Finding Sparse Solutions for the Index Coding Problem

M. A. R. Chaudhry[1], Z. Asad[1], A. Sprintson[1], and M. Langberg[2]
[1]Department of Electrical & Computer Engineering, Texas A & M University;
[2]Computer Science Division, Open University of Israel
email:{masadch,zasad,spalex}@tamu.edu, mikel@openu.ac.il

***Abstract*—The *Index Coding* problem has recently attracted a significant attention from the research community. In this problem, a server needs to deliver data to a set of wireless clients over the broadcast channel. Each client requires one or more packets, but it might have access to the packets requested by other clients as side information. The goal is to deliver the required data to each client with minimum number of transmissions.**

In this paper, we focus on finding *sparse* solutions to the Index Coding problem. In a sparse solution each transmitted packet is a linear combination of at most two original packets. We focus both on *scalar* and *fractional* versions of the problem. For the scalar case, we present a polynomial time algorithm that achieves an approximation ratio of $2 - \frac{1}{\sqrt{n}}$. For the fractional case, we present a polynomial time algorithm that identifies the optimal solution to the problem. Our simulation studies demonstrate that our algorithms achieve good performance in practical scenarios.

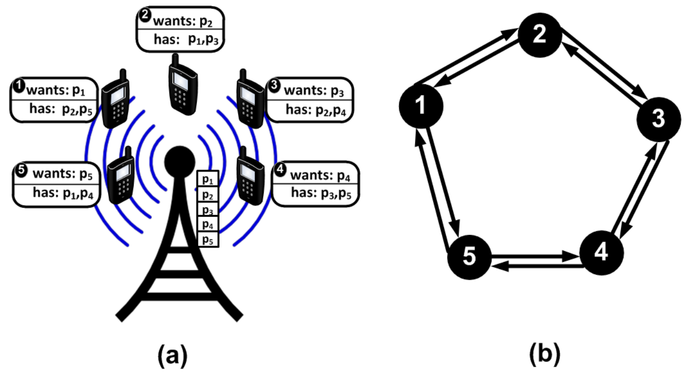

Fig. 1. (a) An instance of the *Index Coding* problem. (b) Corresponding dependency graph.

I. Introduction

The *Index Coding* problem [1], [2] is one of the basic problems in wireless network coding. Recently, it has attracted significant attention from the research community (see e.g., [3]–[6] and references therein).

An instance of the *Index Coding* problem comprises of a server, a set $X = \{c_1, \ldots, c_n\}$ of n wireless clients, and a set $P = \{p_1, \ldots, p_k\}$ of k packets that need to be delivered to the clients. Each client is interested in a certain subset of packets available at the server (*wants* set), and has a (different) subset of packets as side information (*has* set). The server can transmit the packets to clients via a noiseless wireless channel. The goal is to find a transmission scheme that requires the minimum number μ of transmissions to satisfy the requests of all clients.

A simple instance of the Index Coding problem is shown in Figure 1(a), with a server and five clients. The server needs to deliver five packets $p_1, \ldots, p_5$ to five clients, each client wants a unique packet and has access to some side information. It can be verified that all clients can be satisfied by broadcasting three packets, $p_1 + p_2$ and $p_3 + p_4$ and p_5 (all additions are over $GF(2)$). Note that with the traditional approach, all the five packets $p_1, \ldots, p_5$ need to be transmitted.

The research on the *Index Coding* problem can be classified into two major directions. The first direction of research focuses on achievable rate bounds, as well as on the connections between the *Index Coding* problem and the *Network Coding* problem [7]–[9]. The second direction focuses on analyzing the computational complexity of the Index Coding problem as well as developing heuristic approaches to this problem [3]–[6], [10]. In particular, finding the scalar linear solution for the Index Coding problem (in the scalar linear setting) has been shown to be NP-hard. Moreover, finding an approximation solution for the Index Coding problem was also proven to be hard under a widely held complexity assumption [10].

