On the Capacity of Information Networks

Micah Adler* Nicholas J.A. Harvey† Kamal Jain‡
Robert Kleinberg§ April Rasala Lehman ¶

Abstract

We consider information networks in the absence of interference and noise, and present an upper bound on the rate at which information can be transmitted using network coding. Our upper bound is based on combining properties of entropy with a strong information inequality derived from the structure of the network.

*The **undirected k-pairs conjecture** states that the information capacity of an undirected network supporting k point-to-point connections is achievable by multicommodity flows. Our techniques prove the conjecture for a non-trivial class of graphs, and also yield the first known proof of a gap between the sparsity of an undirected graph and its capacity. We believe that these techniques may be instrumental in resolving the conjecture completely. We demonstrate the importance of the undirected k-pairs conjecture by connecting it with a long-standing open question in Input/Output (I/O) complexity. We also show that proving the conjecture would provide the strongest known lower bound for computation in the oblivious cell-probe model and give a non-trivial lower bound for two-tape oblivious Turing machines.*

Finally, we conclude by considering the capacity of directed information networks. We construct a family of directed graphs whose capacity is much larger than the rate achievable using only multicommodity flows. The gap that we exhibit is linear in the number of vertices, edges, and commodities of the graph, which is asymptotically optimal.

1 Introduction

Suppose we are given a communication network with capacitated edges, and a set of data streams each of which has a designated source and sink. What is the maximum rate at which each stream can be simultaneously transmitted through the network? (We call this rate the maximum concurrent network coding rate, or simply the ***coding rate***.) This question can be seen as an information-theoretic counterpart to the problem of determining the maximum concurrent multicommodity flow rate (more simply, the ***flow rate***), which is the maximum rate at which physical commodities with the designated sources and sinks can be simultaneously transmitted through the network. While the flow rate is trivially no greater than the coding rate, it is not clear that the two rates are equal because it may be possible to transmit data more efficiently than physical commodities by applying non-trivial encoding operations that combine data from two or more sources. Indeed, there are examples of directed graphs in which this is possible. For instance, Section 7 presents a family of graphs with n vertices and m edges in which the coding rate exceeds the flow rate by a factor of $\Omega(m)$. (Previously, the best known gap had been $\Omega(\sqrt{m})$.)

In contrast to these directed networks in which the coding rate is significantly higher than the flow rate, there is no known instance of an undirected network in which the two rates differ. This has led several authors to formulate the ***undirected k-pairs conjecture*** [25, 26, 14], which asserts that the coding rate is equal to the flow rate in undirected networks. This conjecture is arguably the most important open problem in the field of network coding, which is the study of information networks that allow coding operations on the data streams. Our main goal in this paper is to introduce new techniques which may be instrumental in resolving this conjecture.

Upper bounds on the flow rate in networks are generally derived using one of the following techniques:

1. **Cut based techniques**, which argue that the combined capacity of the edges in a cut must be greater than or equal to the combined transmission rate of the commodities separated by the cut. These techniques lead to an upper bound which we call the ***sparsity*** bound, which is not tight in general.

2. **LP-duality based techniques**, which relate the flow rate to shortest-path metrics defined by assigning lengths to the edges of the graph. Specifically, given such a metric we can define $distance(i)$ to be the distance between the source and sink of commodity i in this metric. The flow rate r satisfies

$$(1.1) \qquad r \leq \frac{\sum_{e \in E(G)} capacity(e) \cdot length(e)}{\sum_i demand(i) \cdot distance(i)},$$

and there is at least one assignment of edge lengths which makes this inequality tight.

In the case of multicommodity flow, the validity of (1.1) is established by an elementary argument which charges each unit of flow for the total length of the edges it

*Department of Computer Science, University of Massachusetts, Amherst. `micah@cs.umass.edu`. Supported by NSF awards EIA-0080119, CCR-0133664, and ITR-0325726.

†MIT CSAIL. `nickh@mit.edu`. Supported by a Natural Sciences and Engineering Research Council of Canada Scholarship.

‡Microsoft Research. `kamalj@microsoft.com`

§MIT CSAIL and Cornell University Department of Computer Science. `rdk@cs.cornell.edu`. Supported by a Fannie and John Hertz Foundation Fellowship.

¶MIT CSAIL. `rasala@mit.edu`.

consumes. It is tempting to postulate that a similar argument establishes (1.1) for network coding scenarios, but it is not clear how to define the charging scheme. (When an edge combines two or more messages using a non-trivial coding function, how can we say what fraction of the edge's capacity is "consumed" by each of the messages?) In fact, Section 6 presents evidence that there is no elementary argument establishing (1.1) for network coding scenarios, by demonstrating that a proof of the undirected k-pairs conjecture would resolve a long-standing open question in Input/Output (I/O) complexity, imply the strongest known oblivious cell-probe lower bound and provide a non-trivial lower bound for two-tape Turing machines.

Since no proof of (1.1) is known for the coding rate r, the following pair of techniques embody the current state of the art for proving upper bounds in network coding theory:

1. **Pigeonhole-principle based techniques**, which argue that if the messages transmitted on edges in an edge set A uniquely determine the data sent on some of the data streams, then the combined capacity of the edges in A is at least as great as the combined transmission rate of these data streams. These are analogous to the cut based techniques for proving upper bounds on the flow rate, and in fact they establish that the sparsity bound is an upper bound on the coding rate in undirected graphs. However, they cannot improve on the sparsity bound. In fact, prior to this work there was *no known instance* of an undirected graph with a provable gap between the sparsity bound and the coding rate.

2. **Information-theoretic techniques**, which use entropy inequalities to derive a linear program whose optimum is an upper bound on the coding rate. However, unlike the case of multicommodity flow, this linear programming approach was only known to be applicable for *directed acyclic graphs (DAG's)*, and it leads to an exponential-sized linear program whose optimum is not even known to match the coding rate.

In this paper, we strengthen and combine both of these techniques to obtain stronger upper bounds on the coding rate in undirected graphs. Specifically, we settle each of the following questions:

1. **What is the most general form of the pigeonhole-principle argument in network coding theory?** In other words, when do the messages transmitted on an edge set A uniquely determine the data sent on some other edge set B? If so, we say that A ***informationally dominates*** B. Although the definition of informational dominance is straightforward, it is not at all straightforward to *combinatorially* characterize this relation, or to determine algorithmically whether A informationally dominates B. We present such a combinatorial characterization and algorithm in Section 3. Informational dominance strikes us as a fundamental notion, and we regard the combinatorial characterization in Section 3 to be one of this paper's main contributions.

2. **Can the information-theoretic linear programming bound be extended from DAG's to undirected graphs, or more generally to directed graphs with cycles?** Section 4 provides such an extension.

