# Information flows, graphs and their guessing numbers

Søren Riis
Queen Mary, University of London
smriis@dcs.qmul.ac.uk

Submitted: Jan 9, 2006; Accepted: May 28, 2007; Published: Jun 7, 2007
Mathematics Subject Classification: 05C20, 68Q17,68R10

**Abstract**

Valiant's shift problem asks whether all $n$ cyclic shifts on $n$ bits can be realized if $n^{(1+\epsilon)}$ input output pairs ($\epsilon < 1$) are directly connected and there are additionally $m$ common bits available that can be arbitrary functions of all the inputs. If it could be shown that this is not realizable with $m = O(\frac{n}{\log\log n})$ common bits then a significant breakthrough in Boolean circuit complexity would follow. In this paper it is shown that in certain cases all cyclic shifts are realizable with $m = (n - n^{\epsilon})/2$ common bits. Previously, no solution with $m < n - o(n)$ was known, and Valiant had conjectured that $m < n/2$ was not achievable. The construction therefore establishes a novel way of realizing communication in the manner of Network Coding for the shift problem, but leaves the viability of the common information approach to proving lower bounds in Circuit Complexity open. The construction uses the graph-theoretic notion of guessing number.

As a by-product the paper also establish an interesting link between Circuit Complexity and Network Coding, a new direction of research in multiuser information theory.

## 1 Introduction

The problem of proving superlinear lower bounds on the size of circuits for an explicitly defined sequence of Boolean functions is still open after more than 30 years of intensive research in Complexity Theory. The problem is open even if we consider the case where we look for functions with $n$ input bits and $n$ output bits, and where the depth of the circuit is in $O(\log(n))$. For a detailed discussion and further survey of this class of problems and their link to communication complexity and matrix rigidity see [7].

In this paper we relate this fundamental problem in Complexity Theory (more specifically we focus on Valiant's Shift problem that has been - and is still open, for more than 30 years), to a new type of problem in Graph Theory. Each directed graph has (for each $s \in \{2, 3, \ldots\}$) associated a number (the guessing number), we will define in this paper.

The notion of guessing number (that was first introduced by Riis in [10] and [9]) is new. We link the guessing number to solvability of circuit information flow problems. These are problems that are closely related to problems in Network Coding. Network Coding is a new interesting direction of research in multiuser information theory (see for example [5, 8, 4, 6, 14, 1, 13, 2, 3]).

Maybe the main contribution of this paper is to link central problems in Circuit Complexity Theory with the area of Network Coding (multiuser information theory) and Graph Theory (guessing numbers).

In [12] Valiant put forward four related conjectures. It turns out that the two most "risky" of these are not quite valid. We reduce the two other (and more safe) conjectures to a pure graph theoretical problem concerning guessing numbers.

In my judgement (and this paper provide some evidence of this) progress in understanding and bounding guessing numbers for various natural classes of graphs is needed in order to solve some of the long-standing open questions in Circuit Complexity problems.

# 2 A game of cooperation

Consider the following game: assume that $n$ players each has a fair $s$-sided die (each die has its sides labelled as $1, 2, \dots s$). The players (simultaneously) throws their dice in such a manner that no player knows the value of their own die. Suppose each player has to guess the value of their own die. The probability that each of the $n$ players is able to guess correctly the value of their own die is $(\frac{1}{s})^n$.

*Assume now that each player knows the values of all dice except the value of their own die. What is the probability that each of the $n$ players correctly guesses the value of their own die?*

From a superficial perspective it might appear that, since each of the players only has access to "irrelevant" information, the probability that all $n$ players guess their own die value correctly remains $(\frac{1}{s})^n$. As it happens the question is ill-posed since the probability actually depends on which "strategy" the players adopt!

If each player, for example, believes (and acts accordingly) that the sum of all dice values (including their own die) is divisible by $s$, the probability that all players (simultaneously) guess their own dice value correctly is $\frac{1}{s}$.

Thus the players have a collective guessing strategy that ensures that all players are correct if (and only if) one player is right. Either all players are right ($p = \frac{1}{s}$), or all players are wrong ($p = 1 - \frac{1}{s}$).

Intuitively it should be quite clear what a guessing strategy for the players is. Player $j$ ($j \in \{1, 2, 3, \dots, n\}$) receives die values $x_1, x_2, \dots, x_{j-1}, x_{j+1}, \dots, x_n$ and calculates a value $f_j(x_1, x_2, \dots, x_{j-1}, x_{j+1}, \dots, x_n) \in \{1, 2, \dots, s\}$. This value represents player $j$'s guess. Thus each guessing strategy is given by $n$ functions $f_1, f_2, \dots, f_n$. The total number of guessing strategies is $s^{ns^{n-1}}$. For each of these strategies there is associated a probability that all players simultaneously guess correctly their own dice values. A strategy that leads to a probability that is maximal is called an optimal strategy.

