Efficient Algorithms for Index Coding

Mohammad Asad R. Chaudhry and Alex Sprintson
Texas A&M University, College Station, Texas
email:{masadch,spalex}@ece.tamu.edu

***Abstract*— The Index Coding problem has attracted a considerable amount of attention in recent years. The problem is motivated by several applications in wireless networking and distributed computing, including wireless architectures that utilize network coding and opportunistic listening. In this paper, we propose efficient exact and heuristic solutions for this problem. Our numerical study shows that exact solutions can be efficiently obtained for small instances of the problem, while heuristic solutions with low computation time can achieve near-optimal performance for large instances.**

I. Introduction

The Index Coding problem [1], [2] has recently attracted a significant interest from the research community [3]. An instance of the Index Coding problem includes a server, a set of m wireless clients $C = \{c_1, \ldots, c_m\}$, 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 in C. A client $c_i \in C$ is represented by a pair $(W(c_i), H(c_i))$, where $W(c_i) \subseteq P$ is the set of packets required by c_i, and $H(c_i) \subseteq P$ is the set of packets available to c_i. We refer to $H(c_i)$ as the "has" set and $W(c_i)$ as the "wants" set of c_i. The server can broadcast the packets in P or encoding thereof. Our objective is to identify the encoding scheme that satisfies the demands of all clients with the minimum possible number of transmissions. Without loss of generality, we assume that the "wants" set of each client is of cardinality one. Indeed, if it is not the case, each client c_i whose "wants" set contains more than one packet can be substituted by multiple clients whose "wants" sets contain only one packet and whose "has" sets are equal to $H(c_i)$. We define the *coding gain* as the ratio between the minimum number of transmissions needed to satisfy all clients without encoding to the minimum number of transmissions required when encoding is used.

The Index Coding problem is motivated by various applications in wireless networking and distributed computing. For example, efficient index codes are instrumental for the wireless network architectures that utilize the network coding and opportunistic listening techniques [4], [5]. In addition, the Index Coding problem has several applications in satellite communication networks where the clients have limited storage and maintain part of the received information [2].

Figure 1 presents an instance of the Index Coding problem. The central node, referred to as a server, needs to deliver four packets $p_1, \ldots, p_4$ to four clients $c_1, \ldots, c_4$; packet p_i needs to be received by client c_i. Each client c_i has access to some of the packets overheard from prior transmissions. These packets are included in the client's "has" set. It is easy to verify that all clients can be satisfied by broadcasting two packets $p_1 + p_2 + p_3$ and $p_1 + p_4$ (all additions are over $GF(2)$). Since with the traditional approach all packets $p_1, \ldots, p_4$ need to be transmitted, the encoding technique reduces the number of transmissions by 50%, resulting in a coding gain of 2.

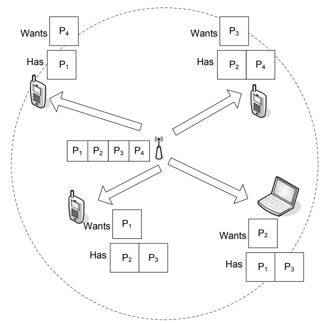

Fig. 1. An instance of the Index Coding problem.

In this paper we present our results in the context of wireless data transmission. However, the considered problem is very general and can arise in many other practical settings. For example, consider a content distribution network that needs to deliver several large files (such as video clips) to different clients. In this setting, if some of the files are already available to some clients then the distribution can be efficiently implemented by multicasting a (small) set of linear combinations of the original files. For example, Figure 2 depicts a video server that has three movies A, B, and C, and three clients that request one of the movies, while two other movies are already available to them. With the traditional approach, the server needs to transmit each movie separately to the client that has requested this movie. With the encoding approach the server only needs to multicast one file $A+B+C$, reducing the total amount of the consumed network resources and alleviating the congestion at the server.

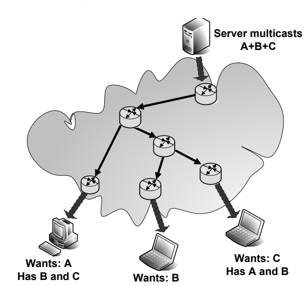

Fig. 2. A content distribution network.