The previous works on the Index Coding problem considered the general setup where the server can encode as many packets as necessary. In this paper, we are focusing on a practically important special case, in which a server can mix at most *two packets* in any single transmission. We refer to this problem as the *Sparse Index Coding (SIC)* problem. With sparse Index Coding, the encoders and decoders can be implemented very efficiently which makes it attractive for practical applications. Furthermore, sparse Index Coding can be implemented over a small field ($GF(2)$), which allows to significantly reduce the size of the packet headers and the associated overhead.

Related work. In [11] we presented the concept of the *Complementary Index Coding (CIC)* problem. In the *CIC* problem, instead of minimizing the number μ of transmissions, the goal is to maximize the number of "saved" transmissions, i.e., $n - \mu$, where n is the number of packets that need to be delivered to the clients. The *CIC* problem seeks to maximize the benefit obtained by employing the coding technique, e.g., for the instance shown in the Figure 1(a), the number of transmissions saved is 2. Note that, if OPT_{IC} is the optimum of the Index Coding problem and OPT_{CIC} is the optimum of the Complementary Index Coding problem, then it holds that $|OPT_{CIC}| = n - |OPT_{IC}|$. This implies that finding the scalar-linear solution for the *CIC* problem is also NP-hard. However, as shown in [11] the *CIC* problem can be successfully approximated in many cases of practical importance as compared to the Index Coding problem problem which has been proven to be hard to approximate. More specifically, [11] presents algorithms with the approximation ratios of $\Omega(\sqrt{n} \cdot \log n \cdot \log\log n)$ and $\Omega(\log n \cdot \log\log n)$, for the scalar and vector linear solutions for the *CIC* problem, respectively. In [12] authors presented a sparse network coding scheme for robust communication in wireless body area networks. In the proposed scheme, each transmitted packet is a combination of at most two original packets. However, [12] assumes a different setting and the proposed techniques are not applicable to the Index Coding problem.

Contributions. In this paper we investigate the *Sparse Index Coding* problem, and consider *scalar* and *vector* linear solutions. In the scalar-linear solution, the packets cannot be split. In a vector-linear solution, each packet can be split into a smaller sub-packets, such that the sub-packets originated from different packets can be combined together. First, we establish a connection between the sparse scalar Index Coding problem and the problem of finding the maximum number of vertex-disjoint cycles (i.e., the cycle packing problem). Thus connection implies that the sparse Index Coding problem is NP-hard. Second, for we present an algorithm that achieves an approximation ratio of $2 - \frac{1}{\sqrt{n}}$ for the scalar linear case, i.e., with our algorithm the number of transmissions is at most $2 - \frac{1}{\sqrt{n}}$ times the optimum. We note that for the sparse Index Coding, the approximation ratio of 2 can be achieved by using the standard approach (which does not include network coding). However, our solution allows to avoid ("save") at least $\frac{1}{\sqrt{n}}$ transmissions compared to the standard solution. Next, we present an algorithm that identifies an optimal vector-linear solution in polynomial time. Finally, we present an experimental study showing the advantage of the proposed algorithms.

II. Model

An instance of the *Index Coding (IC)* problem includes a server s, a set of n wireless clients $X = \{c_1, \ldots, c_n\}$, and a noiseless broadcast channel. The server holds a set of n packets, $P = \{p_1, \ldots, p_n\}$, that need to be transmitted to the clients. We are focusing on the *multiple-unicast* case, i.e., the case in which each packet is required by exactly one client. Each client has a prior knowledge about a subset of packets in P. We denote the subset of P held by client c_i by $H(c_i)$. Without loss of generality, we assume that each client requires exactly one packet. Indeed, a client that requires more than one packet can be represented by multiple clients that have the same side information set, but require different packets. We denote the packet requested by client c_i by w_i.

In a *scalar-linear* solution, each packet is considered to be an element of the Galois field of order q, i.e., $p_i \in GF(q)$. A scalar linear solution includes μ transmissions such that the packet $p^i = \sum_{j=1}^{n} g_i^j \cdot p_j$ transmitted at iteration i, $1 \leq i \leq \mu$ is a linear combination of packets in P, where $g_i = \{g_i^j\} \in GF(q)^n$ is the encoding vector for transmission i. To decode packet w_i, client c_i uses a linear decoding function r_i, such that $w_i = r_i(p^1, \ldots, p^\mu, H(c_i))$. The goal of the *Index Coding (IC)* problem is to find the minimum value of μ such that there exists a set of μ encoding vectors $g_1, g_2, \ldots, g_\mu$ and a set of n decoding functions $r_1, \ldots, r_n$ that allow each client to decode the packet it needs.

