbb{C}$ of subsets of $S = \{u_1, u_2, \ldots, u_n\}$ with weight $w(u_i)$ for each $u_i$, to maximize the weight of covered elements using at most $k$ sets in $\mathbb{C}$. Clearly, MSC generalizes the maximum coverage problem, since MSC reduces to the latter problem if $\mathbb{F}$ contains only one element $F_1$. However, we study MSC only for the case $k = 1$. So while no confusion arises, we use MSC assuming that $k = 1$, and use LWDR assuming that both two conflicts are considered.

### 1.1 Related Works

For the theoretical side, to the best of our knowledge, this paper is the first to address MSC. However, the maximum coverage problem has been well investigated. It is well-known that the greedy algorithm, which repeats selecting from $\mathbb{C}$ the set currently covering maximum-weight uncovered elements (until all elements of $S$ are covered), is with an approximation ratio $1 - \frac{1}{e}$ [10]. The ratio is best possible, since this problem admits no factor $1 - \frac{1}{e} + \epsilon$ approximation even when all $w(u_i)$ are equal, under the assumption that $\mathcal{P} \neq \mathcal{NP}$ [3].

It is worth to note that, unlike the case for the maximum coverage problem, adopting similar idea of the greedy algorithm to MSC can only result in an approximation algorithm with ratio $0.5$ (The ratio can not be analyzed better, since the analysis for the algorithm is already tight). For a given collection $\mathbb{C}$ of subsets of $S = \{u_1, u_2, \ldots, u_n\}$, the minimum set cover (SC) problem is to compute a subset $\mathbb{C}' \subseteq \mathbb{C}$, such that every element in $S$ belongs to at least one member of $\mathbb{C}'$. It has been shown that SC can be approximated within a factor of $1 + \ln |S|$ [12] and is not approximable within $c \log n$, for some $c > 0$ [3]. When the cardinality of all sets in $\mathbb{C}$ are bounded by a given constant $k$, SC remains $\mathcal{APX}$-complete and is approximable within $\sum_{i=1}^{k} \frac{1}{i} - 1/2$ [2]. Moreover, if the number of occurrences of any element in $\mathbb{C}$ is also bounded by a constant $c \geq 2$,
SC remains APX-complete \cite{15} and approximable within a factor $c$ for both weighted and unweighted SC \cite{110}.

For the practical and networking application side, existing literature has discussed data retrieval in depth in the client side of wireless data broadcast. The problem is firstly studied under the assumption that each client is equipped with one antenna, to download multiple requested data items for one single request. Three heuristic schemes have been proposed in \cite{11} to compute a data retrieval schedule with minimum times of switching between the channels. Later, two algorithms have been proposed in \cite{18} to extend data retrieval technique to the case that clients are equipped with multiple antennae. Considering neither of the two conflicts, paper \cite{4} gives an algorithm to find a data retrieval schedule with minimal access latency and with the times of switching between the channels bounded by a given number. A parameterized heuristic scheme has been proposed in \cite{15}, attempting to solve the minimum cost data retrieval problem and find a data retrieval schedule with minimized energy consumption. A factor $-0.5$ approximation algorithm for LWDR also has been presented in the same paper. The key idea of the approximation is first to convert the relationship between the broadcast data items and the time slots to a bipartite graph, and then obtain an approximation solution for LWDR via maximum matching in the bipartite graph. Different to paper \cite{15}, paper \cite{8} converts the relationship between data items and time slots to a directed acyclic graph, and designs algorithms to compute a nearly optimal access pattern for both one antenna and multiple antennae scenarios.

1.2 Our Technique and Main Results

The paper first gives a novel linear programming (LP) formula for MSC, and rounds an optimum solution of the formula based on the randomized linear programming technique, obtaining a factor $-0.632$ approximation algorithm.

**Theorem 2** MSC admits a polynomial-time approximation algorithm with a ratio of $(1 - \frac{1}{K})^K \geq 0.632$, where $K$ is the maximum occurrence times of $u_i$ in all $S_{i,j}$.