3. **Can we improve on the sparsity bound in undirected graphs by combining these two techniques, i.e., by combining combinatorial inequalities derived from informational dominance relations with entropy inequalities derived from information theory?** In Section 5 we improve on the sparsity bound for an infinite family of undirected networks which we call *special bipartite graphs*; in fact, we verify the undirected k-pairs conjecture for special bipartite graphs. Prior to this work, the validity of the undirected k-pairs conjecture was known only for networks in which the sparsity bound equals the flow rate.

We believe that these results constitute significant progress toward resolving the undirected k-pairs conjecture. Next, we present new evidence of the conjecture's importance which also highlights its difficulty: Section 6 shows that proving the conjecture would yield lower bounds which would settle long-standing open questions in I/O complexity and other areas. In particular, we consider the well-studied matrix transposition problem, in which a matrix given in row-major order must be written in column-major order while minimizing I/O operations [11, 3, 10, 1]. Floyd [11], as well as Aggarwal and Vitter [3], have demonstrated lower bounds for this problem under the ***indivisibility*** assumption, where the matrix entries are copied but not otherwise manipulated. Aggarwal and Vitter [3] conjectured that the same lower bound is true without the indivisibility assumption, and described proving this as a challenging and important open problem. We reduce the problem of proving such lower bounds to the problem of proving an upper bound on the capacity of an instance of the k-pairs communication problem. This approach demonstrates that proving the undirected k-pairs conjecture would also resolve Aggarwal and Vitter's long-standing conjecture. Additionally, such a proof would imply the strongest known lower bound in the oblivious cell-probe model and a non-trivial lower bound for two-tape oblivious Turing machines [28].

The preceding discussion concerns the relation between the coding and flow rates in undirected networks. Of equal interest is the relation between these two rates in *directed* networks. In Section 7, we present a family of networks with n vertices and m edges in which the coding rate exceeds the flow rate by a factor of $\Omega(m)$. Previously the best known example, due independently to Li and Li [25] and Harvey et al. [14], established a gap of size $\Omega(n)$ in a network with $m = \Omega(n^2)$ edges. Our stronger example leads to a tight characterization (up to constant factors) of the worst-case gap between the coding and flow rates, as a function of the number of vertices, edges, and commodities.

1.1 Related Work Traditionally, computer scientists have largely considered network capacity in the context of transportation networks rather than information networks. This has lead to a rich theory of network flows [5], multicommodity flows [24], Steiner packings [17, 22], and so on. More recently, the area of network coding has focused on the capacity of information networks where the network nodes can perform coding operations on the information. The seminal work of Ahlswede et al. [4] gives a simple example, shown in Figure 2(b), where the use of coding increases the capacity. Subsequent work focused primarily on ***multicast problems***, where there is a single information source and multiple sinks. The coding rate of a multicast instance is determined by a simple cut condition [4] and the ratio of the coding rate and Steiner packing rate is determined by the integrality gap of a well-known linear program [2]. This ratio can be polylogarithmic [16, 2] in directed graphs and a constant strictly greater than 1 in undirected graphs [27, 2]. Polynomial time algorithms exist to compute optimal network coding solutions to multicast problems [16, 15, 13]. In contrast, the computational complexity of general network coding problems is unknown, although finding a particular type of network coding solution, called a *linear solution*, has been shown to be NP-hard [23].

Recently, much attention has been given to the k-pairs communication problem [21, 25, 14, 26, 27, 18]. This work compared the network coding rate to the corresponding multicommodity flow rate, leading to the undirected k-pairs conjecture [25, 26, 14]. The Okamura-Seymour example was presented in [26] but no formal proof was given that the network coding rate equals 3/4. It has also been shown [20] that no network coding solution in this graph achieves rate exactly 1. However this does not imply a gap between network coding rate and sparsity since we define the network coding rate to be a supremum. A connection between network coding and the matrix transposition problem was also noted by Riis [29].

Sparsity was originally considered in the context of multicommodity flows, and it is well known that there can be a large gap between the flow rate and the sparsity. This gap can be a factor of $\Omega(\log n)$ for undirected graphs and this is tight [24]. For directed graphs, this gap is known to be at least $k-\epsilon$ and $\Omega(\log n)$ [30] and at most $O(\sqrt{n})$ [12]. Closing this gap is a major open problem.

While our work is the first to explicitly define the notion of informational dominance in general graphs and also the first to give necessary and sufficient conditions for informational dominance, the notion was implicit in several prior works which gave only *sufficient* conditions for various useful special cases of informational dominance. Song et al. [31] give a general upper bound on the coding rate for DAGs, in which models of communication are greatly simplified. Their bound uses a local constraint at each vertex that may be regarded as a restricted form of informational dominance. Kramer [19] observes that the d-separation condition from Bayesian networks is sufficient to establish a subset of the informational dominance relationships. Kramer and Savari [20] use this technique to give an upper bound on the capacity of general graphs. Their bound is strictly weaker than our bound from Section 4: for instance, it does not prove Theorem 5.1.

2 Definitions

DEFINITION 2.1. *An instance of the k-pairs communication problem consists of: (1) A graph $G = (V, E)$. (2) A capacity $c(e) \in \mathbb{R}^+$ for each edge e. (3) A set $\mathcal{I}$ of "commodities" of size k. (4) For each $i \in \mathcal{I}$, a source vertex $\sigma(i) \in V$, a sink vertex $\tau(i) \in V$, and a demanded communication rate $d_i \in \mathbb{R}^+$.*

For convenience, we make two modifications to G. First, if G is undirected we replace each undirected edge e with two oppositely-directed edges $\overrightarrow{e}$ and $\overleftarrow{e}$. Second, it is convenient to discuss the flow of information by considering only sets of edges. Therefore, we will treat the sources and sinks as being edges rather than vertices. Formally, we assume that each source has a single out-edge $S(i) = (\sigma(i), s(i))$ and no in-edges, and that each sink has a single in-edge $T(i) = (t(i), \tau(i))$ and no out-edges. We refer to $S(i)$ as the ***source edge*** and $T(i)$ as the ***sink edge*** for commodity i. Let $\mathcal{S} = \{\, S(i) : i \in \mathcal{I} \,\}$ and $\mathcal{T} = \{\, T(i) : i \in \mathcal{I} \,\}$. These edges have infinite capacity. Each source has a ***message*** that is to be communicated to its sink. Let $\mathcal{M}(i)$ be the set of all possible messages for commodity i and let $\mathcal{M} = \prod_i \mathcal{M}(i)$. For a vertex $v \in V$, let $\mathrm{In}(v) \subseteq E$ denote the set of edges whose head is v.