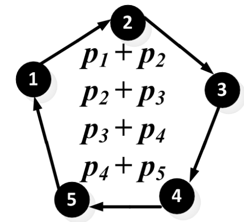

Fig. 2. A *dependency graph* with a cycle and corresponding optimal set of transmissions.

In a *vector-linear* solution, each packet p_i can be subdivided into k smaller size subpackets $p_i^1, \cdots, p_i^k$. Now, each transmitted packet is a linear combination of the subpackets, rather than the original packets. Here, our goal is to find encoding and decoding schemes that minimize the ratio of $\frac{\mu}{k}$, where μ is the number of times a combination of subpackets is transmitted.

In this paper, we focus on the *Sparse Index Coding (SIC)* problem. In this problem each transmitted packet must be a linear combination of *at most two* packets in P. We denote the minimum value of the scalar and vector linear solutions to Problem SIC by OPT^s and OPT^f, respectively.

III. Finding Efficient Scalar-linear solution

In this section, we focus on scalar-linear solutions of the SIC problem. The key idea is to establish a connection between Problem SIC and the problem of finding the maximum number of vertex-disjoint cycles in the corresponding *dependency* graph $G(V, E)$. The dependency graph is defined as follows.

Definition 1 (Dependency Graph $G(V, E)$): Given an instance of the *SIC* problem we define a graph $G(V, E)$ as follows:

- For each client $c_i \in X$ there is a corresponding vertex v_i in V,
- For any two clients, c_i and c_j such that $w_i \in H(c_j)$, there is a directed arc $(v_i, v_j) \in E$.

Figure 1(b) depicts the dependency graph that corresponds to an instance of the *CIC* problem shown in Figure 1(a).

Problem *Vertex Cycle Packing* (VCP) asks for the largest set of vertex-disjoint cycles in the graph $G(V, E)$. We denote the optimal solution to Problem VCP by OPT_{VCP}.

The main idea is to show that for each vertex-disjoint cycle in the dependency graph we can save at least one transmission. To see this, consider the example depicted in Figure 2. In this example, we have a cycle that involves five clients, such that client c_i requires packet p_i. For $i = 2, \ldots, 5$ it holds that the client c_i has the packet required by client c_{i-1}. It is easy to verify that all clients can be satisfied by four transmissions: $p_1 + p_2$, $p_2 + p_3$, $p_3 + p_4$, and $p_4 + p_5$. Indeed, the client c_2 will be satisfied by the transmission $p_1 + p_2$, the client c_3 will be satisfied by transmission $p_2 + p_3$, and so on. The client c_1 will add all the transmissions to obtain $p_1 + p_5$, which will allow it to decode packet p_1.

Our algorithm, referred to as Algorithm *sSIC*, performs the following steps. First, the algorithm constructs the dependency graph $G(V, E)$ for the problem at hand. Next, the *vertex split* graph $G'(V', E')$ of $G(V, E)$ is constructed. The vertex-split graph $G'(V', E')$ is formed from $G(V, E)$ by substituting each vertex $v_i \in V$ by two vertices v'_i and v''_i with an edge (v'_i, v''_i) that connects v'_i and v''_i; and by substituting each edge $(v_i, v_j) \in E$ by an edge (v''_i, v'_j). Finally, we apply the approximation algorithm due to Krivelevich et al. [13] to find an approximate cycle packing in $G'(V', E')$. Next, we identify the set of vertex-disjoint cycles in $G'(V', E')$ that correspond to edge-disjoint cycles in $G(V, E)$. Finally, for each cycle in the dependency graph we identify the set of encoding vectors such that one transmissions is saved per cycle. Note, that Algorithm *sSIC* has a running time of $O(n^3)$.