This work was supported in part by NSF grant ECCS-0725823.

In this paper we present exact and approximate solutions to the Index Coding problem. Our exact solution uses a reduction from the Index Coding problem to the boolean satisfiability problem (SAT) with a small number of variables. This reduction allows us to take advantage of the wealth of heuristics developed for the SAT problem over the last decade. Our heuristic solution is based on graph coloring and color saving algorithms. We verify the performance of our solutions through extensive simulations.

II. Finding an optimal solution through SAT solver

Our exact solution is based on a reduction to the SAT problem which, in turn, can be efficiently solved by available SAT solvers such as *Chaff* [6] or *Minisat* [7]. In this section we assume that all the operations are performed over $GF(2)$.

In what follows we show how to check whether it is possible to satisfy all clients through k transmissions. The minimum value of k can be efficiently identified through the binary search algorithm. We need to check whether there exist k encoding vectors $g_1, \ldots, g_k$ of size n, and m decoding vectors $q_1, \ldots, q_m$ of size k that allow each client to decode the packet in its "wants" set. Here, g_j, $0 \le j \le k$ is the encoding vector used for transmission j, i.e., the packet x_j transmitted at round j is equal to $x_j = \sum_{t=1}^{n} g_j^t \cdot p_t$. Each client $c_i \in C$ uses a corresponding vector q_i to decode the packet in $W(c_i)$. In particular, it computes the following linear combination of the original packets:

$$\sum_{j=1}^{k} q_i^j \cdot x_j = \sum_{j=1}^{k} q_i^j \cdot \sum_{t=1}^{n} g_j^t \cdot p_t = \\ = \sum_{t=1}^{n} p_t \sum_{j=1}^{k} g_j^t \cdot q_i^j = \sum_{t=1}^{n} r_i^t p_t,$$

where $r_i^t = \sum_{j=1}^{k} g_j^t \cdot q_i^j$ defines the linear combination of the original packets used by client c_i to decode the packet in $W(c_i)$. Note that client c_i will be able to decode the packet in $W(c_i)$ only if the following two conditions hold:

1) $r_i^t = 1$ for the packet p_t in the "wants" list $W(c_i)$ of c_i;
2) $r_i^t = 0$ for the every packet p_t that does not belong to either "has" list $H(c_i)$ of c_i or "wants" list $W(c_i)$ of c_i, i.e., $p_t \notin \{H(c_i) \cup W(c_i)\}$.

The two above conditions result in the following constraints on $\{g_i\}$ and $\{q_i\}$:

$$\textstyle\sum_{j=1}^{k} g_j^t \cdot q_i^j \equiv 1 \quad \forall c_i \in C \text{ and } p_t \in W(c_i);$$

$$\sum_{j=1}^{k} g_j^t \cdot q_i^j \equiv 0 \ \ \forall c_i \in C \text{ and } p_t \notin \{H(c_i) \cup W(c_i)\}.$$

This, in turn, can be efficiently transformed into a Boolean problem by substituting the summation and multiplication over $GF(2)$ by the AND ($\wedge$) and XOR ($\oplus$) operations, respectively, over the boolean variables.

$$\oplus_{j=1}^{k} \left(g_j^t \wedge q_i^j\right) \equiv 1 \quad \forall c_i \in C \text{ and } p_t \in W(c_i); \tag{1}$$

$$\oplus_{j=1}^{k} \left(g_j^t \wedge q_i^j\right) \equiv 0 \quad \forall c_i \in C \text{ and } p_t \notin \{H(c_i) \cup W(c_i)\}. \tag{2}$$

While formulas 1 and 2 can be easily transformed into the conjunctive normal form (CNF), a straightforward transformation may result in an exponential number of variables. Accordingly, in order to perform the transformation in an efficient manner, we use the Tstein transformation [8]. Such a transformation guarantees that the size of the CNF representation is linear in the size of equations (1) and (2).