We just showed that an optimal guessing strategy achieves a probability of at least $\frac{1}{s}$, and since the probability that a given single player guess correctly is $\frac{1}{s}$, this probability is indeed optimal. The players have actually many different optimal guessing strategies. One type of optimal strategy generalises naturally the "0 modulo $s$" strategy we already considered. This strategy appears if the players agree in advance to fix a group $G$ with $s$ elements (so each die value is an element in the group $G$). Furthermore the players agree that the product of all $n$ dice values is $1 \in G$ (or any other fixed element $g \in G$). Each player can calculate (also if $G$ is non-commutative) the unique die value that makes the total product 1. If each player 'guesses' according to this strategy, the players guess their own die value correctly if and only if the product (in the group $G$) of the dice values is 1. This happens with probability $\frac{1}{s}$.

Actually, it is not hard to see that the set of optimal guessing strategies consists exactly of the strategies that can be each defined by a $n$ dimensional latin hypercubes of order $s$. A latin hypercube (of order $s$) is the obvious generalisation of a latin square (of order $s$) to higher dimensions. So a 2 dimensional latin hyper cube of order $s$ is an ordinary latin square of order $s$, and a 3 dimensional latin hypercube of order $s$ is a latin cube of order $s$. In general, we can view an $n$-dimensional latin hyper cube of order $s$ as a mapping $f : \{0, 1, 2, \ldots, s-1\}^n \to \{0, 1, 2, \ldots, s-1\}$ that maps $A$ bijectively to $A$ whenever $n-1$ of the arguments of $f$ are fixed.

# 3 Playing the guessing game on a graph

The class of games we considered in the previous section can be viewed as a subclass of a much wider class of cooperative games.

Graphs in our setting are always directed graphs. Formally a graph $G = (V, E)$ is a pair of sets with $E \subseteq V \times V$. As usual there is an edge from $v \in V$ to $w \in V$ if and only if $(v, w) \in E$.

For each graph $G = (V, E)$ and for each value $s \in \{2, 3, 4, \ldots, \}$ we define a cooperative game. The game denoted by $\mathrm{Game}(G, s)$ is played as follows: Each node (vertex) $v \in V$ corresponds to a player, and each of the players independently gets assigned a die value from a finite set $A$ of $s$ elements. As in the previous game the task of the players (as a group) is to maximise the probability that they all simultaneously correctly 'guess' their own dice value. The die value of player $v \in V$ is available to each player $w \in V$ with $(v, w) \in E$. In other words, player $w \in V$ knows the dice value of the players $v \in V$ with $(v, w) \in E$. If $(v, v) \in E$ player $v$ knows the value of his/her own die.

A strategy for a player $j$ in a node of in-degree $d$ is given by a function $f_j$ that maps $\{1, 2, \ldots, s\}^d$ to $\{1, 2, \ldots, s\}$. The total number of cooperative strategies is given by $s^{\Sigma_{j=1}^{n} s^{d_j}}$ where $d_j$ denotes the in-degree in node $j$.

The guessing game in the previous section corresponds to the complete graph on $n$ nodes.

In the guessing game that corresponds to the complete graph, the players have an (optimal) strategy that guarantees all players guess correctly their own die value with probability $\frac{1}{s}$. This probability is a factor $s^{n-1}$ better than uncoordinated random guessing, a fact that will show us that the guessing number of the complete graph is $n-1$ (since $s$ is raised to the power $n-1$).

As an example let us consider the graph on $n$-vertex that forms one oriented cycle. In other words, let $G = (V, E)$ where $V = \{1, 2, \ldots, n\}$ and $E = \{(1, 2), (2, 3), \ldots, (j, j+1), \ldots, (n-1, n), (n, 1)\}$. As in the previous game, it is intuitively clear what is a strategy in this game. Player $j$ receives the value $x_{j-1}$ (and player 1 receives the value $x_n$). A guessing strategy is a set of functions $f_1, f_2, \ldots, f_n$ that each maps $\{1, 2, \ldots, s\}$ to $\{1, 2, \ldots, s\}$. In this game the number of strategies is $s^{ns}$. The players actually have a strategy $\tilde{S}$ that ensures that they are all able to guess their own dice values (simultaneously) with a higher probability than pure uncoordinated random guessing. If each player assumes that the value of their own die is the same as the value they receive, all the players are correct, if and only if all dice values are identical. This happens with probability $(\frac{1}{s})^{n-1}$. This is a factor $s$ times better than pure uncoordinated random guessing. The strategy $\tilde{S}$ is optimal since no cooperative guessing strategy can succeed with higher probability than $(\frac{1}{s})^{n-1}$ (to see this notice that for *any* guessing strategy, any subset of $n-1$ players cannot do better than uncoordinated random guessing. Thus the best that can be achieved is that every time $n-1$ players guess correctly, all $n$ players guess correctly).