Then the paper proves the $\mathcal{NP}$-completeness of MSC, by giving a reduction from the 3-dimensional perfect matching (3DM) problem that is known $\mathcal{NP}$-complete. We note that paper \cite{15} has shown the $\mathcal{NP}$-completeness of LNDR. However, their proof can not be easily extended to show the $\mathcal{NP}$-completeness of MSC. At last, we show that any instance of LWDR can be transfer to an instance of MSC, in which the approximation algorithm can be adopted. This results an approximation algorithm for LWDR and improves the currently best ratio 0.5:

**Theorem 3** LWDR admits a polynomial-time approximation algorithm with a ratio of 0.632.

The remainder of this paper is organized as follows: Section II presents a factor $-0.632$ approximation algorithm for MSC as well as the ratio proof; section III gives the
\(N \mathcal{P}\)– completeness proof of MSC; Section IV gives a transformation from LWDR to MSC, and shows the approximation algorithm for MSC can be extended to LWDR preserving the ratio 0.632; Section V concludes this paper.

2 A Factor–0.632 Approximation Algorithm for MSC

In this section, we shall first give a novel linear programming (LP) formula, which is actually an LP formula for the dual of the maximum sectionalized coverage (MSC) problem. Then we present an approximation algorithm via randomized rounding an optimum solution to the LP formula.

2.1 The LP Formula

It is not easy to obtain an apt LP formula immediately from the definition of MSC. Let \(F = \{F_1, \ldots, F_N\}\), in which \(F_i = \{S_{i,0}\}\) and \(S_{i,j}\) be a subset of \(S = \{u_1, u_2, \ldots, u_n\}\). Each element \(u_i \in S\) is assigned with weight \(w(u_i)\). To obtain the LP formula, we add the following dummy parts:

1. Set \(F := F \cup F'\), in which \(F' = \{F_{N+1}, \ldots, F_{N+n}\}\), \(F_{N+i} = \{S_{N+i,1}\}\), and \(S_{N+i,1} = \{u_i\}\);
2. Assign \(S_{i,j}\) with a new cost \(c(S_{i,j}) = 0\) for \(i \leq N\), and \(S_{i,1}\) with \(c(S_{i,j}) = 1\) for \(i \geq N + 1\).

Formally, MSC is to solve the following integer programming (IP) formula:

\[
\min \sum_{i=1}^{N+n} \sum_{S_{i,j} \in F_i} c(S_{i,j}) \cdot \left( \sum_{k: u_k \in S_{i,j}} w(u_k) \cdot x_{i,j} \right),
\]

such that
\[
\sum_{S_{i,j} \in F_i} x_{i,j} = 1 \quad i = 1, \ldots, N
\]
\[
\sum_{S_{i,j} \in F_i} x_{i,j} \leq 1 \quad i = N + 1, \ldots, N + n
\]
\[
\sum_{i,j: S_{i,j} \supseteq \{u_k\}} x_{i,j} \geq 1 \quad k = 1, \ldots, n
\]
\[
x_{i,j} \in \{0, 1\} \quad i = 1, \ldots, N + n; S_{i,j} \in F_i
\]

where \(x_{i,j} = 1\) if \(S_{i,j}\) is selected and \(x_{i,j} = 0\) otherwise. The above formula is actually a dual of the original MSC problem. Intuitively, let \(W = \sum_{i=1}^{n} w_i\), let \(c_{IP}\) be the value of the objective function of an optimal solution IP \(\textbf{[1]}\), and let \(w_{OPT}\) be the weight of an optimal solution to MSC, we have

\[
W = c_{IP} + w_{OPT}.
\]"
Algorithm 1 A randomized algorithm for MSC.

Input: \( F = \{F_1, \ldots, F_N\} \), where \( F_i = \{S_{i,j}\} \) is a collection of subsets of \( S = \{u_1, u_2, \ldots, u_n\} \) with \( w(u_i) \) for each \( u_i \).

Output: \( C' \) a solution to MSC.

1. \( C' := \emptyset \);
2. Solve LP \( \text{(3)} \) against \( F \) by Karmarkar’s algorithm \[17\], and obtain an optimal solution \( \chi = (x_{1,1}, \ldots, x_{i,j}, \ldots) \);
3. For \( i = 1 \) to \( N \) do
   (a) Set \( S_i := S_{i,j} \) with probability \( x_{i,j}^\ast \); /* \( S_i \) is the element selected for \( F_i \). */
   (b) \( C' := C' \cup \{S_i\} \);
4. Return \( C' \).