Experimental Results: We have implemented the SAT approach and tested it on several random instances of the Index Coding problem. Specifically, given an instance of the original problem we transformed it to the CNF form and then invoked the *Minisat* [7] solver on the resulting CNF formula. Our results show that it is possible to efficiently obtain an optimal solution for instances that include up to 12 clients. For 10 clients we were able to achieve the coding gain as high as 2.4. Figure 3 shows the average coding gain as a function of the number of clients using the SAT-based solution.

III. Heuristic Approaches

The previous works [1], [9] show that it is NP-hard to find an optimal solution for the Index Coding problem. Moreover, it was shown in [10] that finding an approximate solution for this problem is also NP-hard. Accordingly, in this paper we present several heuristic approaches and compare their performance through simulations.

A. Reduction to Graph Coloring

The Index Coding problem is related to the graph coloring problem of an undirected graph $G(V, E)$. In graph coloring, we need to assign a color to each vertex $v \in V$ such that for any edge $(v, u) \in E$, the vertices v and u are assigned different colors, and the total number of colors is minimized. In this section we show a heuristic approach that reduces an instance I of the Index Coding problem to an instance I' of the graph coloring problem.

Specifically, consider an instance I of the Index Coding problem, in which the "wants" set of each client is of cardinality one. Then, we construct an instance $G(V, E)$ to graph coloring problem as follows:

- For each client $c_i \in C$ there is a corresponding vertex v_{c_i} in V;
- Each two vertices v_{c_i} and v_{c_j} are connected by an edge if one of the following holds:
  - Clients c_i and c_j have identical "wants" sets;
  - $W(c_i) \subseteq H(c_j)$ and $W(c_j) \subseteq H(c_i)$.

Let $\hat{V} \subseteq V$ be a clique in $G(V, E)$, i.e., each two vertices of $\hat{V}$ are connected by an edge in G. Note that all clients that correspond to nodes in $\hat{V}$ can be satisfied by one transmission, which includes a linear combination of all packets in their "wants" sets. Thus, we can minimize the number of transmissions by solving the *clique partition* problem [11]. In this problem, we need to partition set V into disjoint subsets

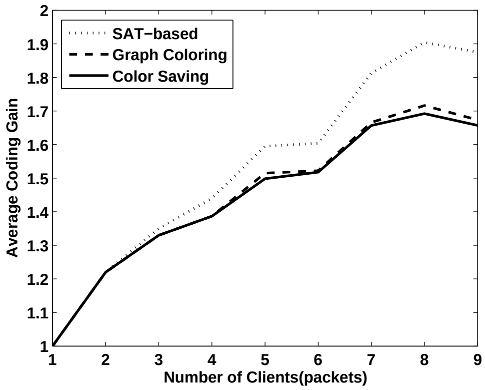

Fig. 3. Comparison of average coding gain between SAT-based, graph coloring and, color saving techniques.

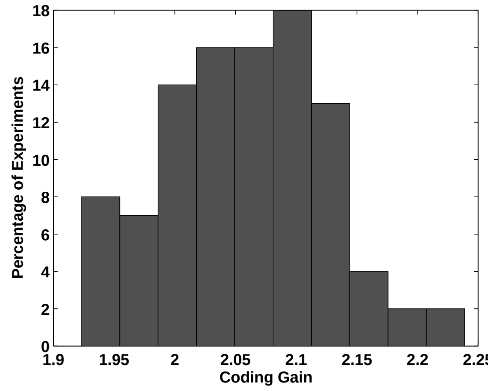

Fig. 4. Histogram of the coding gain values for 150 clients with 100 experiments using sparsest set clustering.

$V_1, V_2, \ldots, V_k$, such that for $1 \leq i \leq k$ the subgraph of G induced by nodes in V_i is a complete graph. Note that the clique partitioning problem for graph $G(V,E)$ is equivalent to the graph coloring problem of the complimentary graph $\hat{G}(V,\hat{E})$ of $G(V,E)$ (the complimentary graph $\hat{G}(V,\hat{E})$ of $G(V,E)$ contains all edges in $V \times V \setminus E$). The graph coloring problem is a well-studied problem with a wealth of efficient heuristic solutions available in the literature.

Our simulation results, depicted in Figure 3 and Table I, show that this technique can be used for a larger number of clients compared to the optimal SAT-based solution although it is suboptimal in terms of the coding gains.