**Definition** :

A graph $G = (V, E)$ has for $s \in N$ *guessing number* $k = k(G, s)$ if the players in the guessing game Game$(G, s)$ associated to $G$ and $s$ have a strategy that ensures that they all guess correctly their own dice values with probability $(\frac{1}{s})^{|V|-k}$.

In other words a graph $G$ has guessing number $k$ if the players have a strategy that succeeds with probability $s^k$ times higher than uncoordinated random guessing.

It turns out that for many graphs the guessing number is independent of $s$ and is an integer. However, in general it is possible to show that there exist graphs where $k = k(G, s)$ depends on $s$, and where the guessing number is not always an integer.

# 4 Information flow problems and guessing numbers

We will now consider another type of problem that turns out to be very closely related to the guessing games.

Assume that we are given an acyclic graph $N$ with input nodes $i_1, i_2, ..., i_n$ and output nodes $o_1, o_2, \ldots, o_n$ as well as an alphabet $A$. Each input node has indegree 0, and each output node has outdegree 0. The problem is whether $N$ can be turned into a circuit that for each array of inputs $(x_1, x_2, ..., x_n) \in A^n$ where $x_j \in A$ is the input to input node $i_j$ $(j = 1, 2, ..., n)$, output the same array $(x_1, x_2, ..., x_n) \in A^n$ i.e. $x_j$ is the output of $o_j$ $(j = 1, 2, ..., n)$.

In other words the problem is whether $N$ can be turned into a circuit that computes the identity function when input nodes and output nodes are listed with increasing index from left to right.

The answer to this problem depends in general not only on $N$ but also on the size of the underlying alphabet $A$ from which the inputs as well as function values are taken. In circuit complexity usually $A$ contain only the two "letters" 0 and 1, but the alphabet can in our setting be any (finite) set given in advance. We say the information flow problem $(N, A)$ is solvable if $N$ can be turned into a circuit that computes the identity function over the alphabet $A$ as explained above.

Let $G_N$ denote the directed graph that appears by identifying each input node with the corresponding output node (i.e. by identifying the nodes $i_j$ and $o_j$ for each $j = 1, 2, ..., n$). This conversion of an information network to the graph $G_N$ is unique.

The (surprising) link between information flow problems and directed graphs is based on the following fact:

**Theorem(1)** :

The information flow problem $(N, A)$ with $n$ input/output nodes has a solution over alphabet $A$ with $|A| = s$ if and only if the graph $G_N$ has guessing number $k(G, s) \geq n$.

Furthermore $G_N$ has guessing number $k(G, s) \geq n$ if and only if $k(G, s) = n$.

An information problem involves mathematically speaking slightly complicated concepts like *set of source nodes*, *set of receiver nodes* as well as set *of requirements* that specifies the destination of each input. As pointed out in [10, 9] the main point of the theorem is that it replace the information flow problem with an equivalent problem that can be expressed in pure graph theoretic terms (no special input or output nodes).

One important point is that each multiple unicast problem - an important type of communication networks problem in Network Coding - essentially is equivalent to an Information flow problem as I just defined it. The conversion from one to the other can be achieved via the well-known graph to line-graph conversion (see for example [8]).

The theorem can in fact be stated in a stronger form:

**Theorem(2)** :

The set of solutions of a information flow problem $(N, A)$ with $n$ input/output nodes are in one-to-one correspondence with the set of guessing strategies (over alphabet $A$ with $|A| = s$) that ensure that the players in the guessing game played on $G_N$ have success with probability (at least) $(\frac{1}{s})^{|G_N|_{nodes} - n}$ (where $|G_N|_{nodes}$ denotes the number of nodes in $G_N$).

Theorem 2 was first stated and proved in [10, 9]. Here I present a proof in a slightly more compact format.

