Index Coding and Network Coding via Rank Minimization

Xiao Huang and Salim El Rouayheb
ECE Department, IIT, Chicago
Emails: xhuang31@hawk.iit.edu, salim@iit.edu

***Abstract*—Index codes reduce the number of bits broadcast by a wireless transmitter to a number of receivers with different demands and with side information. It is known that the problem of finding optimal linear index codes is NP-hard. We investigate the performance of different heuristics based on rank minimization and matrix completion methods for constructing linear index codes over the reals. As a summary of our results, the alternating projections method gives the best results in terms of minimizing the number of broadcast bits and convergence rate and leads to up to 13% savings in communication cost compared to graph coloring algorithms studied in the literature. Moreover, we describe how the proposed methods can be used to construct linear network codes for non-multicast networks.**

I. Introduction

We investigate the performance of different rank minimization methods for constructing linear index codes [2], [3], and therefore linear network codes by the equivalence in [4], [5]. Index codes reduce the number of bits broadcast by a wireless transmitter that wishes to satisfy the different demands of a number of receivers with side information in their caches. Fig. 1 illustrates an index coding example. A wireless transmitter has $n = 4$ packets, or messages, $X_1, \dots, X_4$, and there are $n = 4$ users (receivers) $u_1, \dots, u_4$. User u_i wants packet X_i and has a subset of the packets as side information. The packets in the cache could have been obtained in a number of ways: packets downloaded earlier, overheard packets or packets downloaded during off-peak network hours. Each user reports to the transmitter the indices of its requested and cached packets, hence the nomenclature index coding [6]. Assuming an error-free broadcast channel, the objective is to design a coding scheme at the transmitter, called index code, that satisfies the demands of all the users while minimizing the number of broadcast messages. For instance, the transmitter can always satisfy the demands of all the users by broadcasting all the four packets. However, it can save half of the broadcast rate by transmitting only 2 coded packets, $X_1 + X_2 + X_3$ and $X_1 + X_4$ to the users. Each user can decode its requested packet by using the broadcast packets and its side information. The problem that we focus on here is how to construct linear index codes that minimize the number of broadcast messages.

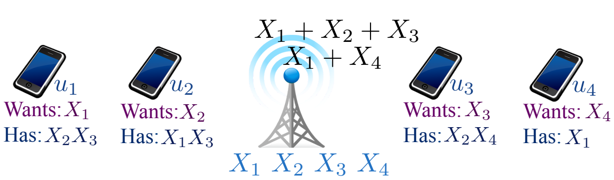


Fig. 1: An index code example.

The full version of this paper can be found in [1].

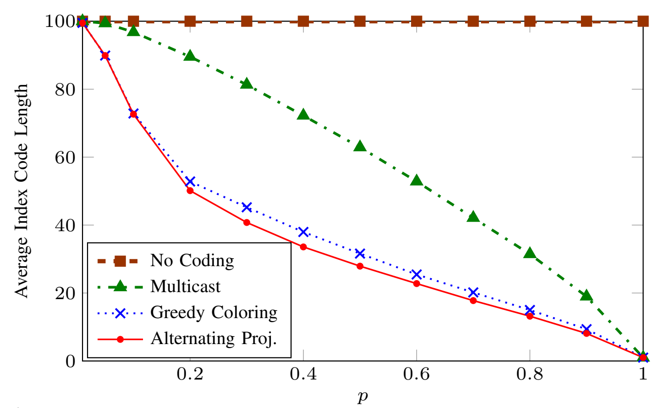


Fig. 2: Comparison of different methods for constructing scalar linear index codes for $n = 100$ users and messages. Each user caches each message independently with probability p (except its requested message).

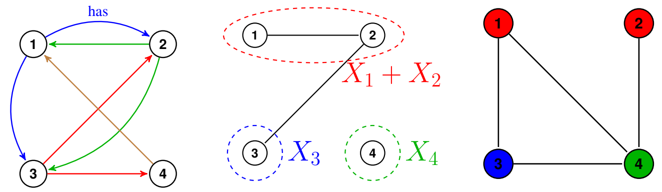
Contribution: Answering the question above turns out to be an NP-hard problem in general [7]–[9]. Motivated by a connection between linear index codes and rank minimization [6], we propose to use rank minimization and matrix completion methods to construct linear index codes. The underlying matrices representing an index coding problem have a special structure that affects the performance of these methods. For instance, the celebrated nuclear norm minimization method [10], [11] does not perform well here. We present our findings on the performance of different other methods, such as alternating projections, directional alternating projections and alternating minimization, through extensive simulation results on random instances of the index coding problem. These methods are performed over the real numbers and give linear index codes over the reals which have applications to topological interference management in wireless networks [12], [13]. As a sample of our results, Fig. 2 compares the performance of index codes obtained by the Alternating Projection (AP) method to other methods studied in the literature. We assumed that packets are cached independently and randomly with probability p. The figure shows the savings in communication cost resulting from using index codes compared to no-coding and linear network coding for multicast (all users decode all messages). Moreover, the AP method leads to up to 13% savings in broadcast messages compared to graph coloring [2], [7].