\[
\min_{i=1}^{N+n} \sum_{S_{i,j} \in F_i} c(S_{i,j}) \cdot \left( \sum_{k: u_k \in S_{i,j}} w(u_k) \cdot x_{i,j} \right) 
\]

such that
\[
\sum_{S_{i,j} \in F_i} x_{i,j} = 1 \quad i = 1, \ldots, N
\]
\[
\sum_{S_{i,j} \in F_i} x_{i,j} \leq 1 \quad i = N+1, \ldots, N+n
\]
\[
\sum_{i,j: S_{i,j} \supseteq \{u_k\}} x_{i,j} \geq 1 \quad k = 1, \ldots, n
\]
\[
0 \leq x_{i,j} \leq 1 \quad i = 1, \ldots, N+n; S_{i,j} \in F_i
\]

Note that \( c(S_{i,j}) = 0 \) for every \( i \leq N \), so the objective function of LP \( \text{(3)} \) can be simplified as to minimize:

\[
\min_{i=N+1}^{N+n} c(S_{i,1}) \cdot x_{i,1} \cdot w(u_{i-N}) = \min_{i=N+1}^{N+n} x_{i,1} \cdot w(u_{i-N}).
\]

2.2 The Randomized Algorithm

Let \( c_{LP} \) be the value of the objective function of an optimal solution to LP \( \text{(3)} \). Then since LP \( \text{(3)} \) is a relaxation of IP \( \text{(1)} \), we have \( c_{LP} \leq c_{IP} \). Let \( \chi = (x_{1,1}^\ast, \ldots, x_{i,j}^\ast, \ldots) \) be an optimal solution to LP \( \text{(3)} \). The key idea of our algorithm is to interpret the fractional value \( x_{i,j}^\ast \) as the probability of selecting \( S_{i,j} \) for \( F_i \). Then the main steps of our algorithm is as in Algorithm \( \text{(4)} \).

Lemma 4 Algorithm \( \text{(4)} \) is a randomized \((1 - \frac{(K-1)K}{K})\)-approximation algorithm with a running time of \( O(n^{3.5}L) \) for MSC, where \( L \) is the maximum length of input and \( K \) is the maximum occurrence times of \( u_i \) in all \( S_{i,j}s \).

Proof. The running time of Algorithm \( \text{(4)} \) is easy to calculate: Step 2 of the algorithm takes \( O(n^{3.5}L) \) to run Karmarkar’s algorithm, other steps take trivial time comparing to Step 2.
For the ratio, let \( \text{SOL} \) and \( w_{\text{SOL}} \) be the output of the algorithm and its weight respectively. So from LP (4), the expected value of the output of the algorithm is

\[
E(w_{\text{SOL}}) = \sum_{i=\text{N+1}}^{\text{N+n}} x_{i,j}^* \cdot w(u_{i-N}) = w_{\text{LP}}.
\]

However, every element of \( S \) is with a probability that not in any \( S_{i,j} \in \text{SOL} \). (That is why we have \( E(w_{\text{SOL}}) = w_{\text{LP}} \leq W - w_{\text{OPT}} \)). Below is the probability that \( u_k \) is not in any \( S_{i,j} \):

\[
\prod_{i: u_k \in S_{i,j}} (1 - \sum_{j: u_k \in S_{i,j}} x_{i,j}^*) \leq \prod_{i,j: u_k \in S_{i,j}} (1 - x_{i,j}^*).
\]

Then since \( \sum_{i,j: S_{i,j} \supseteq \{u_k\}} x_{i,j}^* \) is fixed, we assume that \( f = \prod_{i,j: u_k \in S_{i,j}} (1 - x_{i,j}^*) = \lambda \cdot \sum_{i,j: S_{i,j} \supseteq \{u_k\}} x_{i,j}^* \). It is easy to see \( f \) attains maximum when

\[
\frac{\partial f}{\partial x_{i,j}^*} = \lambda \forall i, j : u_k \in S_{i,j}.
\]

That is, \( \prod_{i,j: u_k \in S_{i,j}} (1 - x_{i,j}^*) \leq \left(1 - \frac{1}{K}\right)^K \), where \( K \) is the number of the occurrence times of \( u_k \) in all \( S_{i,j} \). So any element \( u_k \) has a probability at most \( (1 - \frac{1}{K})^K \) to be absent from every \( S_{i,j} \). Now we have all the ingredients for computing \( E(w_{\text{SOL}}) \):

\[
E(w_{\text{SOL}}) = W - E(c_{\text{LP}}) - (\text{excepted weight of elements absenting every } S_{i,j}) \\
\geq W - c_{\text{IP}} - (1 - \frac{1}{K})^K \cdot w_{\text{OPT}} \\
= (1 - \left(\frac{K-1}{K}\right)^K) \cdot w_{\text{OPT}}.
\]

This completes the proof.