DEFINITION 2.2. *A **network coding solution** for a graph G specifies, for each $e \in E$, an alphabet $\Gamma(e)$ and a function $f_e : \mathcal{M} \to \Gamma(e)$ specifying the symbol transmitted on edge e. These must satisfy two conditions.*

- ***Correctness:*** *Each sink edge carries the message from its corresponding source, i.e.,* $\Gamma(T(i)) = \Gamma(S(i)) = \mathcal{M}(i)$ *and* $f_{T(i)} = f_{S(i)}$ *projects the source message vector onto its* i^{th} *coordinate.*
- ***Causality:*** *Every message transmitted on edge e is computable from information received at its tail vertex at a time prior to the message's transmission.*

The informal specification of the causality criterion in this definition is made precise by requiring the network code to have a *causal computation*, defined as follows.

Definition 2.3. *A* ***causal computation*** *of a network code consists of: (1) A sequence of edges* $\epsilon_1, \ldots, \epsilon_T$ *where each* $e \in E$ *can appear multiple times. (2) A sequence of alphabets* $\Lambda_1, \ldots, \Lambda_T$*. (3) A sequence of coding functions* $\rho_1, \ldots, \rho_T$*. These must satisfy the following conditions: (A) For each function* ρ_t *such that* $\epsilon_t = (u, v)$ *is not a source edge, the value of* ρ_t *is uniquely determined by the values of the functions in the set* $\{\rho_x : x < t \text{ and } \epsilon_x \in \mathrm{In}(u)\}$*. (B) For each edge e, the cartesian product of the alphabets in the set* $\{\Lambda_i : \epsilon_i = e\}$ *is equal to* $\Gamma(e)$*. (C) For each edge e, the set of coding functions* $\{\rho_i : \epsilon_i = e\}$ *together define the coding function* f_e *specified by the network coding solution.*

We use information theory to define the ***rate***, or efficiency, of a solution. Suppose that the message M_i on source edge $S(i)$ is chosen independently and uniformly at random from $\mathcal{M}_i$. Letting $M = (M_1, \ldots, M_k)$, we can define for each edge set $A = (e_1, e_2, \ldots, e_{|A|})$ the ***combined edge function*** $f_A(x) = (f_{e_1}(x), f_{e_2}(x), \ldots, f_{e_{|A|}}(x))$. The random variable $f_A(M)$ is denoted by Y_A and its entropy is denoted $H(A)$. When A is a singleton set $\{e\}$ we use the notations Y_e and $H(e)$ in place of $Y_{\{e\}}$ and $H(\{e\})$.

Definition 2.4. *A network coding solution for a graph G* ***achieves rate*** *r if there exists a constant* $b \geq 0$ *such that (1)* $H(S(i)) \geq r \cdot d_i \cdot b$ *for each commodity i; and (2) for each* $e \in E$*, if G is directed,* $H(e) \leq c(e) \cdot b$*, or if G is undirected,* $H(\overrightarrow{e}) + H(\overleftarrow{e}) \leq c(e) \cdot b$*. For a given network coding problem, the* ***network coding rate****, also known as the* ***capacity****, is defined to be the supremum of the rates of all network coding solutions.*

A key property of entropy that we will frequently use is submodularity: for any sets A and B of random variables, $H(A) + H(B) \geq H(A \cup B) + H(A \cap B)$. The following generalization is also useful, and will be applied in the proof of Theorem 5.1.

Lemma 2.5. (n-way submodularity)
Let $A_1, \ldots, A_n$ *be a collection of sets and let* B_i *be the union of all their i-way intersections. Then* $\sum_{i=1}^{n} H(A_i) \geq \sum_{i=1}^{n} H(B_i)$.

3 Informational Dominance

A key ingredient in our upper bound technique is a relation among edge sets called ***informational dominance***. Suppose that an eavesdropper knows the coding function of each edge in G and has access to all the messages transmitted on an edge set A. If this information *always* allows the eavesdropper to determine the values of the messages transmitted on some other edge set B, we say that A informationally dominates B. In this section, we formally define informational dominance and give a polynomial time algorithm for finding the set of all edges informationally dominated by a given set. This algorithm relies on a graph theoretic characterization of informational dominance.

Definition 3.1. *An edge set A* ***informationally dominates*** *edge set B if for all network coding solutions and k-tuples of messages x and y,* $f_A(x) = f_A(y)$ *implies* $f_B(x) = f_B(y)$*. For an edge set A,*

$$\mathrm{Dom}(A) = \{\, e : A \textit{ informationally dominates } e \,\}.$$

It is important to note that this definition quantifies over all network coding solutions, which must successfully transmit information from all sources to their sinks. The informational dominance relation is clearly a pre-order (i.e., it is reflexive and transitive). However, it is not clear if determining whether a set of edges A informationally dominates a set of edges B is even recursively decidable. The difficulty arises because the informational dominance relation must hold for *all* network coding solutions, of which there are infinitely many. However, we now present a characterization of the informational dominance relation that is entirely based on the structure of the graph.

The following subgraph $G(A, i)$ of G is used in the characterization of informational dominance.

Definition 3.2. *Given a graph G, an edge set A and a commodity i, let* $G(A, i)$ *be the graph obtained from G by the following process: (1) Remove any edge or vertex that does not have a path to* $T(i)$ *in G. (2) Remove all edges of A. (3) Remove any edge or vertex that is not reachable from a source edge in the remaining graph.*

Theorem 3.3. *For* $A \subseteq E$*, the set* $\mathrm{Dom}(A)$ *satisfies the following four conditions. Furthermore, if* $B \subseteq E$ *satisfies these conditions, then* $\mathrm{Dom}(A) \subseteq B$*. (1)* $A \subseteq \mathrm{Dom}(A)$*. (2)* $S(i) \in \mathrm{Dom}(A)$ *iff* $T(i) \in \mathrm{Dom}(A)$*. (3) Every edge in* $E \backslash \mathrm{Dom}(A)$ *is reachable in* $G \backslash \mathrm{Dom}(A)$ *from a source. (4) If* $S(i) \notin \mathrm{Dom}(A)$*, then* $S(i)$ *and* $T(i)$ *are in the same weakly connected component in* $G(\mathrm{Dom}(A), i)$*.*

The proof that $\mathrm{Dom}(A)$ necessarily satisfies these conditions is omitted due to space constraints. However, in Lemma 3.4 and Corollary 3.5 below, we establish the more illuminating half of Theorem 3.3, i.e., that

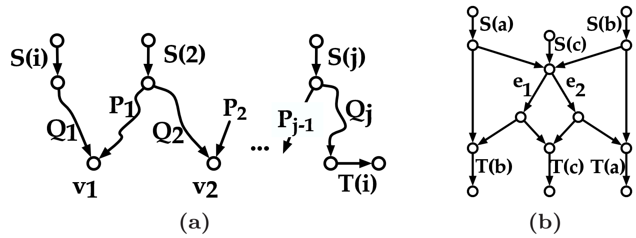

Figure 1: (a) An example of an indirect walk from $S(i)$ to $T(i)$. (b) The split butterfly instance. Each edge has capacity 1 and each commodity has demand 1.