Over the recent years, several equivalences and connections have been established between index coding and other problems. These connections can be leveraged to apply the rank minimization methods presented here to these equivalent problems. For instance, using the reduction between index coding and network coding devised in [4], [5] to show the equivalence of the two problems, the methods proposed here could be readily applied to construct linear network codes over the reals [14], [15] for general non-multicast networks. Similarly, these methods can be used to construct certain class of locally repairable codes over the reals using the duality between index codes and locally repairable codes established in [16], [17]. Our computer code for constructing linear index codes, network codes and locally repairable codes is available online [18].

Related work: Index coding was introduced by Birk and Kol in [2] as a caching problem in satellite communications. The work of [6] established the connection between linear index codes and the minimum rank of the side information graph representing the problem. The sub-optimality of linear index codes was shown in [19]–[21]. The work of [22] further explored the connection to graph coloring and uncovered surprising properties on index coding on the direct sums of graphs. Linear programming bounds were studied in [23] and connections to local graph coloring and multiple unicast networks were investigated in [24] and [17], respectively. The work in [25] investigated the property of index codes on random graphs. Tools from network information theory [26], [27] and distributed source coding [28] were used to tackle the index coding problem. Related to index coding is the line of work on distributed caching in [29], [30]. Recently, a matrix completion method for constructing linear index codes over finite fields was proposed in [31], and a method for constructing quasi-linear vector network codes over the reals was described in [32].

II. Model

An instance of the index coding problem is defined as follows. A transmitter or server holds a set of n messages or packets, $\mathcal{X} = \{X_1, \dots, X_n\}$, where the X_i's belong to some alphabet. There are n users, $u_1, \dots, u_n$. Let $W_i \subset \mathcal{X}$ ("wants" set) represents the packets requested by u_i, and the set $H_i \subset \mathcal{X}$ ("has" set) represents the packets available to u_i as side information in its cache. WLOG, we can assume that W_i contains only one packet, otherwise the user can be represented by multiple users satisfying this condition. We assume that initially the transmitter does not know which packets are cached at each user, and the users tell the transmitter the indices of the packets they have in an initial stage. Typically, the alphabet size is much larger than the number of packets n, so the overhead in the initial stage is negligible. The transmitter uses an error-free broadcast channel to transmit information to the terminals. The objective is to design a coding scheme at the transmitter, called index code, that satisfies the demands of all the users while minimizing the number of broadcast bits. We will focus on scalar linear index codes in which the messages belong to a certain field ($GF(q)$ or $\mathbb{R}$) and the transmitted messages are linear combinations of these messages.



(a) G_d (b) G (c) $\bar{G}$

Fig. 3: (a) Side information graph G_d of the example in Fig. 1. (b) A clique cover for its undirected subgraph G and the corresponding index code. (c) Graph coloring of the complement graph $\bar{G}$.

III. Connections to Graph Coloring and Rank Minimization

An index coding problem, with n messages and n users can be represented by a directed graph G_d, referred to as side information graph, defined on the vertex set $\{1, 2, ..., n\}$. An edge (i, j) is in the edge set of G_d iff user u_i caches packet X_j. The side information graph G_d representing the instance in Fig. 1 is depicted in Fig. 3(a). Its maximal undirected subgraph G in Fig. 3(b) is obtained from G_d by replacing any two edges in opposite directions by an undirected edge, and removing the remaining directed edges. We will say that G_d is undirected if G_d and G are the same graph. It can be shown that $L_{min} \leq \chi(\bar{G})$ [2], [7], where $\chi(\bar{G})$ is the chromatic number $\chi(\bar{G})$ of the complement graph $\bar{G}$ (See Fig. 3(c)).