By simple arithmetical calculation, it is easy to see the ratio will be 0.75 when \( K = 2 \), be 0.704 when \( K = 3 \), and be 0.684 when \( K = 4 \). Further, following the inequality as in Proposition 5, the ratio of our algorithm would be \( 1 - e^{-1} \approx 0.632 \) when \( K \) is unbounded. The proof of Proposition 5 is only arithmetical calculation, we put it in the appendix due to the length limit.

**Proposition 5** For any positive integer \( K \), \( 1 - (\frac{K-1}{K})^K \geq 1 - e^{-1} \) holds.

We note that our algorithm works for MSC only for selecting at most one \( S_{i,j} \), from each \( F_i \). For further approximating the more generalized MSC problem that selects \( k \) elements from each \( F_i \), we shall try transforming the problems to the restricted \( k \)-disjoint paths problem, and try adopting the technique for approximating the restricted \( k \)-disjoint paths problem [6,7]. Besides, some evidences suggest the approximation ratio \( 1 - \frac{1}{e} \approx 0.632 \) should be tight for MSC.

One evidence is that the maximum coverage problem admits no polynomial-time
approximation algorithm with ratio better than $1 - \frac{1}{e}$ [3]. Yet we have not found a proof for the tight inapproximability of MSC. Anyhow, we would like to make the following conjecture:

Conjecture 1. MSC admits no polynomial-time approximation algorithm with ratio $1 - \frac{1}{e} + \epsilon$ for any constant $\epsilon > 0$ unless $\mathcal{P} = \mathcal{NP}$.

The derandomization follows the same line as a traditional derandomization process (e.g., derandomization of the randomized algorithm for MAX-SAT and etc as in [14]), we put it in the appendix because of length limit.

3 $\mathcal{NP}$—Completeness Proof for MSC

In this section, we show that the maximum sectionalized coverage (MSC) problem is $\mathcal{NP}$—complete by giving a reduction from the 3-dimensional perfect matching (3DM) problem, which is known $\mathcal{NP}$—complete [3]. For a given positive integer $K$ and a set $T \subseteq X \times Y \times Z$ where $X$, $Y$ and $Z$ are disjoint and $|X| = |Y| = |Z|$, 3DM is to decide whether there exists a perfect matching for $T$, i.e., a subset $M \subseteq T$ with $|M| = |X|$, such that no elements in $M$ agree in any coordinate.

Theorem 6 (Unweighted) MSC is $\mathcal{NP}$—complete.

Proof. MSC is apparently in $\mathcal{NP}$. So we need only to give the reduction from 3DM to the decision form of Unweighted MSC: Given a positive integer $K$ and $\mathcal{F} = \{F_1, \ldots, F_N\}$, where $F_i = \{S_{i,j}\}$ is a collection of subsets of $S = \{u_1, u_2, \ldots, u_n\}$, does there exist a $S_{i,j}$ of each $F_i$ such that $\sum_{i=1}^{N} |S_{i,j}| \geq K$?

For an instance of decision 3DM, the corresponding instance of MSC is constructed as below:

1. For each element of $X$, say $x_i$, add $F_i$ to $\mathcal{F}$;
2. For each element of $T$, say $\langle x_i, y_j, z_k \rangle$, add a set $S = \{y_j, z_k\}$ to $F_i$.

For example, given the instance of 3DM(3) as below

$$T = \{\langle x_1, y_1, z_1 \rangle, \langle x_1, y_2, z_1 \rangle, \langle x_1, y_3, z_2 \rangle, \langle x_2, y_1, z_2 \rangle, \langle x_3, y_2, z_3 \rangle, \langle x_3, y_3, z_3 \rangle\},$$

the corresponding instance of MSC will be

$$\mathcal{F}_1 = \{\{y_1, z_1\}, \{y_2, z_1\}, \{y_3, z_2\}\}$$
$$\mathcal{F}_2 = \{\{y_1, z_2\}\}$$
$$\mathcal{F}_3 = \{\{y_2, z_3\}, \{y_3, z_3\}\}$$