$\mathrm{Dom}(A)$ is the unique minimal superset of A meeting these conditions.

LEMMA 3.4. *If all four conditions of Theorem 3.3 are satisfied by a set of edges B, then B does not informationally dominate any edge in $E \setminus B$.*

Proof. Our first objective is to prove that B does not informationally dominate any source or sink edge in $G \setminus B$. Let $\mathcal{I}_0$ denote the set of all commodities whose source edge is in $G \setminus B$. We will construct an example of a network code in which the message alphabet $\mathcal{M}(i)$ is equal to the field $\mathbb{F}_2$ for $i \in \mathcal{I}_0$, $\mathcal{M}(i) = \{0\}$ for $i \notin \mathcal{I}_0$, and such that there exist two k-tuples of messages $x, y \in \mathcal{M}$ such that $f_B(x) = f_B(y)$ but $x_i \neq y_i$ for all $i \in \mathcal{I}_0$. In fact, x and y will be specified by setting $x_i = 0, y_i = 1$ for $i \in \mathcal{I}_0$, and setting $x_i = y_i = 0$ for $i \notin \mathcal{I}_0$.

Suppose that $S(1)$ is in $G \setminus B$. Conditions 2 and 4 imply that $T(1)$ is also in $G \setminus B$ and that $S(1), T(1)$ belong to the same weakly connected component of $G(B, 1)$. Using the definition of $G(B, 1)$, we can conclude that there exist paths $Q_1, P_1, \ldots, P_{j-1}, Q_j$ such that:

1. The first node in Q_1 is the head of $S(1)$ and the last node in Q_j is the tail of $T(1)$.
2. The last node in Q_ℓ is the same as the last node in P_ℓ for all $\ell < j$.
3. The first node in each path is the head of a source edge in $G(B, 1)$.

We call such a sequence of paths an ***indirect walk*** from $S(1)$ to $T(1)$ in $G(B, 1)$. Without loss of generality, $S(i)$ is the edge whose head is the first node in Q_i. Let v_i denote the last node of path segment Q_i. By the definition of $G(B, 1)$, for $1 \leq i < j$ there exists a path R_i from v_i to the tail of $T(1)$ in G.

We now sketch the portion of our network coding solution that transmits M_1 from $S(1)$ to $T(1)$. All edge alphabets are vector spaces over $\mathbb{F}_2$, and $\oplus$ denotes addition in one of these vector spaces. The code sends M_i from $S(i)$ to v_i on Q_i $(1 \leq i \leq j)$; M_{i+1} from S_{i+1} to v_i on P_i $(1 \leq i < j)$; and $M_i \oplus M_{i+1}$ to $T(1)$ on R_i $(1 \leq i < j)$. Let us now consider the dimension of the alphabets. Initially imagine that all edge alphabets are set to $\{0, 1\}$. If any of the Q, P or R paths intersect then we must accommodate all overlapping transmissions by increasing the alphabets of the edges. For an edge e which appears in p of these paths, we put $\Gamma(e) = \{0, 1\}^p$. Increasing the alphabet size reduces the rate of the solution but this is irrelevant for our proof.

Let us now consider the information received at the tail of $T(1)$. The message $M_i \oplus M_{i+1}$ is received from the path R_i $(1 \leq i < j)$, and M_j is received from the path Q_j. Therefore, the tail of $T(1)$ receives j linearly independent combinations of the j messages $M_1, M_2, \ldots, M_j$. Thus M_1 can be computed from the values of these j symbols and therefore M_1 can be transmitted on edge $T(1)$. Also observe that the paths R_i may intersect B, but the paths P_i, Q_i do not. Thus every bit of information transmitted on edges in B by this portion of the network code is equal to $M_i \oplus M_{i+1}$, which equals 0 when $M = x$ or $M = y$.

Arguing similarly for each of the commodities in the set $\mathcal{I}_0$, we obtain a network coding solution in which each sink edge $T(i)$ $(i \in \mathcal{I}_0)$ can compute the corresponding message M_i, yet every bit sent transmitted on edges in B is equal to $M_i \oplus M_j$ for some $i, j \in \mathcal{I}_0$, and hence $f_B(x) = f_B(y) = (0, 0, \ldots, 0)$. This completes the proof that $S(i)$ and $T(i)$ are not informationally dominated by B for $i \in \mathcal{I}_0$.

Now consider an edge e in $G \setminus B$ that is not a source or sink edge. By condition 3 of Theorem 3.3, e is reachable from some source $S(i)$ in $G \setminus B$. As we have just shown, there are two k-tuples of messages x and y and a network coding solution such that $f_B(x) = f_B(y)$ but $x_j \neq y_j$. We may augment this solution by additionally sending M_j from $S(j)$ to e. Thus edge e is not informationally dominated by B.

COROLLARY 3.5. *Any set of edges B that satisfies the four conditions of Theorem 3.3 contains* $\mathrm{Dom}(A)$.

Proof. If $\exists e \in \mathrm{Dom}(A) \setminus B$ then $e \in \mathrm{Dom}(B)$ since $B \supseteq A$. Thus $e \in \mathrm{Dom}(B) \setminus B$, contradicting Lemma 3.4.

We now consider how to efficiently compute the informational dominance relation. Given A, every set B that is informationally dominated by A is contained in $\mathrm{Dom}(A)$. Thus it is sufficient to construct $\mathrm{Dom}(A)$ efficiently. The approach is to greedily grow a set of edges $\tilde{A}$ that are informationally dominated by A, where initially $\tilde{A} = A$. We repeatedly check the conditions of Theorem 3.3 for $\tilde{A}$ and, if any is violated, we obtain a new edge that can be added to $\tilde{A}$. Letting m denote the number of edges in G, it is clear that Conditions (1)-(3) can be checked in time $O(m)$, and condition (4) can be checked in time $O(km)$ since $G(B, i)$ can be constructed in time $O(m)$ for each commodity i, and the weak components of $G(B, i)$ can be computed in time $O(m)$ as well. Our procedure terminates when $\tilde{A}$

satisfies the four conditions of Theorem 3.3. In total, constructing $\mathrm{Dom}(A)$ requires only $O(k^2 m)$ steps since we may charge the work required to check condition (4) to the commodity added to $\tilde{A}$ during that step.

4 General Upper Bound on Network Capacity

The definition of informational dominance implies that if edge set A informationally dominates edge set B, then each of the random variables $Y_A, Y_{A\cup B}$ uniquely determines the value of the other. Consequently, $H(A) = H(A \cup B)$. Combining this equation with some well-known properties of entropy functions [8], we obtain the following five constraints which must be satisfied by every network coding solution.