It was shown in [6] that finding L_{min} is equivalent to minimizing the rank of a certain matrix M. For instance, this matrix M for the example in Fig. 1 is given by

$$M = \begin{array}{c} \\ u_1 \\ u_2 \\ u_3 \\ u_4 \end{array} \begin{array}{c} \begin{array}{cccc} X_1 & X_2 & X_3 & X_4 \end{array} \\ \begin{pmatrix} 1 & * & * & 0 \\ * & 1 & * & 0 \\ 0 & * & 1 & * \\ * & 0 & 0 & 1 \end{pmatrix} \end{array} .$$

The matrix M is constructed by setting all the diagonal elements to 1's, a star in the $(i, j)^{th}$ position if edge (i, j) exists in G_d, i.e., user u_i caches packet X_j, otherwise the entry is 0. The intuition is that the ith row of M represents the linear coefficients of the coded packet that user u_i will use to decode X_i. Hence, the zero entries enforce that this coded packet does not involve packets that u_i does not have as side information. The packets that u_i has can always be subtracted out of the linear combination. The goal is to choose values for the stars "$*$" from a certain field $\mathbb{F}$ such that the rank of M is minimized. The saving in transmitted messages can be achieved by making the transmitter only broadcast the coded packets that generate the row space of M. It turns out that this formulation of index coding coincides with the minimum rank of a graph, minrk defined by Haemers [33]. Therefore, the optimal rate for a scalar linear index code $L_{min} = \text{minrk}(G) \leq \bar{\chi}(G_d)$ [6].

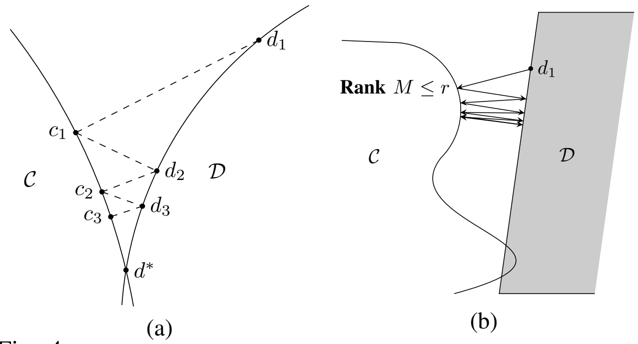


Fig. 4: (a) Alternating Projections (AP) method between two convex sets. (b) AP method for the index coding problem (see Eqs. (1) and (2)).

IV. Index Coding on Random Undirected Graphs

We focus on the case in which the side information graph G_d is undirected. Details on the directed case can be found in [1]. Our approach is to use convex optimizing methods to find L_{min} by minimizing the rank of the matrix M over the reals. The nuclear norm minimization method [10] always output the maximum rank n (instead of min rank) because all the diagonal entries in M are 1s. We found the alternating projections (AP) method [34], [35] to be the most promising.

A. Alternating Projection Method

Given two convex regions $\mathcal{C}$ and $\mathcal{D}$, a sequence of alternating projections between these two regions converges to a point in their intersection as illustrated in Fig. 4(a) [34]–[36]. Therefore, completing the index coding matrix M by choosing values for the "*" such that M has a low rank r can be thought of as finding the intersection of two regions $\mathcal{C}$ and $\mathcal{D}$ in $\mathbb{R}^{n\times n}$, in which

$$\mathcal{C} = \{M \in \mathbb{R}^{n\times n}; \text{rank}(M) \le r\}, \tag{1}$$

and

$$\mathcal{D} = \{M \in \mathbb{R}^{n\times n}; m_{ij} = 0 \text{ if } (i,j) \notin G \ \text{ and } m_{ii} = 1, \quad i = 1,\dots,n\}. \tag{2}$$

Note that $\mathcal{C}$ is not convex and therefore convergence of the AP method is not guaranteed. However, the AP method can give a certificate, which is the completed matrix M, that a certain rank r is achievable. Therefore, we will use the AP method as a heuristic as described in Algorithm APIndexCoding.

The projection of a matrix on the region $\mathcal{C}$ is obtained by singular value decomposition (SVD) [37]. We noticed from our simulations that a considerable improvement in performance and convergence rate, can be obtained by projecting on $\mathcal{C}' \subseteq \mathcal{C}$, the set of positive semi-definite matrices of rank less or equal than r,

$$\mathcal{C}' = \{M \in \mathbb{R}^{n\times n}; M \succeq 0 \text{ and } \text{rank}(M) \le r\}. \tag{3}$$

The projection on $\mathcal{C}'$ is done by eigenvalue decomposition and taking the eigenvectors corresponding to the r largest eigenvalues, as done in Step 8. The Projection on $\mathcal{D}$ is obtained by resetting the fixed entries to their fixed values in M, as done in Step 9 and 10. Step 11 uses the ℓ^2 norm, $\|\cdot\|$, which is equal to the largest singular value of the matrix.