The formal description of Algorithm *sSIC* is presented in Figure 3.

We proceed to analyze the correctness of Algorithm *sSIC*. In the following two lemmas we prove that $n - OPT^s = OPT_{VCP}$.

Lemma 1: $n - OPT^s \geq OPT_{VCP}$.

Proof: Let $S = \{s_1, s_2, \ldots, s_m\}$ be a packing of vertex-disjoint cycles in $G(V, E)$. Then, we construct an solution to Problem SIC that includes, for each cycle $s_i = \{v_{i_1}, v_{i_2}, \ldots, v_{i_l}, v_{i_1}\}$ packets $p_{i_j} + p_{i_{j+1}}$ for $j = 1, \ldots, l-1$, where l is the size of the cycle. It is easy to verify that the total number of transmitted packets is equal to $n - m$, i.e., for each cycle, one transmission is "saved." Also, it is easy to see that this is a valid solution to Problem SIC. Indeed, each client c_{i_j}, $2 \leq j \leq l$ can recover its required packets w_{i_j} directly from transmitted packet $p_{i_{j-1}} + p_{i_j}$. We note that

$$\sum_{j=1}^{l-1} p_{i_j} + p_{i_{j+1}} = p_{i_l} + p_{i_1}.$$

Thus, client c_{i_1} can also recover the packet it requires. ∎

Next, we show that $OPT_{VCP} \geq n - OPT^s$.

Lemma 2: $n - OPT^s \leq OPT_{VCP}$.

Proof: Let $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_{OPT^s}\}$ be an optimal solution to Problem SIC. Note that each $\gamma_i \in \Gamma$ is a combination of at most two packets in P, these packets are referred to as the *support* of γ_i. First, we define a set $P_1 \subseteq P$ that includes all packets $p_i \in P$ for which it holds that p_i belong to the span of Γ. We define $P_2 = P \setminus P_1$. We say that two packets $p_i \in P_2$ and $p_j \in P_2$ are *connected* if a linear combination of p_i and p_j belongs to the span of Γ. Note that the connectivity is a transitive property, hence P_2 can be divided to equivalence classes $P_2^1, P_2^2, \ldots$, such that each equivalence class includes connected packets. Note that the number of equivalence classes is equal to $n - OPT^s$.

Let $V_2^1, V_2^2, \ldots$ be subsets of vertices of V that correspond to equivalence classes $P_2^1, P_2^2, \ldots$. We show that for each V_2^i it holds that the subgraph of G induced by V_2^i contains at least one cycle. Since the subsets $V_2^1, V_2^2, \ldots$ are disjoint, this will be sufficient to show that $G(V, E)$ contains at least $n - OPT^s$ disjoint cycles.

Let V_2^i be a subset that includes two or more nodes and let G^i be a subgraph of G induced by V_2^i. We show that the in-degree of each node $v_j \in G^i$ is at least one. Indeed, let v_j be a node in V_2^i and let c_j be the client that corresponds to v_j. Let $\hat{\gamma}$ be a vector in span of Γ used by c_j to decode packet w_j in its wants set. It is easy to see that $\hat{\gamma}$ includes at least one packet, say v_l in P_2^i. Then, there exists an edge (v_l, v_j) in G^i and the lemma follows. ∎

For example, consider an instance of the *IC* problem as shown in the Figure 1. The optimal solution to the Problem *SIC* is given by: $\Gamma = \{p1 + p2, p3 + p4, p5\}$. In this example: $P_1 = \{p_5\}$, and $P2 = \{p_1, p_2, p_3, p_4\}$. The corresponding equivalence classes are as follows: $P_2^1 = \{p_1, p_2\}$ and $P_2^2 = \{p_3, p_4\}$, with $V_2^1 = \{v_1, v_2\}$ and $V_2^2 = \{v_3, v_4\}$ respectively. The subgraph corresponding to V_2^1 consists of two arcs (v_1, v_2) and (v_2, v_1), and hence corresponds to a cycle between v_1 and v_2. Similarly the subgraph corresponding to V_2^2 contains a cycle between v_3 and v_4.