- *Polymatroid inequalities:* The entropy function H is a non-negative, non-decreasing, submodular set function. In other words, for all $A, B \subseteq E$, we have $H(A) \geq 0$, $H(A \cup B) \geq H(A)$, $H(A) + H(B) \geq H(A \cap B) + H(A \cup B)$.
- *Informational Dominance:* For $A, B \subseteq E$, if A informationally dominates B, then $H(A) = H(A \cup B)$.
- *Independence of sources:* For any set $S(i_1), \ldots, S(i_\ell)$ of sources, $H(S(i_1), \ldots, S(i_\ell)) = \sum_{j=1}^{\ell} H(S(i_j))$.
- *Correctness:* For every commodity i, the edges $S(i)$ and $T(i)$ transmit the same symbol. Hence, for $A \subseteq E$, $H(A \cup \{S(i)\}) = H(A \cup \{T(i)\})$.
- *Rate:* A solution of rate r exists iff there is a constant b such that $H(S(i)) \geq r d_i b$ for all i, and for every edge e, $H(e) \leq c(e)b$ (for directed graphs), or $H(\overrightarrow{e}) + H(\overleftarrow{e}) \leq c(e)b$ (for undirected graphs).

These constraints can be viewed as a linear program whose optimum value bounds the maximum concurrent network coding rate. An unfortunate aspect of this LP is that its size is exponential in the problem size. A similar LP bound that applies only to directed acyclic graphs was presented by Song, Yeung and Cai [31], who raised the question of generalizing this bound to all graphs as an important open problem. Our work answers this question by defining and characterizing the informational dominance relation in arbitrary graphs.

4.1 Example of the Upper Bound Technique

As a concrete example of applying the five constraints listed above, consider an instance of the k-pairs communication problem, called the ***split butterfly***, which is illustrated in Figure 1(b). We will prove that the network coding rate of the split butterfly instance is $2/3$, which is equal to the maximum concurrent multicommodity flow rate. Note that there is no elementary proof of this bound using a single cut and the pigeonhole principle, since there is no pair of edges which informationally dominates all three sources.

LEMMA 4.1. *The network coding rate of the split butterfly instance is* $2/3$.

Proof. Using Theorem 3.3, or arguing directly from the graph structure, we obtain the following informational dominance relations: $S(b) \in \mathrm{Dom}(\{S(a), e_1\})$, $S(a) \in \mathrm{Dom}(\{S(b), e_2\})$, $S(c) \in \mathrm{Dom}(\{S(a), S(b), e_1, e_2\})$. These observations imply the inequalities (4.2)-(4.4).

$$
\begin{aligned}
&(4.2) \quad H(S(a), e_1) = H(S(a), S(b), e_1)\\
&(4.3) \quad H(S(b), e_2) = H(S(a), S(b), e_2)\\
&(4.4) \quad H(S(a), S(b), e_1, e_2)\\
&\qquad = H(S(a), S(b), S(c), e_1, e_2)\\
&(4.5) \quad H(S(a), e_1) + H(S(b), e_2)\\
&\qquad = H(S(a), S(b), e_1) + H(S(a), S(b), e_2)\\
&(4.6) \quad H(S(a)) + H(S(b)) + H(e_1) + H(e_2)\\
&\qquad \geq H(S(a), S(b), e_1, e_2) + H(S(a), S(b))\\
&(4.7) \quad H(e_1) + H(e_2) \geq H(S(a), S(b), e_1, e_2)\\
&(4.8) \quad H(e_1) + H(e_2) \geq H(S(a), S(b), S(c), e_1, e_2)\\
&(4.9) \quad H(e_1) + H(e_2)\\
&\qquad \geq H(S(a)) + H(S(b)) + H(S(c))\\
&(4.10) \quad (c(e_1) + c(e_2)) \geq (d_a + d_b + d_c) r
\end{aligned}
$$

(4.5) follows by summing (4.2) and (4.3). (4.6) follows from (4.5) by submodularity. (4.7) follows from (4.6) because the sources are independent. (4.8) follows from (4.7) and (4.4). (4.9) follows from (4.8) since the sources are independent. (4.10) follows from (4.9) and Definition 2.4, assuming rate r is achievable. Thus $2/3 \geq r$, since all edges have capacity 1 and all commodities have demand 1.

5 The Capacity of Undirected Graphs

We now consider the relationships between multicommodity flow, network coding, and the sparsity of edge cuts in undirected graphs. Given a k-pairs communication problem on a graph $G = (V, E)$ we define a ***cut*** to be an edge set $A \subseteq E$. If every path from $S(i)$ to $T(i)$ in G intersects A then we say that A ***separates*** commodity i. The sparsity of a cut A and the graph G are respectively defined as

$$\mathscr{S}(A) = \frac{\sum_{e \in A} c(e)}{\sum_{i:\ A \text{ separates } i} d_i} \quad \text{and} \quad \mathscr{S}_G = \min_{A \subseteq E} \mathscr{S}(A).$$

Sparsity illustrates a key difference between undirected and directed graphs: for undirected graphs, sparsity is an upper bound on the network coding rate, whereas for directed graphs it is not. For the undirected case, this claim can be verified by observing that $S(i) \in \mathrm{Dom}(A)$ whenever A is an undirected cut which separates the source and sink for commodity i, and then applying the informational dominance constraint from Section 4. For the directed case, the claim follows from our results in Section 7.

For undirected graphs, we have:

$$\text{flow rate} \;\leq\; \text{coding rate} \;\leq\; \text{sparsity}. \tag{5.11}$$

It is known that for some graphs (e.g., constant degree expanders [24]), the sparsity can exceed the maximum multicommodity flow rate by a factor as large as $\Omega(\log n)$. Hence, for some graphs, at least one of the inequalities in Equation (5.11) is strict. The undirected k-pairs conjecture asserts that the first inequality is *never* strict. In this section we make progress on this conjecture by proving that the second inequality is strict in some instances, e.g., the Okamura-Seymour example illustrated in Figure 2(a). An extension of this proof verifies the undirected k-pairs conjecture for an infinite class of graphs which contains the Okamura-Seymour example.

Let $G = (V \cup W, E)$ be a bipartite graph. Consider an instance of the k-pairs communication problem in G where, for every commodity, the source and sink both belong to V or both belong to W. Let $E(W, V)$ denote the set of directed edges from W to V in G. Let $\mathcal{S}(V)$ denote the set of sources in V, and $\mathcal{S}(W)$ the set of sources in W. For a vertex v, let $\text{In}(v)$ denote the set of incoming edges of v in G, and let $\mathcal{T}(v)$ denote the set of sink edges in $\mathcal{T}$ whose tail is v. Note that $\mathcal{S}(v) \subseteq \text{In}(v)$, $\text{In}(V) = E(W, V) \cup \mathcal{S}(V)$, and $\mathcal{S} = \mathcal{S}(V) \cup \mathcal{S}(W)$.