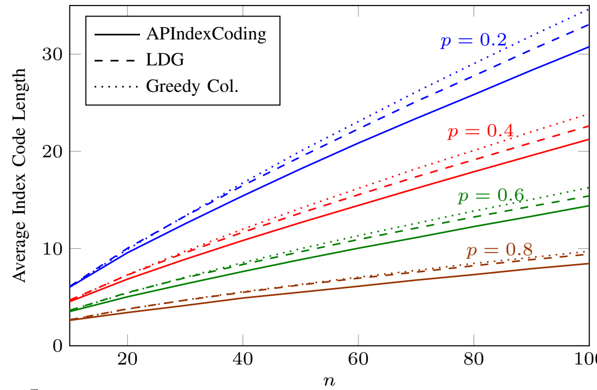


Fig. 5: Average index code length obtained by APIndexCoding, LDG and Greedy Coloring on random undirected graphs $G(n,p)$.

B. Simulation Results

We tested the performance of Algorithm APIndexCoding on randomly generated graphs. We used the Erdos-Renyi model to generate random undirected graphs $G(n,p)$ on n vertices where edges between two vertices are chosen iid with probability p. We compare the performance of Algorithm APIndexCoding to greedy coloring and Least Difference Greedy (LDG) [2], [3]. Also, we have tested the Alternating minimization method [34], [38], [39] but we have discarded it since it converged very slowly for the cases we tested.

Fig. 5 shows the average rank obtained by the APIndexCoding Algorithm for n between 0 and 100 and different values of p. In all our simulations, each data point is obtained by running the algorithms on 1000 graph realizations and $\epsilon = 0.001$ in the stopping criterion. The APIndexCoding

---

Algorithm APIndexCoding: Alternating projections method for index coding

---

Input: Graph G (or G_d)
Output: Completed matrix M^* with low rank r^*
1 Set r_k = greedy coloring number of $\bar{G}$;
2 **while** $\exists M \in \mathcal{C}'$ *such that rank*$M \le r_k$ **do**
3   Randomly pick $M_0 \in \mathcal{C}'$. Set $i = 0$ and $r_k = r_k - 1$;
4   **repeat**
5     $i = i + 1$;
    /* Projection on $\mathcal{C}'$ (or $\mathcal{C}$) */
6   Find the eigenvalue decomposition $M_{i-1} = U\Sigma V^T$, with $\Sigma = \text{diag}(\sigma_1,\dots,\sigma_n)$, $\sigma_1 \ge \dots \ge \sigma_n$;
7   Set $\sigma_l = 0$ if $\sigma_l < 0$, $l = 1,\dots,n$;
8   Compute $M_i = \sum_{j=1}^{r_k} \sigma_j u_j v_j^T$;
    /* Projection on $\mathcal{D}$ */
9     $M_{i+1} = M_i$ Set diagonal entries of M_{i+1} to 1;
10   Change the $(a,b)^{th}$ position in M_{i+1} to 0 if edge (a,b) does not exist in G;
11   **until** $\|M_{i+1} - M_i\| \le \epsilon$;
12 **end**
13 **return** $M^* = M_i$ and $r^* = r_k$.

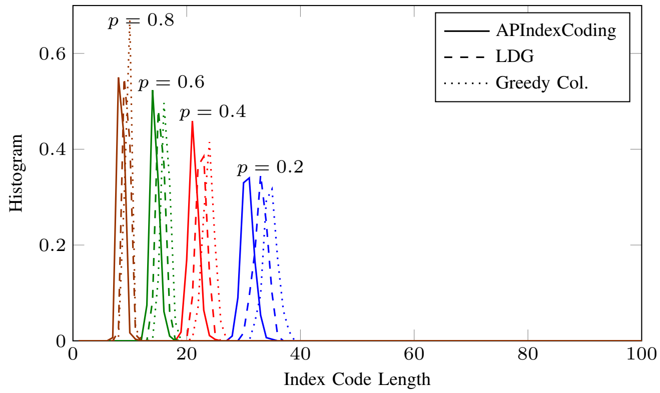


Fig. 6: Histogram of index code length obtained by APIndexCoding, LDG and Greedy Coloring on random undirected graphs when $n = 100$.