Lemma 3: Sparse Index Coding (SIC) problem is an NP-Complete problem. Furthermore, it is quasi-NP-hard to approximate the number of transmissions "saved ", i.e., $n - OPT^s$, within a factor of $O(\log^{1-\varepsilon} n)$ for any constant $\varepsilon > 0$.

Proof: By combining lemmas 1 and 2 we get $n - OPT^s = OPT_{VCP}$. Then, by using the inapproximability result of the *VCP* problem [13], the lemma follows. ∎

We conclude with the following theorem:

Theorem 1: Algorithm *sSIC* finds a scalar-linear solution to the Sparse Index Coding problem with approximation ratio of $2 - \frac{1}{\sqrt{n}}$. The algorithm also allows to "save" at least a factor of $\frac{1}{\sqrt{n}}$ times the optimum saving.

Proof: Let OPT^s be the optimal solution for Problem SIC. By lemmas 1 and 2, the maximum number of vertex-disjoint cycles that can be packed in $G(V, E)$ is $OPT_{VCP} = n - OPT^s$. Then, by using the approximation algorithm due to Krivelevich et al. [13] we can identify at at least $\frac{OPT_{VCP}}{\sqrt{n}}$ cycles. Thus, our algorithm requires at most

$$n - \frac{OPT_{VCP}}{\sqrt{n}} = n - \frac{n - OPT^s}{\sqrt{n}} \tag{1}$$

transmissions. This implies that the algorithm achieves an approximation ratio of

$$\frac{1}{\sqrt{n}} + \frac{n - \sqrt{n}}{OPT^s}. \tag{2}$$

Note that $OPT^s \geq \frac{n}{2}$ since each transmission is a combination of at most two packets. Hence, by Equation (2) the approximation ratio is bounded by $2 - \frac{1}{\sqrt{n}}$.

By Equation (1), the algorithm "saves" at least $\frac{n - OPT^s}{\sqrt{n}}$ transmissions compared the standard solution that does not use coding. Since the optimal solution to Problem SIC saves $n - OPT^s$ transmissions, that algorithm allows to save at least a factor of $\frac{1}{\sqrt{n}}$ times the optimum saving. ∎

IV. Finding Efficient Vector-linear Solution

In this section, we present an algorithm, referred to as Algorithm *vSIC*, that finds an optimum vector-linear solution to Problem SIC. The algorithm exploits the connection between Problem SIC and the problem of finding an optimal fractional solution for the cycle packing problem, defined as follows. Let C be a set that includes all cycles in the graph $G(V, E)$ and let $\psi, C \to R$ be a function that maps each cycle $c \in C$ to a real number. Our goal is to find a function ψ that maximizes $\sum_{c \in C} \psi(c)$ such that for each $v \in V$, it holds that $\sum_{c \in C} \psi(c) \leq 1$. We denote by OPT_{VCP}^f the optimum fractional solution to the vertex-disjoint cycle packing.

The algorithm includes the following steps. Given an instance of Problem *IC*, we first construct the dependency graph

Algorithm sSIC ():

```
From the given instance of the IC problem, construct the dependency graph G(V, E);
From the given dependency graph construct the Vertex Split Graph G'(V', E')
C' = ∅
While there exists a directed cycle in G'(V', E') do:
    Find a cycle c' of minimum length
    Add c' to C'
    Delete all the edges of c' from G'(V', E')
For each cycle c' ∈ C' do:
    Identify the corresponding cycle c in G(V, E);
    Transmit a set of |c| − 1 transmissions that satisfy all clients in c
For each v_i ∈ G(V, E) not included in any cycle c, transmit the packet required by the corresponding client c_i
```

Fig. 3. Algorithm $sSIC$

$G(V,E)$. Then, we apply the algorithm due to Yuster and Nutov [14] to find an optimal vertex-disjoint cycle packing $\psi : C \rightarrow R$ in $G(V,E)$.[1] Then, we find the minimum integer number k for which it holds that $k\psi(c)$ is an integer for any $c \in C$. Such number exists because for each $c \in C$, $\psi(c)$ is a rational number. Next, we divide each packet $p_i \in P$ into k smaller size subpackets $p_i^1, \dots, p_i^k$.