Since $\mathcal{F}$ can be constructed in polynomial time, it suffices to show that an instance of 3DM is feasible iff the corresponding instance of MSC has a feasible solution with $\sum_{i=1}^{N} |S_{i,j}| \geq 2|X|$ (where $|X| = N$).
Firstly, suppose there exists a perfect matching for the 3DM instance, say $M = \{(x_1, y_{i_1}, z_{k_1}), (x_2, y_{i_2}, z_{k_2}), \ldots, (x_{|X|}, y_{i_{|X|}}, z_{k_{|X|}})\}$, such that no elements in $M$ agree in any coordinate, i.e., for any $l \neq l'$, $\{x_l, y_{i_l}, z_{k_l}\} \cap \{x_{l'}, y_{i_{l'}}, z_{k_{l'}}\} = \emptyset$ holds. Let $C' = \{S_{i,j} = \{y_{i_l}, z_{k_l}\} | (x_{i}, y_{i_l}, z_{k_l}) \in M\}$. Then $C'$ is a feasible solution to MSC with $|\bigcup_{i=1}^{|X|} S_{i,j}| = 2|X|$.

Conversely, assume that $C' = \{S_{i,j} = \{y_{i_l}, z_{k_l}\} | i = 1, \ldots, N\}$ is a feasible solution to the instance of MSC. Then we have $|\bigcup_{i=1}^{|X|} S_{i,j}| \geq 2|X|$. Because $|S_{i,j}| = 2$ holds for every $i = 1, \ldots, |X|$, $S_{i,j} \cap S_{i',j'} = \emptyset$ holds for every $i \neq i'$. That is, $y_{i_l} \neq y_{i_{l'}}$ and $z_{k_l} \neq z_{k_{l'}}$ hold for every $i \neq i'$. Therefore, $M = \{(x_1, y_{i_1}, z_{k_1}), (x_2, y_{i_2}, z_{k_2}), \ldots, (x_{|X|}, y_{i_{|X|}}, z_{k_{|X|}})\}$ is a perfect matching for the given 3DM instance. This completes the proof.

Further, we have the $\mathcal{NP}$-completeness for a bounded version of MSC:

**Corollary 7** (Unweighted) MSC is $\mathcal{NP}$-complete, even if $|S| \leq 2$, $|F| \leq 3$ and the times of occurrences of any element in all $S_{i,j}$s of all $F_i$s is at most 3.

**Proof.** In the transformation of the proof of Theorem 6, the cardinality of every $S_{i,j}$ is 2, i.e., $|S_{i,j}| = 2$. Moreover, it is known that 3DM is $\mathcal{NP}$-complete even if the number of occurrences of any element in $X$, $Y$ or $Z$ is bounded by “3” \[13\]. Following the transformation, the bound “3” on element occurrences of 3DM will bound both $|F_i|$ and the occurrences of any element in all $S_{i,j}$s of all $F_i$s.

### 4 A Transformation from LWDR to MSC

The transformation is based on two observations. The first basic observation of the transformation is that LWDR can be solved in polynomial time when the number of time slots are fixed. Further, we have:

**Proposition 8** All conflict-free sequences can be computed in time $O(\delta \cdot m^3)$, where $m$ is number of channels and $\delta > 0$ is a fixed integer that bounds the total number of time slots.

**Proof.** Below is a simple exact algorithm that satisfies Proposition 8.

1. **For** each sequence of data items broadcast in the time interval **do**
   - Check whether there exist conflicts in the sequence;
2. **Return** the conflict-free sequence containing most different data items.

Apparently, the number of the sequences is at most $m^\delta$, and checking conflicts in a sequence take at most $\delta$ time. So the running time of the above algorithm is $O(\delta \cdot m^3)$. This completes the proof.

The second observation is that, most of the hardness to solve LWDR comes from conflict-2. Note that LWDR without conflict-2 is polynomial solvable, and LWDR with conflict-2 is $\mathcal{NP}$-hard.
Combining the two observations, the key idea of the transformation is to divide an instance of LWDR to a number of small subinstances, each of which contains at most with $2 \cdot \delta$ time slots, such that the divided version of LWDR is immediately an instance of MSC, and that a factor $-\alpha$ approximation to divided LWDR is a factor $-(\alpha + \frac{1}{2})$ approximation to original LWDR. The formal layout of the algorithm is as in Algorithm 2:

**Lemma 9** For any instance of LWDR, Algorithm 2 will compute a corresponding instance of MSC of polynomial size in time $O(p \cdot (W - w_{OPT}))$, such that $w_{OPT}(1 - \frac{1}{2}) \leq W - c_{LP}$, where $p$ is a polynomial on the length of input, $W = \sum_i w_i$, $w_{OPT}$ is the weight of an optimum solution to the LWDR instance, and $c_{LP}$ is the value of the objective function of an optimum solution to LP (3) against the output of the algorithm.