Algorithm always outperforms LDG and greedy coloring. For instance, an improvement of 13% over greedy coloring is obtained for $n = 30$ and $p = 0.8$. Fig. 6 shows the histogram of the distribution of the rank by APIndexCoding which suggests a concentration around the mean of ranks returned by APIndexCoding. Note that in Fig. 2 the graph G_d is directed, but the results on alternating projections were obtained by applying APIndexCoding algorithm on the undirected subgraph G. We also tested the APIndexCoding algorithm on all non-homomorphic directed graphs on at most 5 vertices as reported in [40]. APIndexCoding was always able to find the optimal index coding length except for when it is not an integer (28 graphs on $n = 5$ vertices).

C. Convergence Rate and Running Time

We ran the simulations on a DELL XPS i7 - 16GB Memory Desktop using Matlab software. Fig. 7 depicts the average time taken by the APIndexCoding algorithm to converge on a random undirected graph $G(n, p)$. To speed up the converge time, we tested a variant of the AP method, called Directional Alternating Projections (DirAP) [1], [41]. DirAP can lead to considerable savings in time as seen in Fig. 7.

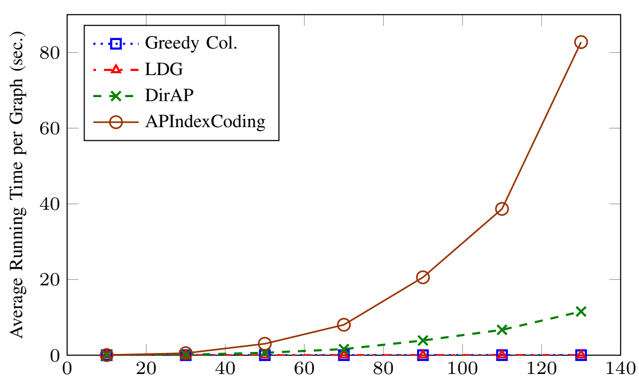


Fig. 7: Running time of APIndexCoding and directional APIndexCoding on random undirected graphs when $p = 0.2$.

D. Decoding Error Analysis

The APIndexCoding algorithm returns the matrix M^* with low rank r^*. However, M^* is not in $\mathcal{C}$ in general, but is very "close" to a matrix in $\mathcal{C}$ (in ℓ^2 norm distance) as dictated by the stopping criteria of the algorithm. This will cause a small decoding error at the users side. The following lemma bounds the decoding error for bounded messages, i.e., $|X_i| \leq X_{max},\ i = 1, \ldots, n$. The proof can be found in [1].

Lemma 1: Let $\mathbf{X} = [X_1, X_2, \ldots, X_n]^T$ be the message vector at the transmitter. Assume that the index code given by matrix M^* is used and let $\hat{\mathbf{X}} = [\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_n]^T$ be the messages decoded by the users. Then,

$$\|\mathbf{X} - \hat{\mathbf{X}}\| \leq \epsilon X_{\max}\sqrt{n}. \tag{4}$$

V. Network Coding via Rank Minimization

The rank minimization heuristics presented here provide a computational tool for constructing linear network codes for non-multicast networks.

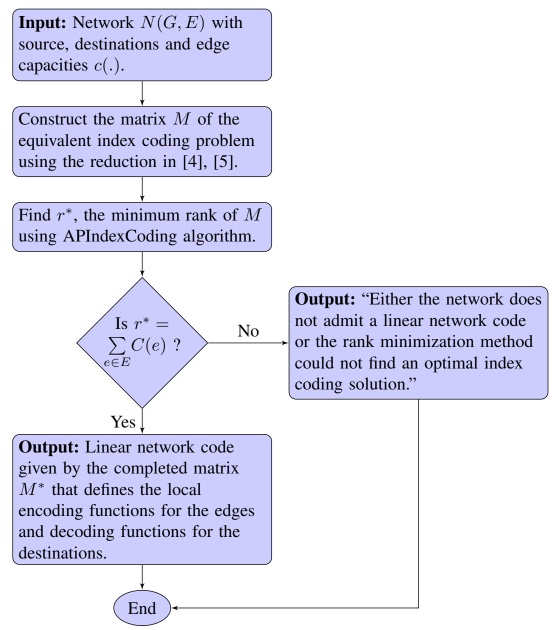


Fig. 8: Linear network coding via rank minimization.