**Proof of theorem 1 and 2**: Consider the graph $G_N = (V, E)$. The set $V$ of nodes can be divided into two disjoint sets: the set $I$ of nodes in $G_N$ that corresponds to the inner nodes in $N$ (i.e. nodes that are not input or output nodes in $N$), and the set $J$ of $n$ nodes in $G_N$ that corresponds to the $n$ input and $n$ output nodes in $N$. The set $I$ consists of $|G_N| - n$ nodes. The sub-graph of $G_N$ restricted to $I$ is an acyclic graph (since $N$ is acyclic). Thus as we already noticed for any strategy by the players (it does not matter which) the nodes in $I$ all guess correctly their own die value with probability $(\frac{1}{s})^{|I|}$. But, this shows that the probability all players in $G_N$ guess correctly their own die value is at most $(\frac{1}{s})^{|I|}$. Theorem 1 follows because this probability can be achieved if and only if the players in $J$ (corresponding to the output nodes in $N$) are able to work out their own die value with probability 1 (given that all players in $I$ correctly worked out their own die values).

To prove Theorem 2 consider a guessing strategy (i.e. a set of specific functions assigned to the nodes in $G_N$).

If we assign the same functions to the information network $N$ (the output nodes $o_1, o_2, \ldots, o_n$ get assigned the specific functions assigned to the nodes $J$ in $G_N$).

Conversely, any attempted solution to $N$ can be converted to a guessing strategy by the same assignment. Thus the space of coding functions for $N$ is in a natural 1-1 correspondence with the space of guessing strategies to the graph $G_N$. Furthermore, a coding function for $N$ solves the information problem for $N$ if and only if the conditional probability that the $n$ nodes in $J$ guess correctly their own die values (*given* that all "inner" nodes (i.e. all the nodes in $I$) guess correctly their own die values) is 1. ♣

# 5 Analysis of a specific class of graphs

One interesting class of graphs that are linked to the conjectures I will put forward later, consists of (what I will call) Clock-graphs. For each pair of numbers $(n, r)$ with $r \leq \lfloor (n-1)/2 \rfloor$ we define a graph $G_{clock}(n, r)$. It has vertex set $V := \{0, 1, 2, \ldots, n-1\}$ and edge set $E := \{(v, w) : w - v \in \{1, 2, \ldots, r\}$, where the difference $w - v$ is calculated modulo $n$.

We will show:

**Proposition A** :

The graph $G_{clock}(n, r)$ has guessing number $r$.

**Proof:** Consider the subgraph of $G_{clock}(n, r)$ that contains the vertex $\{0, 1, 2, \ldots, n-r-1\}$. This graph is acyclic (because $r \leq \lfloor (n-1)/2 \rfloor$) so the players cannot do better than pure uncoordinated random guessing. Thus the nodes $0, 1, 2, \ldots, n-r-1$ are all correct with probability $(\frac{1}{s})^{n-r}$. Thus, the players in the graph $G_{clock}(n, r)$ are all correct with probability at most $(\frac{1}{s})^{n-r}$. Equality appears if and only if the players have chosen a strategy that ensures that all $n$ players guess correctly their own die value if and only if players $0, 1, 2, \ldots, n-r-1$ all guess their own dice value.

Using simple linear algebra it is now straight forward for the players to derive their own dice values from the $n-r$ values $x_j + x_{j+1} + \ldots + x_{j+r} = 0$ modulo $s$ for $j = 0, 1, \ldots, n-r-1$ (public information). ♣

A more general class of clock graphs appears if we, for each $n$ and for each subset $S \subset \{0, 1, 2, \ldots, n-1\}$ with $0 \notin S$ and $v \in S \to -v \notin S$, define a graph $G_{clock}(n, S)$ with vertex set $V := \{0, 1, 2, \ldots, n-1\}$ and edge set $E := \{(v, w) : w - v \in S\}$. The resulting graphs have no self-loops (since $0 \notin S$) and it has no undirected edges (since $(v, w) \in E \to w - v \in S \to v - w \notin S \to (w, v) \notin E$).

It is an open problem whether the graph $G_{clock}(n, S)$ in general has guessing number $\leq |S|$.

# 6 A Game with public information

Let me briefly return to the $n$ player game on the complete graph $K_n$. As we noticed the players have a guessing strategy so they all guess correctly if the sum of the dice values are 0 modulo $s$.

Actually if the players know the sum of the dice values modulo $s$ each player can deduce the value of their own die. Put in other terms, if each player has access to a public channel that can broadcast one of $w = s$ possible messages (i.e. the channel has bandwidth $b(G, s) := 1$) they can all (provided the message broadcast is the correct one), deduce the value of their own die.