Since $\text{In}(v)$ contains all inbound edges to vertex v, it informationally dominates all outbound edges from v. Hence $\text{In}(v) \cup \mathcal{T}(v) \subseteq \text{Dom}(\text{In}(v))$, implying that $H(\text{In}(v) \cup \mathcal{T}(v)) \leq H(\text{In}(v))$. By the correctness constraint, we may replace the sink random variables with source random variables. Now consider the set of these entropy inequalities for all $v \in V$. Examining the left-hand sides of these inequalities, each edge in $E(W, V)$ appears exactly once and each source in $\mathcal{S}(V)$ appears exactly twice. Adding these equations together and applying Lemma 2.5 we obtain

$$\begin{aligned} H(\text{In}(V)) + H(\mathcal{S}(V)) \;&\leq\; \sum_{v \in V} H(\text{In}(v)) \\ &\leq \sum_{e \in E(W,V)} H(e) + H(\mathcal{S}(V)). \end{aligned} \tag{5.12}$$

We now claim that $\mathcal{S} \subseteq \text{Dom}(\text{In}(V))$. This holds because for any $S(i) \in \mathcal{S}(W)$, there is no indirect walk from $S(i)$ to $T(i)$ in $G \setminus \text{In}(V)$. This yields the inequality $H(\text{In}(V)) \geq H(\mathcal{S})$. Substituting this into Equation (5.12) and canceling terms, we obtain

$$\sum_{e \in E(W,V)} H(e) \;\geq\; H(\mathcal{S}).$$

By symmetry, a similar inequality holds for the edges in $E(V, W)$. Summing these two inequalities, we obtain an entropy inequality which implies that $\sum_{e \in E} c(e) \geq 2 \sum_{i \in \mathcal{I}} r d_i$. This shows that $r \leq \big(\sum_e c(e)\big) / \big(2 \sum_i d_i\big)$.

This inequality is tight in instances where each source-sink pair is joined by a 2-hop path, and the optimum of the dual to the multicommodity flow LP is achieved by assigning length 1 to every edge of G; we call such instances ***special bipartite graphs***. The Okamura-Seymour example is a special bipartite graph. It is not hard to come up with infinitely many other bipartite graphs satisfying this property.

Theorem 5.1. *The maximum multicommodity flow rate equals the network coding rate for all instances of the k-pairs communication problem on special bipartite graphs.*

6 Network Coding, I/O Complexity, Cell-Probe Model, and Turing Machines

Generalizing Theorem 5.1 to all graphs would prove the undirected k-pairs conjecture. In this section, we provide evidence that this problem is difficult: proving it would resolve a long-standing open question in Input/Output (I/O) complexity, imply the strongest known lower bound in the oblivious cell-probe model, and imply a non-trivial bound for a certain model of two-tape Turing machines. We consider the I/O complexity model of Floyd [11], although our techniques apply equally well to Aggarwal and Vitter's more sophisticated model [3]. In Floyd's model, there is a slow and large memory consisting of pages, each of which contains p records. We assume here that each record consists of a single bit. The basic operation in this model is to read in two pages, and write a new page based on the information in the two read pages. Floyd's original model makes an ***indivisibility*** assumption: the new page consists of a subset of the records in the two read pages. We consider here the effect of relaxing that assumption: the new page can be an arbitrary function of the two read pages.

We consider the matrix transposition problem, where the input is a $p \times p$ matrix stored in row-major order, and the objective is to produce the transpose of the matrix: the same bits stored in the slow memory in column-major order. Floyd demonstrates that, in his model, $p \log p$ I/O operations are sufficient for this problem, and, with the indivisibility assumption, this number of I/Os are also necessary. Aggarwal and Vitter conjectured that the same lower bound is true without the indivisibility assumption, and described proving this as a challenging and important open problem. Any super-linear lower bound on this problem would be a significant advancement.

As a first step, we describe an alternative proof of a lower bound assuming indivisibility. We then show that this new proof would imply lower bounds without the indivisibility assumption if the undirected k-pairs conjecture were true. For this lower bound we assume that the computation is oblivious: the pattern of memory accesses does not depend on the values in the matrix. This assumption is natural for the matrix transposition problem, since the mapping of bits in the input to bits of the output is oblivious.

THEOREM 6.1. *Any indivisible matrix transposition algorithm requires at least $p \log p$ I/O operations.*

Proof. For any computation C performed by the I/O machine, we can construct a graph $G(C)$, which has a vertex for every for every input block and every block written by the I/O machine. We denote the input blocks by $s_1, \ldots, s_p$. There are also vertices corresponding to the (column major) output of the I/O machine, which we denote by $t_1, \ldots, t_p$. The graph has directed edges for each operation, where each non-input vertex v has two edges directed towards it: one from each vertex read by the I/O machine on the operation that produces v.

We also assign to each vertex v the set of records that are written on the operation corresponding to v. Note that for each pair i, j $(1 \leq i, j \leq p)$, there is a record r_{ij} that is assigned to both s_i and t_j. Let N_{ij} be the number of vertices that are assigned r_{ij}. We now claim that for any j $(1 \leq j \leq p)$, $\sum_{i=1}^{p} N_{ij} \geq p \log p$. To see this, note that if a record r is assigned to a non-input vertex v, then r must also be assigned to one of the two vertices with edges directed towards v. Thus, for each i, there is a path from s_i to t_j, with r_{ij} assigned to each vertex. Since every vertex has in-degree at most 2, the union of these paths is a rooted tree with p leaves and at most 2 children per internal node. This establishes the claim. Consequently, $\sum_{i=1}^{p} \sum_{j=1}^{p} N_{ij} \geq p^2 \log p$. The theorem follows from the fact that the total number of records assigned to any vertex can be at most p.

The following theorem shows that if we remove the indivisibility assumption then the $\Omega(p \log p)$ lower bound for matrix transposition still holds, assuming that the undirected k-pairs conjecture is true.

THEOREM 6.2. *If there is an oblivious I/O machine algorithm for matrix transposition that requires $\alpha(p) = o(p \log p)$ operations then there is an undirected k-pairs communication problem with p^2 commodities, where the network coding rate is a factor of $\Omega(\frac{p \log p}{\alpha(p)+p})$ larger than the multicommodity flow rate.*

Proof. Any computation C that uses $N(C)$ I/O operations can be converted to a computation that uses $O(N(C))$ I/O operations and reads any page at most twice. (Simply make a new copy of a page when reading it for the second time.) Thus, we henceforth assume that any page of slow memory is read at most twice. For any computation C requiring $\alpha(p)$ operations, we again consider the graph $G(C)$. Let $\overline{G}(C)$ be the graph obtained by undirecting the edges of $G(C)$. We consider the k-pairs communication problem in $\overline{G}(C)$, where there are p^2 commodities: one for each record r_{ij}, with source s_i, sink t_j, and demand 1. Each edge has capacity p.