The main idea is to use the efficient reduction in [4], [5] to transform a given network coding problem $\mathcal{NC}$ to an index coding problem $\mathcal{IC}$ and then to apply the APIndexCoding algorithm to $\mathcal{IC}$. Suppose that $\mathcal{NC}$ is defined over a network $N(V, E)$ with vertex set V, edge set E, and each edge $e \in E$ has capacity $c(e)$. The reduction guarantees the following property: $\mathcal{NC}$ has a network code over a certain alphabet that allows all the destinations to decode their messages with zero probability of error if and only if $\mathcal{IC}$ has an index code of length $r^* = \sum_{e \in E} c(e)$ over the same alphabet. This property gives the algorithm illustrated in the flow chart of Fig. 8. This algorithm was implemented in Matlab and can be found and tested on the link in [18].

VI. Conclusion

We have investigated the performance of rank minimization methods for constructing linear index codes over the reals. Our simulation results indicate that the Alternating Projections method and its Directional variant, always outperform (smaller code length) graph coloring algorithms, and they converge much faster than the Alternating Minimization method. Due to the special structure of the underlying matrices representing the index coding problem (all ones diagonal), the well-studied nuclear norm minimization method performs badly here. Our results lead to the following open questions that we plan to address in our future work: (i) Can the proposed methods here be adapted to construct linear index codes over finite fields? (ii) Under what conditions on the index coding matrices, can these methods be given theoretical guaranties to construct optimal linear index codes?

VII. Acknowldgement

The second author would like to thank Prof. Stephen Boyd for suggesting the use of the alternating projections method, Borja Peleato-Inarrea and Carlos Fernandez for discussions on the alternating minimization method and Alex Dimakis for insightful discussions on index coding and graph coloring.

References

[1] X. Huang and S. El Rouayheb, "Index Coding and Network Coding via Rank Minimization." arXiv:1504.06844, 2015.

[2] Y. Birk and T. Kol, "Informed-source coding-on-demand (ISCOD) over broadcast channels," *INFOCOM*, vol. 3, pp. 1257–1264, 1998.

[3] Y. Birk and T. Kol., "Coding on demand by an informed source (ISCOD) for efficient broadcast of different supplemental data to caching clients," *IEEE Transactions on Information Theory*, vol. 52, pp. 2825–2830, June 2006.

[4] S. El Rouayheb, A. Sprintson, and C. Georghiades, "On the index coding problem and its relation to network coding and matroid theory," *IEEE Transactions on Information Theory*, vol. 56, July 2010.

[5] M. Effros, S. El Rouayheb, and M. Langberg, "An Equivalence between Network Coding and Index Coding," *IEEE International Symposium on Information Theory*, pp. 967–971, 2013.

[6] 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*, pp. 197–206, 2006.

[7] S. El Rouayheb, M. A. R. Chaudhry, and A. Sprintson, "On the minimum number of transmissions in single-hop wireless coding networks," in *Information Theory Workshop (ITW)*, 2007.

[8] R. Peeters, "Orthogonal Representations Over Finite Fields and the Chromatic Number of Graphs," *Combinatorica*, vol. 16, no. 3, pp. 417–431, 1996.

[9] M. A. R. Chaudhry, Z. Asad, A. Sprintson, and M. Langberg, "On the complementary index coding problem," in *IEEE International Symposium on Information Theory*, pp. 244–248, IEEE, 2011.

[10] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational mathematics*, vol. 9, no. 6, pp. 717–772, 2009.

[11] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM review*, vol. 52, no. 3, pp. 471–501, 2010.

[12] H. Maleki, V. R. Cadambe, and S. A. Jafar, "Index coding: An interference alignment perspective," in *International Symposium on Information Theory*, 2012.

[13] S. A. Jafar, "Topological interference management through index coding," *Information Theory, IEEE Transactions on*, vol. 60, no. 1, pp. 529–568, 2013.

[14] B. K. Dey, S. Katti, S. Jaggi, D. Katabi, M. Médard, and S. Shintre, ""Real" and "Complex" Network Codes: Promises and Challenges," *Network Coding, Theory and Applications. NetCod 2008. Fourth Workshop on*, pp. 1–6, January 2008.

[15] N. Goela and M. Gastpar, "Reduced-dimension linear transform coding of correlated signals in networks," *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, vol. 60, no. 6, 2012.