As another example, consider the graph $G = (V, E)$ with vertex set $V := \{0, 1, 2, \ldots, n-1\}$ and edge set $E := \{(0, 1), (1, 2), \ldots, (j, j+1), \ldots, (n-2, n-1), (n-1, 0)\}$. We already notice that its guessing number was 1, i.e. that the players have a strategy that guarantees they all guess correctly with probability $(\frac{1}{s})^{n-1}$ and that this is the best possible result they can achieve. Assume that the players know (through a public broadcast) the values $x_0 - x_1, x_1 - x_2, \ldots, x_j - x_{j+1}, \ldots, x_{n-2} - x_{n-1}$ where $x_j$ denotes the value of die $j$. Then each player can deduce their own die value. Player $j+1$ knows the value of $x_j$ and $x_{j+1} - x_j$ from which $x_{j+1}$ can be calculated. Player 0, can calculate $x_0 - x_{n-1}$ and know the value of $x_{n-1}$ from which $x_0$ can be derived. Thus if the public channel has bandwidth $b(G, s) := n - 1$ (can broadcast $s^{k-1}$ messages), the players can always derive the value of their own die.

For an alphabet size $s$ we define the *information defect* $b(G, s)$ of a graph $G$ as $\log_s(w)$ where $w$ denotes the smallest number of public messages $m_1, m_2, \ldots, m_w$ that can be broadcasted to all players, ensuring that each player is always able to deduce the value of his/her die.

In general, where $G = (V, E)$ can be any graph, the guessing number $k(G, s)$ and the information defect $b(G, s)$ are related. In general, $k(G, s) + b(G, s) \geq |V|$ and for many graphs $k(G, s) + b(G, s) = |V|$.

**Lemma 3** :

For any graph $G = (V, E)$ and any $s \in \{2, 3, 4, \ldots\}$ $|V| \leq k(G, s) + b(G, s)$.

**Proof:** Let $G$ be a (directed) graph and with information defect $b = b(G, s)$. Then, by definition, there exists a method by which we, by broadcasting one of $s^{b(G,s)}$ messages to all nodes, can insure that each node $j$ can derive their own die value $x_j \in A$. If the dice values $x_1, x_2, \ldots, x_{|V|}$ are selected randomly (and independent), each message $m$ is broadcasted with a certain probability $p(m)$. Since $\Sigma_m p(m) = 1$, there must exist a message $m_0$ that is broadcasted with probability $\geq (\frac{1}{s})^{b(G,s)}$. This is a factor $s^{|V|-b(G,s)}$

better than pure uncoordinated random guessing in the guessing game on $G$. Thus if the players in the guessing game all guess their own die value under the assumption that the public message $m_0$ is broadcasted, they are all correct if this is indeed the case. Thus the guessing number $k(G, s)$ is $\geq |V| - b(G, s)$ and $b(G, s) + k(G, s) \geq |V|$. ♣

We will now state and prove a theorem related to Theorem 1.

**Theorem 4** :

> An information flow problem $(N, A)$ with $n$ input nodes, $n$ output nodes and $r$ internal nodes, has a solution if and only if the graph $G_N$ has information defect $b(G, |A|) \leq r$. Furthermore $G_N$ has information defect $b(G, |A|) \leq r$ if and only if $b(G, |A|) = r$.

**Proof:** (only if): Assume first that the flow problem $N$ has a solution. We want to show that $b(G, s) \leq r$. The set of nodes in $G_N$ can be divided into two disjoint sets. The set $I$ of nodes that corresponds to the inner nodes in $N$ and the set $J$ of $n$ nodes in $G_n$ that corresponds to the $n$ input and $n$ output nodes in $N$. For each inner node let $z_j$ denote the actual value of player $j$'s die. Let $\tilde{z}_j$ denote the value the corresponding node in $N$ would take according to the solution where all $n$ inputs are given by the actual die values assigned to the $n$ nodes in $J$. Assume that all players have access to the values $z_j - \tilde{z}_j$ for each $j \in I$. This information can be provided by a public channel of bandwidth $r$. Each player can now calculate their own die value. To see this consider first a player that corresponds to an inner node $j$ in $G_N$. This player has access to some inner nodes and possibly to some input nodes (nodes in $J$). From these values and the public information, the player can calculate $\tilde{z}_j$. And from $z_j - \tilde{z}_j$ (that is publicly available) the player can calculate $z_j$.

Consider now a node in $j \in J$. This node $j$ has access to nodes in $I$, as well as possibly some nodes in $J$. For each node $i \in I$ that have a value that is available to node $j$, the node has access to $z_i$ as well as the public information $z_j - \tilde{z}_i$. From this node $j$ can calculate $\tilde{z}_i$. But then each node $j \in J$ can calculate $x_j$.