B. Sparsest Set Clustering

In order to extend the number of clients for which the solution can be used, we use the "divide-and-conquer" approach. With this approach, the clients are divided into different groups, the problem is then solved for each group separately. The solution for the original problem is then combined from the solution of the subproblems.

The important decision in such an algorithm is to partition the clients into groups, in a way that minimizes the overall number of transmissions. We note that it is desirable to find a partition in which the clients that belong to different groups have in common as few packets in their "has" and "wants" sets as possible. For this purpose we construct an auxiliary directed graph $G(V,E)$ as follows:

- For each client $c_i \in C$ there is a corresponding vertex $v_{c_i} \in V$;
- Each two vertices v_{c_i} and v_{c_j} are connected by a directed edge (v_{c_i}, v_{c_j}) if one of the following holds:
  - Clients c_i and c_j have identical "wants" sets;
  - $W(c_i) \subseteq H(c_j)$.

Then, our goal is to find a partition of $G(V,E)$ into two clusters of (almost) equal size such that the total number of edges that connect different clusters is minimized. To that end, we use the following greedy heuristic:

1) Partition the set V into two subsets V_1 and V_2 of (almost) equal size (in an arbitrary way);
2) If there exists a node $v \in V_1$ that has more neighbors in V_2 than in V_1, move v to V_2;
3) If there exists a node $v \in V_2$ that has more neighbors in V_1 than in V_2, move v to V_1.

This procedure continues until we have nodes in V_1 or V_2 that have more neighbors in the opposite cluster than the number of neighbors in their current cluster or the cluster of desired size is obtained. The resulting two subgraphs are recursively partitioned through the same procedure until all the size of each cluster is less than or equal to the desired value.

Figure 4 shows a histograms of coding advantage for 100 experiments. In each experiment we generated 150 clients with random "has" and "wants" sets. Then we divided the clients to subsets of at most seven clients each. For each subset, the optimal number of transmissions was found through SAT-based approach described in Section II.

C. Using Color Saving Heuristic

The last heuristic we used is based on the approximation algorithm for maximizing the number of unused colors described in [12], where the number of unused colors is the difference between the number of vertices in the graph and the number of colors used for the coloring. The algorithm in [12] proposes an approximation solution for this problem which is guaranteed to be within a factor of $\frac{2}{3}$ of the optimal solution.

The algorithm performs the following procedure:

1) Construct an undirected graph $G(V,E)$ as described in Section III-A;
2) **While** there exists a clique of size 3 in $G(V,E)$ **do**
   a) Find a clique $\{v_i, v_j, v_k\}$ of size 3 in $G(V,E)$;
   b) Create a packet that satisfies all clients in $\{v_i, v_j, v_k\}$;
   c) $V := V \setminus \{v_i, v_j, v_k\}$;
3) Compute a maximum matching of $G(V,E)$;
4) For each pair $\{v_i, v_j\}$ in the matching create a packet that satisfies all clients in $\{v_i, v_j\}$;
5) Create a new packet for each one of the remaining vertices of V.

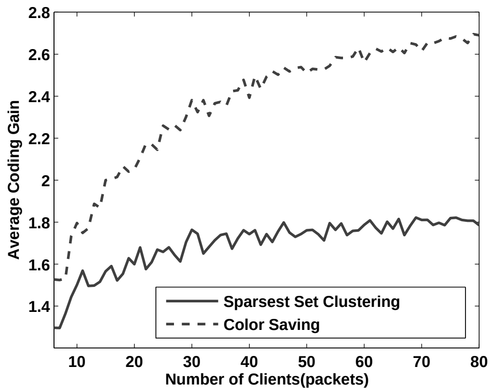

Fig. 5. Comparison of average coding gain between the sparsest set clustering and color caving techniques.

Our results show that this algorithm is very fast, can work with a very large number of clients and has a very good performance in terms of minimizing the overall number of transmissions. Figure 5 depicts the comparison of the color saving and sparsest set clustering techniques and shows that the color saving technique clearly outperforms sparsest set clustering, though for very large number of clients sparsest set clustering is the only option giving results in a reasonable amount of time.