We first show that the network coding rate must be at least 1. To see this, construct the network coding solution where each edge has alphabet $\{0,1\}^p$, and the function f_e for any edge e directed out of vertex v in the graph $G(C)$ is the vector of values computed during the I/O operation that wrote vertex v. Since the final result of the I/O computation must have column j of the matrix stored in block t_j, every commodity r_{ij} must be sent successfully to its destination t_j. Next we consider the multicommodity flow rate in $\overline{G}(C)$. Let $L(i,j)$ be the distance from node s_i to t_j in $\overline{G}(C)$. Since $\overline{G}(C)$ has maximum degree at most 4, for any j, $\sum_{i=1}^{p} L(i,j) = \Omega(p \log p)$. Since each $d_i = 1$, the concurrent multicommodity flow rate r is the minimum amount of any commodity that is sent. Let a_{ijk} be the amount of commodity r_{ij} that is sent on edge e_k. It must be the case that for any commodity r_{ij}, $\sum_k a_{ijk} \geq rL(i,j)$. Thus, $\sum_{i,j,k} a_{ijk} \geq rp^2 \log p$. Since the capacity of any edge is at most p, and there are at most $O(\alpha + p)$ edges, it must be the case that $r = O(\frac{\alpha+p}{p \log p})$.

We have shown that proving the undirected k-pairs conjecture implies a $\Omega(p \log p)$ lower bound for matrix transposition in the I/O complexity model. This model may be viewed as a special case of the cell-probe model where each cell contains p bits, there are p read-write cells, and the program is of size $2^{2p} + O(1)$ (i.e., there is free internal storage of $2p$ bits). Thus another consequence of the undirected k-pairs conjecture would be a $\Omega(p \log p)$ lower bound for oblivious matrix transposition algorithms in the cell-probe model. This would represent significant progress since the strongest known oblivious cell-probe lower bound for any problem only provides a $\omega(p)$ lower bound for *space-bounded* computation (i.e., when there are $o(p)$ read-write cells) [7, 6].

Furthermore, proving the undirected k-pairs conjecture implies a non-trivial lower bound for a certain model of two-tape Turing machines. Specifically, we consider *oblivious* Turing machines [28], where, for any input of size n, the Turing machine tape head follows the same sequence of left and right moves. This follows since the following theorem shows that the I/O complexity model can simulate the oblivious Turing machine.

THEOREM 6.3. *Any two-tape Turing machine computation that runs in β steps can be performed in $O(\beta/p)$ steps of an I/O machine. Also, any oblivious two-tape*

Turing machine computation that runs in β steps can be performed in $O(\beta/p)$ steps of an oblivious I/O machine.

7 Network Coding Gap of Directed Graphs

We turn now to considering the ***network coding gap*** for directed graphs, i.e., the ratio of the coding rate and the flow rate. We describe a family of directed graphs that have a large network coding gap. (This family of graphs also demonstrates that the network coding rate can exceed the sparsity, though this fact was already well-known, e.g., because of the instance illustrated in Figure 2(b).) The graphs are defined recursively, where $\mathscr{G}(1)$ is the graph shown in Figure 2(b), in which all commodities have unit demand, all edges have unit capacity, and the network coding gap is equal to 2 (as is proved below). The graph $\mathscr{G}(n)$ is built from $\mathscr{G}(n-1)$ as follows: for each commodity i in $\mathscr{G}(n-1)$, we eliminate the demand for commodity i and replace it with two commodities each with unit demand, by taking a copy of $\mathscr{G}(1)$, deleting the edge $e = (a, b)$ from it, and adjoining the resulting graph to $\mathscr{G}(n-1)$ by identifying $a \in V(\mathscr{G}(1))$ with $s(i) \in V(\mathscr{G}(n-1))$ and identifying $b \in V(\mathscr{G}(1))$ with $t(i) \in V(\mathscr{G}(n-1))$. This construction is illustrated in Figure 2(c).

We now analyze the network coding gap in $\mathscr{G}(n)$. An inductive argument shows that the total multicommodity flow rate in $\mathscr{G}(n)$ is at most 1, since all flow must cross edge e. Hence the concurrent multicommodity flow rate is at most $1/2^n$ since there are 2^n commodities. However, $\mathscr{G}(n)$ has a rate 1 network coding solution where edge alphabets are $\Gamma = \{0, 1\}$ and each node simply sends the XOR of its inputs on all of its outputs. This proves the first half of the following theorem; a proof of the second half will appear in the full version of this paper.

THEOREM 7.1. *For all $k \geq 2$, there is an instance of the k-pairs communication problem on a DAG $G = (V, E)$ with $|V| = \Theta(k)$ and $|E| = \Theta(k)$ where the network coding gap is $\Theta(k)$. Moreover, for any instance of the k-pairs communication problem on a graph $G = (V, E)$, the network coding gap is at most* $\min\{|E|, |V|, k\}$.

8 Open problems

The most prominent open question related to our work is the undirected k-pairs conjecture. Resolving the conjecture in either direction would profoundly increase our understanding of the power of network coding as a model of information transmission in networks, and resolving it affirmatively would also lead to new lower bounds as described in Section 6. Without settling the conjecture for general networks, it would still be interesting to discover other infinite classes of graphs for which the conjecture can be verified, or to prove that the gap between sparsity and network coding rate in undirected graphs can be as large as $\Omega(\log n)$, as in the case of the maximum multicommodity flow rate.

It is known [32] that entropy functions satisfy some non-trivial inequalities which are *not* consequences of the polymatroidal inequalities; these are known as *non-Shannon-type* inequalities. Characterizing all such inequalities remains a challenging open problem. It is also unknown whether there are instances in which combining non-Shannon-type inequalities with the inequalities defined in Section 4 yields a tighter bound on network coding rate than can be proved by applying the inequalities of Section 4 alone.

Much of the past work on network coding has focused on the special case of *linear* or *vector-linear* codes, in which the edge alphabets $\Gamma(e)$ are vector spaces over a field $\mathbb{F}$, and the coding functions f_e are linear functions. It is possible that one can prove stronger upper bounds in this special case, or perhaps even settle the undirected k-pairs conjecture. Dougherty et al [9] have recently given an example of a directed network in which linear codes can not achieve the maximum rate achievable by non-linear codes. How large can the gap be between linear and non-linear network coding rates?

Finally, most questions concerning the computational complexity of network coding problems remain wide open. Some hardness results for linear network coding were proved in [23], but nothing is known about the complexity of general (i.e., non-linear) network coding problems. For example, neither of the following alternatives is currently known to be false:

- Given an instance of the general network coding problem in a directed graph G and a real number r, there is a polynomial-time algorithm which computes a solution achieving rate r if one exists, and otherwise reports that this is impossible.
- Given an instance of the k-pairs communication problem in an undirected graph G and a real number r, it is recursively undecidable to determine whether the network coding rate is less than r.