(if): Assume that we can send $r$ messages through the public channel in such a fashion each node in $G_N$ can calculate its own value. But then $G_n$ has information defect $b(G_N, s) \leq r$. We already showed that the guessing number $b(G_N, s) + k(G_N, s) \geq |V| = n + r$. Thus $k(G_N, s) \geq n$ which according to Theorem 1 ensures that the Circuit information flow problem $N$ has a solution. Again according to Theorem 1, $k(G_N, s) = n$ and thus $b(G_N, s) \geq r$. Thus $b(G_N, s) = r$ if and only if $b(G_N, s) \leq r$. ♣

# 7 Valiant Conjectures

In [12] (based on [11]) Valiant put forward four conjectures. In this section I will present the conjectures using Valiant's terminology. Valiant introduces his conjectures as follows:

> Let $G$ be a bipartite graph with node set $X \cup Y$ where $X = \{x_1, x_2, \ldots x_n\}$ and $Y = \{y_1, y_2, \ldots y_n\}$ denote input variables and output functions respectively.

Suppose that edges are defined implicitly by a mapping $\tau$ where $\tau(y_i) \subseteq X$ is the set of input nodes that are adjacent to $y_i$ in the graph. Define $m$ Boolean functions $f_1(x), f_2(x), \dots f_m(x)$ and $n$ further Boolean functions $g_1, g_2, \dots, g_n$, where $g_i$ has $m+|\tau(y_i)|$ Boolean arguments, such that with some abuse of notation, each $y_i$ can be an arbitrary function of these $m$ common bits and the inputs $\tau(y_i)$ to which it is connected directly in $G$. Valiant goes on to say that $G$ realizes a permutation $\rho$ with $m$ common bits if there exists $f_1, f_2, \dots f_m, g_1, g_2, \dots g_n$ such that for all $i, 1 \le i \le n$

$$x_{\rho(i)} = g_i(f_1(x), \dots f_m(x), \tau(y_i)).$$

In other words (still quoting Valiant), for the fixed $G$ and given $\rho$ one can find the appropriate Boolean functions such that $y_i$ realises the permutation $\rho$ of $\{x_i\}$ for all truth assignments of $\{x_i\}$. Valiant then goes on to conjecture:

**Conjecture(1)** :

If $G$ has degree $\le 3$ (for sufficiently large $n$) there is a permutation $\rho$ such that $G$ does not realize $\rho$ with $n/2$ common bits.

**Conjecture(2)** :

If $G$ has $O(n^{1+\epsilon})$ edges for some $\epsilon > 0$, for $n$ sufficiently large, there is a permutation $\rho$ such that $G$ does not realize $\rho$ with $n/2$ common bits.

**Conjecture(1′)** :

If $G$ has degree $\le 3$ (for sufficiently large $n$) there is a cyclic shift $\rho$ such that $G$ does not realize $\rho$ with $n/2$ common bits.

**Conjecture(2′)** :

If $G$ has $O(n^{1+\epsilon})$ edges for some $\epsilon > 0$, for $n$ sufficiently large, there is a cyclic shift $\rho$ such that $G$ does not realize $\rho$ with $n/2$ common bits.

Conjecture $2'$ is in some sense the most important of Valiant's conjectures. Valiant showed that if this conjecture have direct consequences and leads to new results in circuit complexity. These Complexity questions have now been open for more than 30 years. As it happens, we will show that Conjecture $2'$ is false. However, it can be replaced by a slightly stronger (and presumably correct) conjecture that essentially have the desired consequences in Circuit Complexity Theory.

# 8 Expressing Valiant's conjectures in terms of guessing numbers

Assume that we are given a graph $G$ with vertex set $V := \{0, 1, 2, \dots, n-1\}$ and edge set $E \subseteq V \times V$.

For each $s \in \{0, 1, 2, \ldots, n-1\}$ we define the graph $G^s$ ($G$ shifted by $s$) as the graph with vertex set $V$ and edge set $E^s := \{(i, j+s) : i \in V, (i, j) \in E\}$ where $j+s$ is calculated modulo $n$. This definition depends on both the graph $G$ as well as on the labelling of its vertex.

For each permutation $\pi \in S_n$ we also define the graph $G^\pi$ ($G$ permuted by $\pi$) as the graph with vertex set $V$ and edge set $E^\pi := \{(i, \pi(j)) : i \in V, (i, j)) \in E\}$. Unlike the shift operation, the permutation operation is independent of the labelling of the underlying vertex set.

We will now state four conjectures that are equivalent to the four conjectures discussed in the previous section.