[16] A. Mazumdar, "On a Duality Between Recoverable Distributed Storage and Index Coding," *IEEE International Symposium on Information Theory (ISIT)*, 2014.

[17] K. Shanmugam and A. G. Dimakis, "Bounding multiple unicasts through index coding and locally repairable codes," in *IEEE International Symposium on Information Theory*, pp. 296–231, IEEE, 2013.

[18] X. Huang and S. El Rouayheb, "APIndexCoding Matlab Code." http://www.ece.iit.edu/~salim/software.html, 2015.

[19] 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*, pp. 161–168, 2007.

[20] S. El Rouayheb, A. Sprintson, and C. Georghiades, "On the Relation Between the Index Coding and the Network Coding Problems," *In proceedings of IEEE International Symposium on Information Theory (ISIT)*, 2008.

[21] A. Blasiak, R. Kleinberg, and E. Lubetzky, "Lexicographic products and the power of non-linear network coding," *In Proceedings of 52nd Annual IEEE Symposium on Foundations of Computer Science*, pp. 609–618, 2011.

[22] N. Alon, E. Lubetzky, U. Stav, A. Weinstein, and A. Hassidim, "Broadcasting with side information.," *In Proceedings of 49th Annual IEEE Symposium on Foundations of Computer Science*, pp. 823–832, 2008.

[23] A. Blasiak, R. Kleinberg, and E. Lubetzky, "Index coding via linear programming," in *arXiv preprint arXiv:1004.1379*, 2010.

[24] K. Shanmugam, A. G. Dimakis, and M. Langberg, "Local graph coloring and index coding," in *International Symposium on Information Theory*, 2013.

[25] I. Haviv and M. Langberg, "On linear index coding for random graphs," in *IEEE International Symposium on Information Theory*, pp. 2231–2235, IEEE, 2012.

[26] F. Arbabjolfaei, B. Bandemer, Y.-H. Kim, E. Sasoglu, and L. Wang, "On the capacity region for index coding," in *IEEE International Symposium on Information Theory*, pp. 962–966, IEEE, 2013.

[27] F. Arbabjolfaei and Y.-H. a. Kim, "Local time sharing for index coding," in *2014 IEEE International Symposium on Information Theory*, pp. 286–290, IEEE, 2014.

[28] S. Unal and A. B. Wagner, "General index coding with side information: Three decoder case," in *IEEE International Symposium on Information Theory*, pp. 1137–1141, IEEE, 2013.

[29] M. Maddah-Ali and U. Niesen, "Fundamental limits of caching," in *International Symposium on Information Theory*, 2013.

[30] M. A. Maddah-Ali and U. Niesen, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *arXiv preprint arXiv:1301.5848*, 2013.

[31] H. Esfahanizadeh, F. Lahouti, and B. Hassibi, "A matrix completion approach to linear index coding problem," *Information Theory Workshop (ITW), IEEE*, pp. 531–535, 2014.

[32] M. Schwartz and M. Médard, "Quasi-linear network coding," *International Symposium on Network Coding*, 2014.

[33] C. E. Shannon, "The zero error capacity of a noisy channel," *Information Theory, IRE Transactions on*, vol. 2, no. 3, pp. 8–19, 1956.

[34] M. Fazel, H. Hindi, and S. Boyd, "Rank minimization and applications in system theory," in *American Control Conference*, vol. 4, pp. 3273–3278, IEEE, 2004.

[35] S. Boyd and J. Dattorro, "Alternating projections." EE392o, Stanford University, 2003.

[36] L. M. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR Computational Math. and Math. Physics*, vol. 7, no. 3, pp. 200–217, 1967.

[37] G. Y. C. Eckart, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, pp. 211–218, September 1936.

[38] P. Jain, P. Netrapalli, and S. Sanghavi, "Low-rank matrix completion using alternating minimization," in *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pp. 665–674, ACM, 2013.

[39] M. Hardt, "Understanding alternating minimization for matrix completion," *Foundations of Computer Science (FOCS), IEEE 55th Annual Symposium on*, pp. 651–660, 2014.

[40] Y.-H. Kim. http://circuit.ucsd.edu/~yhk/indexcoding.html.

[41] L. E. Ghaoui and S. lulian Niculescu, *Advances in Linear Matrix Inequality Methods in Control*. the Society for Industrial and Applied Mathematics., 1987.