Acknowledgments

The authors would like to thank the many people with which they have had helpful discussions on this subject. These include Christian Borgs, Jennifer Chayes, Phil Chou, Uri Feige, Michel Goemans, David Karger, Lap Chi Lau, Eric Lehman, Tom Leighton, Laci Lovász, Mohammad Mahdian, Aranayak Mehta, Peter Montgomery, Amin Saberi, Madhu Sudan, Kunal Talwar, Vijay Vazirani, Raymond Yeung, and Gideon Yuval. We also thank Mihai Pătraşcu for explaining the connection between our work and cell-probe lower bounds.

References

[1] M. Adler and T. Leighton. Compression using efficient multicasting. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing (STOC)*, May 2000.

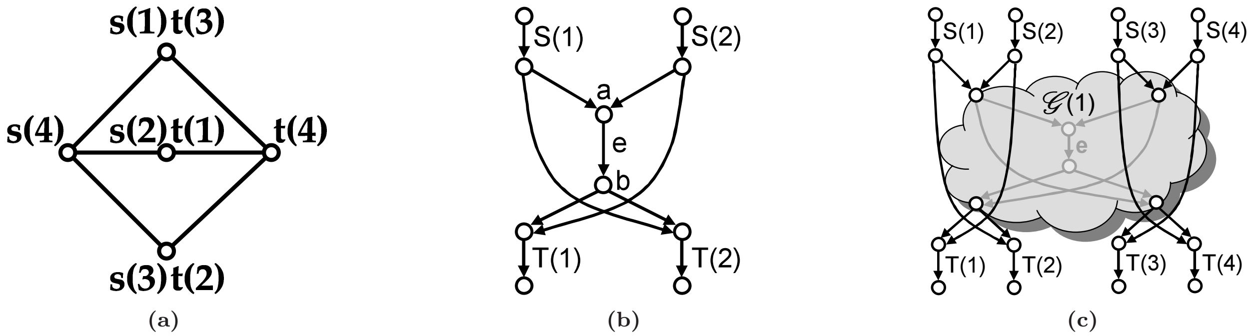

Figure 2: (a) The Okamura-Seymour example. Each edge has capacity 1 and each commodity has demand 1. The sparsity is 1 and the flow rate and coding rate are both 3/4. (b) The instance $\mathscr{G}(1)$. The edge e has unit capacity and all others have infinite capacity. (c) The instance $\mathscr{G}(2)$ is constructed recursively from $\mathscr{G}(1)$.

[2] A. Agarwal and M. Charikar. On the advantage of network coding for improving network throughput. In *Proceedings of the IEEE Information Theory Workshop*, Oct. 2004.

[3] A. Aggarwal and J. Vitter. The input/output complexity of sorting and related problems. *Communications of the ACM*, pages 116–127, Sept. 1988.

[4] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, July 2000.

[5] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[6] L. Babai, N. Nisan, and M. Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *Journal of Computer and System Sciences*, 45(2):204–232, Oct. 1992.

[7] P. Beame, M. Saks, X. Sun, and E. Vee. Time-space trade-off lower bounds for randomized computation of decision problems. *Journal of the ACM*, 50(2):154–195, Mar. 2003.

[8] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 1991.

[9] R. Dougherty, C. Freiling, and K. Zeger. Insufficiency of linear coding in network information flow. *IEEE Transactions on Information Theory*, 51(8), Aug. 2005.

[10] C. Dwork. *Bounds on Fundamental Problems in Parallel and Distributed Computation*. PhD thesis, Cornell University, 1984.

[11] R. Floyd. Permuting information in idealized two-level storage. In R. Miller and J. Thatcher, editors, *Complexity of Computer Calculations*, pages 105–109. Plenum, 1972.

[12] M. T. Hajiaghayi and H. Räcke. An $O(\sqrt{n})$-approximation algorithm for directed sparsest cut. Submitted Manuscript.

[13] N. J. A. Harvey, D. R. Karger, and K. Murota. Deterministic network coding by matrix completion. In *Proceedings of the 16th Annual Symposium on Discrete Algorithms (SODA)*, pages 489–498, Jan. 2005.

[14] N. J. A. Harvey, R. D. Kleinberg, and A. R. Lehman. Comparing network coding with multicommodity flow for the k-pairs communication problem. Technical Report 964, MIT LCS, September 2004.

[15] T. Ho, R. Koetter, M. Médard, D. R. Karger, and M. Effros. The benefits of coding over routing in a randomized setting. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, June 2003.

[16] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen. Polynomial time algorithms for multicast network code construction. *IEEE Transactions on Information Theory*, 51(6), June 2005.

[17] K. Jain, M. Mahdian, and M. Salavatipour. Packing Steiner trees. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 266–274, Jan. 2003.

[18] K. Jain, V. Vazirani, R. W. Yeung, and G. Yuval. On the capacity of multiple unicast sessions in undirected graphs. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, Sept. 2005.

[19] G. Kramer. Capacity results for the discrete memoryless network. *IEEE Transactions on Information Theory*, 49(1):4–21, Jan. 2003.

[20] G. Kramer and S. Savari. Progressive d-separating edge set bounds on network coding rates. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, Sept. 2005.

[21] G. Kramer and S. A. Savari. On networks of two-way channels. In *Proceedings of the DIMACS Workshop on Algebraic Coding Theory and Information Theory*, Dec. 2003.

[22] L. C. Lau. An approximate max-Steiner-tree-packing min-Steiner-cut theorem. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 61–70, 2004.

[23] A. R. Lehman and E. Lehman. Complexity classification of network information flow problems. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Jan. 2004.

[24] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46(6):787–832, Nov. 1999.

[25] Z. Li and B. Li. Network coding in undirected networks. In *Proceedings of the 38th Annual Conference on Information Sciences and Systems (CISS)*, Mar. 2004.

[26] Z. Li and B. Li. Network coding: The case of multiple unicast sessions. In *Proceedings of the 42nd Allerton Annual Conference on Communication, Control, and Computing*, Sept. 2004.

[27] Z. Li, B. Li, D. Jiang, and L. C. Lau. On achieving throughput with network coding. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Mar. 2005.

[28] N. Pippenger and M. J. Fischer. Relations among complexity measures. *Journal of the ACM*, 26(2):361–381, Apr. 1979.

[29] S. Riis. Linear versus non-linear boolean functions in network flow. In *Proceedings of the 38th Annual Conference on Information Sciences and Systems (CISS)*, Mar. 2004.

[30] M. Saks, A. Samorodnitsky, and L. Zosin. A lower bound on the integrality gap for minimum multicut in directed networks. *Combinatorica*, 24(3):525–530, 2004.

[31] L. Song, R. Yeung, and N. Cai. Zero-error network coding for acyclic networks. *IEEE Transactions on Information Theory*, 49:3129–3139, Dec. 2003.

[32] Z. Zhang and R. W. Yeung. On characterization of entropy function via information inequalities. *IEEE Transactions on Information Theory*, 44(4), July 1998.