Next, we create a fractional dependency graph $\hat{G}(\hat{V},\hat{E})$. This graph is constructed similarly to the dependency graph for the scalar case, with the difference that the nodes in $\hat{V}$ correspond to subpackets, and not to the original packets. Specifically, graph $\hat{G}(\hat{V},\hat{E})$ is defined as follows:

- For each subpacket p_i^j of a packet $p_i \in P$ there is a corresponding vertex v_i^j in $\hat{V}$
- There is a directed edge from v_i^j and v_l^h if and only if it holds that $p_i \in H(c_l)$, where c_i are c_l are clients requesting packets p_i and p_l, respectively.

Graph $\hat{G}(\hat{V},\hat{E})$ has the following property. For each fractional cycle packing ψ of graph $G(V,E)$ of size α, exists a set of vertex-disjoint cycles $\hat{C}$ in $\hat{G}(\hat{V},\hat{E})$ of size αk. Given a fractional cycle packing ψ in $G(V,E)$, the integer cycle packing $\hat{C}$ in $\hat{G}(\hat{V},\hat{E})$ can be identified through the following procedure. For each cycle $c \in C$ for which it holds that $\psi(c) > 0$ we can identify $k \cdot \psi(c)$ vertex-disjoint cycles in $\hat{G}$ such that for each node $v_i \in C$, each of the corresponding cycles use one of the nodes in $\{v_i^1, \dots, v_i^k\}$. We then remove $k \cdot \psi(c)$ vertex-disjoint cycles from $\hat{G}$ and repeat the procedure for the next cycle in C.

Now, for each cycle $\hat{c} \in \hat{C}$ we generate $|\hat{c}| - 1$ linear combinations of the subpackets $\{p_i^j\}$ that correspond to vertices in $\hat{c}$. Each such cycle will *save* one subpacket, that is for a cycle that includes l vertices in $\hat{G}(\hat{V},\hat{E})$ (that correspond to l clients), we transmit $l-1$ packets that satisfy all l clients. In total, αk subpackets will be saved, i.e., the total number of transmission is equal to $(n-\alpha)k$. This corresponds to saving α original packets. For each $v_i^j \in \hat{V}$ not included in any cycle in $\hat{C}$, transmit the corresponding subpacket p_i^j.

We proceed to establish the correctness of Algorithm *vSIC*.

Lemma 4: $n - OPT^f = OPT^f_{VCP}$.

Proof: First, we show that $n - OPT^f \leq OPT^f_{VCP}$. Consider an optimal vector-linear solution to Problem SIC. Let k be the number of subpackets in each packet with this solution. We note that the vector-linear solution with respect to the original packets is equivalent to the scalar-linear solution with respect to the subpackets. Then, by Lemma 2, it holds that $k(n - OPT^f)$ is less or equal to the maximum size of integer cycle packing in the fractional dependency graph described above. This, in turn implies that $k(n - OPT^f) \leq k(OPT^f_{VCP})$ or $n - OPT^f \leq OPT^f_{VCP}$.

We proceed to show that $n - OPT^f \geq OPT^f_{VCP}$. Consider an optimal fractional cycle packing $\psi(c)$ in the dependency graph (as defined in Definition 1, Section III). Let k be the minimum integer number for which it holds that $k\psi(c)$ is an integer for any $c \in C$. As we discussed above, this implies that there exists a integer cycle packing of size $k \cdot OPT^f_{VCP}$ in the fractional dependency graph. By Lemma 1 this implies that $k(n - OPT^f) \geq k \cdot OPT^f_{VCP}$ or $n - OPT^f \geq OPT^f_{VCP}$ and lemma follows. ■

Theorem 2: Algorithm *vSIC* finds, in polynomial time, an optimal vector-linear solution to Problem SIC.