**Conjecture 1** :

> Let $G$ be a graph with $n$ vertex and assume that all vertex have out-degree (in-degree) $\leq 3$. Then (provided $n$ is sufficiently large) there exists a permutation $\pi \in S_n$ such that $G^\pi$ has information defect $b(G^\pi, 2) > n/2$.

**Conjecture 2** :

> Let $\epsilon > 0$. Let $G$ be a graph with $n$ vertex and less than $n^{1+\epsilon}$ edges. Then (provided $n$ is sufficiently large) there exists a permutation $\pi \in S_n$ such that $G^\pi$ has information defect $b(G^\pi, 2) > n/2$.

**Conjecture 1′** :

> Let $G$ be a graph with $n$ vertex and assume that all vertex have out-degree (in-degree) $\leq 3$. Then (provided $n$ is sufficiently large) there exists a shift $s \in \{0, 1, 2, \ldots, n\}$ such that $G^s$ has information defect $d(G^s, 2) > n/2$.

**Conjecture 2′** :

> Let $\epsilon > 0$. Let $G$ be a graph with $n$ vertex and less than $n^{1+\epsilon}$ edges. Then (provided $n$ is sufficiently large) there exists a shift such that $G^s$ has information defect $d(G^s, 2) > n/2$.

Sparse graphs might have a relatively high guessing number (i.e. allow a lot of coherence). However, intuitively having a high guessing number is a "sporadic" property and that in general gets destroyed when the graph is modified through shifts or permutations.

Despite this intuition, there is a limit to how much shifts can insure a drop in guessing numbers of a sparse graph. In the next section we will show that Conjecture 2′ is actually false and that it is possible to construct a sparse graph $G$ together with a labelling such that $G$ as well as all shifts $G^s$ have guessing number $> n/2$ and information defect $< n/2$.

As we already pointed out, Conjecture 2′ is in some sense the most important of Valiant's conjectures. Had it been correct, this would have had direct consequences for providing new lower bounds in circuit complexity. None of the other three conjectures are known to have such strong consequences. Never-the-less, the conjectures are interesting in their own right and we will reduce these conjectures to certain questions about guessing numbers.

The good news (for proving a non-linear, O(log(n))-depth lower bound for the shift problem) is that a simple modification of Valiant's Conjecture $2'$, does imply such a lower bound.

This follows (according to Theorem 2.2 in [7]) from [11] where the following proposition can be extracted:

**Proposition C** :

> For every $\epsilon > 0, c$ and $d$, there exists $K$ such that if $F$ (the shift function) can be computed by a circuit of size $cn$ and depth $d\log(n)$, then it can be computed by a graph $G$ of degree at most $n^{\epsilon}$ with $\frac{Kn}{\log(\log(n))}$ common bits.

# 9 Counter examples to Valiant's shift conjecture

We have restated Valiant four conjectures as propositions about the behaviours of the information defect under shift and permutations.

Consider the graph $G$ in Figure 1. Undirected edges represent bidirectional edges. Later this graph will be defined as $G_{9,3,1}$. The graph $G(= G^0)$ as well as its 8 shifted versions $G^1, G^2, \ldots, G^8$ are all isomorphic (disregarding the labelling). Each of the graphs $G^j$, $j = 0, 1, 2, \ldots, 8$ has guessing number 6 and has information defect 3.
In terms of Valiant's notions, the bipartite graph $B_G$ associated to $G$ has each shift realised by 3 common bits (the bipartite $B_G$ is the graph containing the 18 nodes $\{i_0, i_1, \ldots, i_8\}$ and $\{o_1, o_2, \ldots, o_8\}$ with an edge $(i_v, o_w)$ if and only if $(v, w) \in G$). Valiant's Conjecture suggest that at least $n/2$ bits (i.e. 5) common bits are needed for some shift.

The graph $G := G_{23,5,1}$ in Figure 2(a) is invariant under shifts (if we disregard the labelling). The graph in Figure 2(b) is a subgraph of $G$ and it has guessing number 14 and information defect 9. Thus the graph $G = G_{23,5,1}$ has guessing number $\geq 14$ and information defect $\leq 9$. In Valiant's terminology the bipartite graph $B_G$ associated to graph the graph $G = G_{23,5,1}$ has each of its 23 shifts realised by 9 (rather than at least 12 as suggested by the conjecture) common bits.

We now generalise these examples. Consider $n, d, r \in N$ with $k < n/2$ and $0 \leq r \leq n-1$ we let $G_{n,d,r}$ denote the graph that has vertex set $\{0, 1, 2, \ldots, n-1\}$ and edge set $E = E_{n,d,r}$ consisting of all edges $(i, j)$ with $i + j \in \{r, r+1, r+2, \ldots, r+d-1\}$ (where the sum is calculated modulo $n$).