Proof. First of all, for every $l \geq 2$, we have $(l - 1) \cdot \delta + 1 \leq o_l \leq l \cdot \delta - 1$ and $(l - 2) \cdot \delta + 1 \leq o_{l-1} \leq (l - 1) \cdot \delta - 1$. So $o_l - o_{l-1} \leq 2\delta$ holds, and hence the output of Algorithm 2 is within a polynomial size.

Secondly and apparently, Algorithm 2 repeats Step 4 for at most $O(p \cdot (W - w_{OPT}))$ iterations, each iteration takes $O(n^{3.5}L)$ to solve the LP formula. Hence, Algorithm 2 is with a pseudo polynomial time $O(p \cdot (W - w_{OPT}))$. Note that the time complexity can be improved to polynomial time by applying the classical technique of developing a FPTAS [14].

Thirdly, for the approximation ratio, we shall only give the key idea of the proof. Following Algorithm 2 $o_l$ is the time slot, ignoring all data items of which cause smaller weight increment on $c_{LP}$, comparing to other time slots in $\{\delta \cdot (l - 1) + 1, \ldots, \delta \cdot l - 1\}$. That is, selecting time slot $o_l$ as a dividing time slot will increase at most $\frac{1}{2} \cdot w_{o_l}$ on $c_{LP}$, and hence decrease at most $\frac{1}{2} \cdot w_{o_l}$ on the weight of the solution, where $w_{o_l}$ is the weight of the sequence between time slot $\delta \cdot (l - 1) + 1$ and $\delta \cdot l - 1$ in the optimum solution of LP (3).

5 Conclusion

In this paper, we introduced the maximum sectionalized coverage (MSC) problem, and presented its $\mathcal{NP}$-completeness proof. Then we gave a novel linear programming (LP) formula for MSC. Based on randomized rounding a solution to the formula, we obtained a factor $-0.632$ approximation algorithm, which is one of the main works of this paper. We conjectured that the approximation ratio achieves the best possible performance guarantee, observing that the close related maximum coverage problem (or namely, $k$-set cover) admits no approximation with ratio better than $1 - \frac{1}{e}$ [3]. We also note that our algorithm works for MSC only for selecting at most one $S_{i,j}$ from each $F_i$. Last but not the least, we showed that the largest weight data retrieval (LWDR) problem can be transferred to MSC, such that the approximation algorithm can be adopted resulting in a ratio of 0.632. To the best of our knowledge, this is the first approximation for LWDR with ratio better than 0.5. In future, we shall further investigate the inapproximability of MSC, and design approximation algorithm for a more generalized version of MSC which selects $k$ sets from each section $F_i$. 
Algorithm 2 A Transformation from LWDR to MSC.

**Input:** A fixed integer $\delta \geq 2$, and an instance of LWDR (i.e., a set of data items to download $D = \{d_1, d_2, \ldots, d_n\}$ with weights $w_1, \ldots, w_n$, together with their occurrences in channels $c_1, \ldots, c_m$ and time slots $t_1, t_2, \ldots, t_T$);

**Output:** An instance of MSC, i.e., $F = \{F_1, \ldots, F_N\}$, where $F_i = \{S_{i,j}\}$ is a collection of subsets of $D = \{d_1, d_2, \ldots, d_n\}$ with $w(d_i) = w_i$.