Proof: By Lemma 4 it holds that $OPT^f = n - OPT^f_{VCP}$. Then, the theorem follows from the fact that the algorithm due to [14] finds an optimal solution to fractional cycle packing problem. ■

V. Performance Evaluation

In this section we evaluate the performance of the proposed scalar-linear solution for the *Sparse Index Coding* problem. More specifically, we evaluate the performance of the Algorithm *sSIC* and compare it with the optimal linear solution to the scalar *Index Coding* problem. Throughout this section the *optimal solution* refers to the SAT-based time-efficient scalar-linear optimal solution over $GF(2)$ for the *Index Coding* problem as presented in [6].

The experimental setup is as follows. We consider n clients, where each client c_i requires packet p_i. The *has* set $H(c_i)$ for each client c_i is chosen randomly. More specifically, first, the cardinality ℓ_i of the *has* set $H(c_i)$ for each client c_i is selected from a uniform random distribution on integers $1, \dots, n-1$. Then we randomly choose ℓ_i packets for $H(c_i)$ out of n packets $\{p_1, \cdots, p_n\}$. Throughout this section each value in the simulation plots represents an average over 100 runs.

Figure 4 shows the comparison of Algorithm *sSIC* and the optimal solution for the average *coding gain*, where the *coding gain* is defined as the ratio between the minimum number of transmissions needed to satisfy all clients without encoding and the minimum number of transmissions required when

[1] The algorithm due to [14] finds an optimal fractional edge-dijoint cycle packing, however, it is possible use this algorithm to find optimal vertex-disjoint cycle packing by applying it to the vertex-split graph (as explained in Section III).

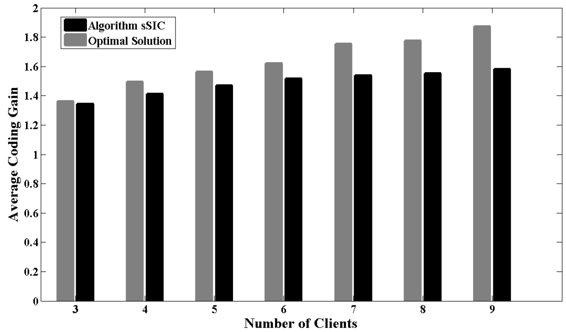

Fig. 4. Average *Coding Gain* versus number of clients for both the *optimal solution* and the solution using algorithm *sSIC*.

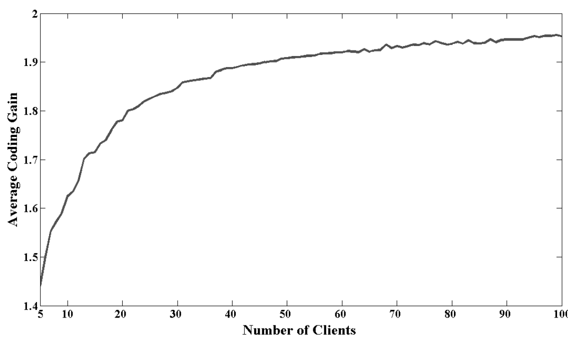

Fig. 5. Average *Coding Gain* versus number of clients for the solution using algorithm *sSIC*.

scalar-linear coding is used. For example for the instance of the *IC* problem shown in Figure 1, the *coding gain* is $\frac{5}{3}$. The results show that on average the ratio of the average *coding gains* given by the optimal solution and Algorithm *sSIC* differ by a factor less than 1.5. Table I shows the comparison of the running time for Algorithm *sSIC* and the time required by the algorithm due to [6] to find an optimal solution. The comparison was performed over a Pentium 4 machine with 2.8 GHz processor. The results show that computing the optimal solution takes considerably more time than the proposed algorithm. Note that while finding the optimal solution requires significant running time even for nine clients, the solution using the algorithm *sSIC* can be efficiently computed even for hundred clients. Figure 5 shows the plot of the average *coding gain* versus number of clients for the solution computed using the algorithm *sSIC*. The plot shows that even for a large number of clients the proposed solution on average saves 49% of the transmissions compared to the solution without encoding.

VI. Conclusion

In this paper, we consider the *Sparse Index Coding* (SIC) problem. In this problem, each transmitted packet is a linear combination of at most two packets over a small field ($GF(2)$). This problem is important in practical settings due to low complexity of encoding and decoding.