IV. Numerical results

Figure 3 depicts the comparison of the average coding gain for SAT-based, graph coloring and color saving solutions to the Index Coding problem. It shows that when the number of clients is very small all techniques yield almost same average coding gain as there are very few coding opportunities. However, if the number of clients is large, then the SAT-based solution (which is the optimal method) yields more gain as compared to the other techniques. The graph coloring and color saving techniques give almost the same coding gain. We also observed that average coding gain increases with the number of clients.

SAT-based solution can identify the solution efficiently for instances with up to nine clients. For instances with a larger number of clients, graph coloring is a more efficient technique. We were able to solve problem instances for up to 25 clients using graph coloring within a few minutes on a computer with 3.0 GHz Pentium IV processor. Color saving turned out to be the fastest technique up to 80 clients as we were able to solve problem instances with up to 80 clients within a few seconds. Table I shows running time comparison of SAT-based, graph coloring and color saving solutions. While color saving is very fast up to 80 clients, for larger instances the sparsest set clustering turns out to be the fastest technique. Table II shows the comparison of the running time of the sparsest set clustering (with cluster size of 5) and color saving approaches.

V. Conclusion

The paper presents efficient solutions to the Index Coding problem. Our algorithms minimize the number of transmissions necessary for satisfying the requests of all clients. We presented an optimal solution using a SAT solver and several efficient heuristic solutions using the graph coloring, sparsest set clustering and color saving techniques and studied their performance through extensive simulations.

No. of Clients	SAT-based	Graph Coloring	Color Saving
2	0.63	0.03	0.00506
3	0.91	0.06	0.07654
4	0.7341	0.06942	0.01524
5	0.9391	0.07914	0.01492
6	0.9622	0.1043	0.0454
7	3.579	0.1618	0.1285
8	11.93	0.2637	0.09848
9	82.97	0.7654	0.05246

TABLE I

CPU time (in seconds) required by SAT-based, graph coloring and color saving techniques.

No. of Clients	Sparsest Set Clustering	Color Saving
6	0.5636	0.03013
10	0.7807	0.0552
20	1.587	0.14
40	3.28	0.514
80	6.764	5.829
100	9.375	15.719
160	15.656	143.188

TABLE II

CPU time (in seconds) required for sparsest set clustering and color saving techniques.

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*, 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] E. Lubetzky and U. Stav. Non-linear Index Coding Outperforming the Linear Optimum. *In Proceedings of 48th Annual IEEE Symposium on Foundations of Computer Science*, pages 161–168, 2007.

[4] Y. Wu, J. Padhye, R. Chandra, V. Padmanabhan, and P. A. Chou. The Local Mixing Problem. In *Proceedings of Information Theory and Applications Workshop*, San Diego, California, 2006.

[5] 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.

[6] M. Moskewicz, C. Madigan, Y. Zhao, L. Zhang, and S. Malik. Chaff: Engineering an Efficient SAT Solver. Las Vegas, NV, June 2001.

[7] N. Een and N. Srensson. An Extensible SAT-solver. In *Proceedings of 6th International Conference on Theory and Applications of Satisfiability Testing*, 2003.

[8] G. S. Tseitin. On the Complexity of Derivation in Propositional Calculus. In A. O. Slisenko, editor, *Studies in Constructive Mathematics and Mathematical Logic, Part 2*, pages 115–125. Consultants Bureau, New York, 1970.

[9] S. Y. El Rouayheb, M. A. R. Chaudhry, and A. Sprintson. On the Minimum Number of Transmissions in Single-Hop Wireless Coding Networks. Lake Tahoe, CA, September 2007.

[10] M. Langberg and A. Sprintson. On the Hardness of Approximating the Network Coding Capacity. To appear in the Proceedings of International Symposium on Information Theory (ISIT) 2008.

[11] M.R. Garey and D.S. Johnson. *Computers and Intractability*. Freeman, San Francisco, CA, USA, 1979.

[12] R. Hassin and S. L. Haddad. Maximizing the Number of Unused Colors in the Vertex Coloring Problem. *Information Processing Lett*, 52(2):87–90, 1994.