1. Set $F := \{F_1, \ldots, F_{\lceil T/\delta \rceil}\}$, $z := +\infty$; /* $z$ is the value of the objective function of an optimum solution to LP (3) against $F$;*/
2. Set $O' := \{o'_1, o'_2, \ldots, o'_{\lceil T/\delta \rceil}\}$, $O := \{o_1, o_2, \ldots, o_{\lceil T/\delta \rceil}\}$ and $o'_i := o_i := \delta \cdot i$ initially; /* $O'$ and $O$ keep the subscripts of time slots that divide the given LWDR instance.*/
3. For $l = 1$ to $\lceil T/\delta \rceil$ do
   - Compute $F_l$, the collection of sets corresponding to all conflict-free sequences in time slots $\{t_{o'_{l-1}+1}, \ldots, t_{o'_{l-1}}\}$; /* Following the proof of Proposition 8 all conflict-free sequences can be computed in time $O((o'_{l-1} - o'_{l-2}) \cdot m^{o'_{l-1} - o'_{l-2}})$.*/
4. Solve LP (3) against $F := \{F_1, \ldots, F_{\lceil T/\delta \rceil}\}$;
5. If $z > c_{LP}$ then /*A better division is found.*/
   - (a) Set $z := c_{LP}$, $o_l := o'_l$; /* Recall that $c_{LP}$ is the value of the objective function of an optimum solution to LP (3).*/
   - (b) Set $i := 1$, $j := 1$ and go to Step 3; /* The division is changed better, so start over.*/
6. If $j = \delta - 1$ then
   - (a) Set $i := i + 1$;
   - (b) Set $O' := O$; /* Restore $O'$ from $O$.*/
   - (c) Set $o'_i := \delta \cdot i - j$ and go to Step 3;
   Else
   - (a) Set $j := j + 1$;
   - (b) Set $O' := O$; /* Restore $O'$ from $O$.*/
   - (c) Set $o_i := \delta \cdot i - j$ and go to Step 3;
7. If $i = \lceil T/\delta \rceil$ then Return $F$; /*There exists no $i \in \{1, \ldots, \lceil T/\delta \rceil \}$ and $j \in \{1, \ldots, \delta - 1\}$, for which the replacement of $o'_i$ by $\delta \cdot i - j$ decreases $z$.*/
References

1. Reuven Bar-Yehuda and Shimon Even. A linear-time approximation algorithm for the weighted vertex cover problem. *Journal of Algorithms*, 2(2):198–203, 1981.
2. Rong-chii Duh and Martin Fürer. Approximation of k-set cover by semi-local optimization. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 256–264. ACM, 1997.
3. Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.
4. Xiaofeng Gao, Zaixin Lu, Weili Wu, and Bin Fu. Algebraic data retrieval algorithms for multi-channel wireless data broadcast. *Theoretical Computer Science*, pages 1–8, 2011.
5. Michael R Garey and David S Johnson. Computer and intractability. *A Guide to the Theory of NP-Completeness*, 1979.
6. Longkun Guo, Kewen Liao, Hong Shen, and Peng Li. Efficient approximation algorithms for computing k-disjoint restricted shortest paths. *27th ACM Symposium on Parallelism in Algorithms and Architectures*, page accepted, 2015.
7. Longkun Guo, Hong Shen, and Kewen Liao. Improved approximation algorithms for computing k disjoint paths subject to two constraints. In *Computing and Combinatorics, 19th International Conference, COCOON 2013, Hangzhou, China, June 21-23, 2013. Proceedings*, pages 325–336, 2013.
8. Ping He, Hong Shen, and Hui Tian. Efficient approximation algorithm for data retrieval with conflicts in wireless networks. In *Proceedings of International Conference on Advances in Mobile Computing Multimedia, MoMM ’13*, pages 224:224–224:233, New York, NY, USA, 2013. ACM.
9. Dorit S Hochbaum. Approximation algorithms for the set covering and vertex cover problems. *SIAM Journal on computing*, 11(3):555–556, 1982.
10. Dorit S Hochbaum. Approximating covering and packing problems: set cover, vertex cover, independent set, and related problems. In *Approximation algorithms for NP-hard problems*, pages 94–143. PWS Publishing Co., 1996.
11. Ali R Hurson, Angela Maria Muñoz-Avila, Neil Orchowski, Behrooz Shirazi, and Yu Jiao. Power-aware data retrieval protocols for indexed broadcast parallel channels. *Pervasive and Mobile Computing*, 2(1):85–107, 2006.
12. David S Johnson. Approximation algorithms for combinatorial problems. In *Proceedings of the fifth annual ACM symposium on Theory of computing*, pages 38–49. ACM, 1973.
13. Viggo Kann. Maximum bounded 3-dimensional matching is max snp-complete. *Information Processing Letters*, 37(1):27–35, 1991.
14. Bernhard Korte, Jens Vygen, B Korte, and J Vygen. *Combinatorial optimization*, volume 1. Springer, 2002.
15. Zaixin Lu, Yan Shi, Weili Wu, and Bin Fu. Efficient data retrieval scheduling for multi-channel wireless data broadcast. In *INFOCOM, 2012 Proceedings IEEE*, pages 891–899, march 2012.
16. Christos Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 229–234. ACM, 1988.
17. A. Schrijver. *Theory of linear and integer programming*. John Wiley & Sons Inc, 1998.
Appendix