We present both *scalar* and *vector* linear solutions for Problem SIC with provable performance guarantees. In particular,

No. of Clients	Optimal Solution	Algorithm *sSIC*
3	0.91	0.0026
4	0.7341	0.0034
5	0.9391	0.0042
6	0.9622	0.0049
7	3.579	0.0061
8	11.93	0.0071
9	82.97	0.0078

TABLE I
Average CPU time (in seconds) required by the optimal solution, and the algorithm *sSIC*.

our algorithm yields a scalar linear solution which has at most $2 - \frac{1}{\sqrt{n}}$ more transmissions than the optimal. For the vector-linear case, we present an algorithm that yields an optimal solution. In addition, we show that finding an optimal solution for the scalar-linear case is an NP-complete problem. We also present an extensive simulation study that demonstrate the advantages of the proposed solution in practical settings.

As a future work, we would like to focus on the multiple multicast case in which a packet can be requested by multiple clients. We would also like to address the practical setting of noisy broadcast channel.

References

[1] Z. Bar-Yossef, Y. Birk, T. S. Jayram, and T. Kol. Index Coding with Side Information. *In Proceedings of 47th Annual IEEE Symposium on Foundations of Computer Science(FOCS)*, pages 197–206, 2006.

[2] Y. Birk and T. Kol. Informed-Source Coding-on-Demand (ISCOD) over Broadcast Channels. In *Proceedings of INFOCOM'98*, 1998.

[3] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft. XORs in the Air: Practical Wireless Network Coding. In *Proceedings of SIGCOMM '06*, pages 243–254, 2006.

[4] B. Hassanabadi, L. Zhang, and S. Valaee. Index coded repetition-based MAC in vehicular ad-hoc networks. In *Proceedings of the 6th IEEE Conference on Consumer Communications and Networking Conference*, pages 1180–1185. Institute of Electrical and Electronics Engineers Inc., The, 2009.

[5] S. Sorour and S. Valaee. Adaptive network coded retransmission scheme for wireless multicast. In *Proceedings of the 2009 IEEE international Symposium on Information Theory*, pages 2577–2581. Institute of Electrical and Electronics Engineers Inc., The, 2009.

[6] M.A.R. Chaudhry and A. Sprintson. Efficient algorithms for index coding. In *IEEE Conference on Computer Communications (INFOCOM) Workshops, 2008.*, pages 1–4, 2008.

[7] Anna Blasiak, Robert D. Kleinberg, and Eyal Lubetzky. Index coding via linear programming. *CoRR*, abs/1004.1379, 2010.

[8] N. Alon, E. Lubetzky, U. Stav, A. Weinstein, and A. Hassidim. Broadcasting with side information. In *IEEE 49th Annual IEEE Symposium on Foundations of Computer Science, 2008. FOCS'08*, pages 823–832, 2008.

[9] S. El Rouayheb, A. Sprintson, and C. Georghiades. On the index coding problem and its relation to network coding and matroid theory. *Information Theory, IEEE Transactions on*, 56(7):3187 –3195, july 2010.

[10] M. Langberg and A. Sprintson. On the hardness of approximating the network coding capacity. In *IEEE International Symposium on Information Theory(ISIT), 2008*, pages 315–319, 2008.

[11] M.A.R. Chaudhry, Z. Asad, A. Sprintson, and M. Langberg. On the Complementary Index Coding Problem. In *IEEE International Symposium on Information Theory(ISIT), 2011*.

[12] Eimear Byrne, Akiko Manada, Stevan Jovica Marinkovic, and Emanuel M. Popovici. A graph theoretical approach for network coding in wireless body area networks. *CoRR*, abs/1102.3603, 2011.

[13] M. Krivelevich, Z. Nutov, M.R. Salavatipour, J.V. Yuster, and R. Yuster. Approximation algorithms and hardness results for cycle packing problems. *ACM Transactions on Algorithms (TALG)*, 3(4):48, 2007.

[14] R. Yuster and Z. Nutov. Packing directed cycles efficiently. In *Proceedings of the 29th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, 2004.