Derandomization

The main idea is to select $S_i$ for $F_i$ greedily and sequentially: $S_1$ is first to select, then $S_2$ and so on. Note that the corresponding LP (3) must be changed accordingly. Assume that the selection of $S_1, \ldots, S_h$ is complete, and the algorithm is currently selecting $S_{h+1}$. Then the LP formula should be:

$$\min \sum_{i=h+1}^{N+n} \sum_{S_{i,j} \in F_i} c(S_{i,j}) \left( \sum_{k: u_k \in S_{i,j}} w(u_k) \cdot x_{i,j} \right)$$

such that

$$\sum_{S_{i,j} \in F_i} x_{i,j} = 1 \quad i = h + 1, \ldots, N$$

$$\sum_{S_{i,j} \in F_i} x_{i,j} \leq 1 \quad i = N + 1, \ldots, N + n$$

$$\sum_{i,j: S_{i,j} \supseteq \{u_k\}} x_{i,j} \geq 1 \quad u_k \notin \bigcup_{i=1}^{h} S_i$$

$$0 \leq x_{i,j} \leq 1 \quad i = 1, \ldots, N + n; S_{i,j} \in F_i$$

(5)

The reason of the change of Inequality (6) is that, if $u_k \in \bigcup_{i=1}^{h} S_i$, then $u_k$ is covered by $\bigcup_{i=1}^{h} S_i$ and needn’t to be covered in later processes. The detailed algorithm is as shown in Algorithm 3.

**Algorithm 3** A randomized algorithm for MSC.

**Input:** $F = \{F_1, \ldots, F_N\}$, where $F_i = \{S_{i,j}\}$ is a collection of subsets of $S = \{u_1, u_2, \ldots, u_n\}$ with $w(u_i)$ for each $u_i$;

**Output:** $C'$ a solution to MSC.

1. $C' := \emptyset$;
2. For $i = 1$ to $N$ do
   (a) For each $S_{i,j} \in F_i$ do
      i. Set $z_j := \infty$;
         /* $z_j$ is the value of the objective function of an optimum solution to LP 5 */
      ii. Set $S_i := S_{i,j}$;
      iii. Solve LP 5 by Karmarkar’s algorithm [17], and assign $z_j$ with the current value of the objective function;
   (b) Select $j'$, such that $z_{j'} \leq z_j$ for every $S_{i,j} \in F_i$;
   (c) $C' := C' \cup \{S_{i,j'}\}$
3. Return $C'$. 

Proof of Proposition 5

Proof. We shall first show that \( f(K) \) is a monotone increasing function when \( K \geq 2 \), and then to show that \( \lim_{K \to +\infty} f(K) = e^{-1} \). For the former, we have the derivative of \( f(K) \):

\[
f'(K) = \left( \frac{K-1}{K} \right) K' = \left( e^{K \left( \ln(K-1) - \ln K \right)} \right)' = e^{K \left( \ln(K-1) - \ln K \right)} \cdot \left( \frac{1}{k-1} + \ln \frac{K-1}{K} \right).
\]

Because for \( K = 2 \), \( \frac{1}{k-1} + \ln \frac{K-1}{K} > 0 \) and its derivative \( \left( \frac{1}{k-1} + \ln \frac{K-1}{K} \right)' = 0 \) both hold, we have \( f'(K) > 0 \) for any \( K \geq 2 \).

Then for the latter, the limitation of \( f(K) \) is:

\[
\lim_{K \to +\infty} f(K) = \lim_{K \to +\infty} \frac{e^{K \left( \ln K - \ln(K+1) \right)}}{\left( \frac{1}{k-1} + \ln \frac{K-1}{K} \right)'} = e^{-1}.
\]

Therefore, even when \( K \) is very large, the ratio in Lemma 4 will still be bounded by \( 1 - \left( \frac{K}{K+1} \right)^K \geq 1 - e^{-1} \approx 0.632 \).