**Lemma 5** :

> Let $n \in N$ be an odd number and let $k < n/2$. Then for each $r = 0, 1, 2, \ldots, n-1$ the graphs $G_{n,d,r}$ have guessing number $k(G, s) \geq \frac{n}{2} + \frac{d}{2}$ and information defect $b(G, s) \leq \frac{n}{2} - \frac{d}{2}$

**Proof:** Let $W = \{r, r+1, \ldots, r+d-1\}$. Let $n_0 := \frac{n+1}{2}$ and let $d_0 := \frac{d-1}{2}$. The graph $G_{n,d,r}$ contains $d$ "self-loops" since the number of elements $i$ with $i + i \in W$ is $k$ (since $n$ is

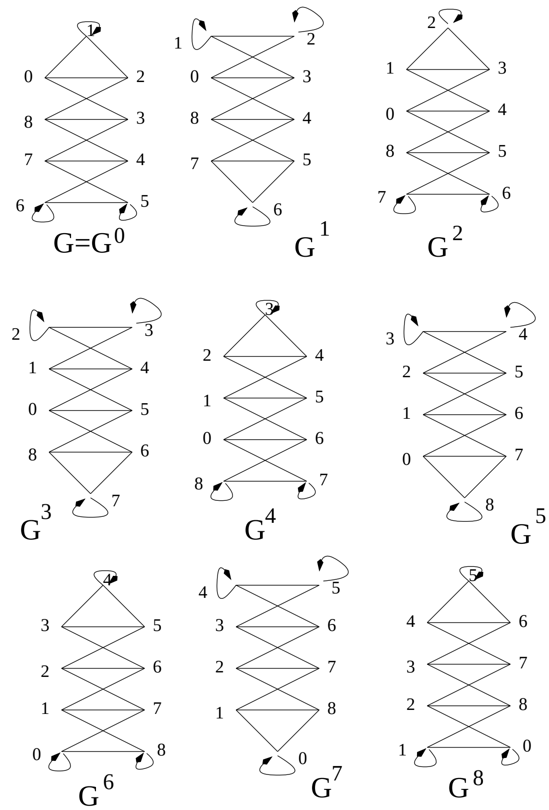


Figure 1: $G_{9,3,1}$ and its 8 shifted versions

odd). Assume first that $r = 2v$ is even. In this case the $d$ vertex $v, v+1, v+2, \ldots, v+d_0$ as well as $n_0+v, n_0+(v+1), \ldots n_0+(v+d_0-1)$ have self-loops. The edges $(v-i, v+d_0+i)$ and $(v+d_0+i, v-i)$ for $i = 1, 2, \ldots, n_0-d_0-1$ form a partitioning of the $n-d$ vertex that have no self-loop.

Thus $G_{n,d,r}$ contains a disjoint union of $\frac{n-d}{2}$ cliques of size 2, as well as $d$ vertex with self-loop. Thus $G_{n,d,r}$ has guessing number $k(G,s)$ at least $\frac{n}{2} + \frac{d}{2}$. The graph $G_{n,d,r}$ has information defect $b(G,s) \leq \frac{n}{2} - \frac{d}{2}$, since we can simply broadcast the messages $x_{v-i} + x_{v+d_0+i}$ for $i = 1, 2, \ldots, n_0-d_0-1$ through the public channel. ♣

Let $n \in N$ be an odd number and let $k \in N$ be an odd number. Let $\tilde{G}_{n,d}$ denote the class of graphs $G_{n,d,r}$ with $r = 0, 1, 2, \ldots, n-1$.

There are, of course, many "sparse" graphs with guessing number above $n/2$. The point of the graphs in $\tilde{G}_{n,d}$ considered in the previous lemma is that:

**Lemma 6** :

> The class $\tilde{G}_{n,d}$ is closed under shifts. More specifically, for any shift $s$, the graph $G_{n,d,r}$ shifted by $s$ is identical to $G_{n,d,r+s}$ (i.e. $G^s_{n,d,r} = G_{n,d,r+s}$).

**Proof:** Let $G \in \tilde{G}_{n,d}$ and let $s \in \{0, 1, 2, \ldots, n-1\}$ be a shift. The graph $G^s_{n,d,r}$ ($G_{n,d,r}$ with the heads of all edges shifted by $s$) is given by $G_{n,d,s+r}$ i.e. by a graph in $\tilde{G}_{n,d}$. ♣

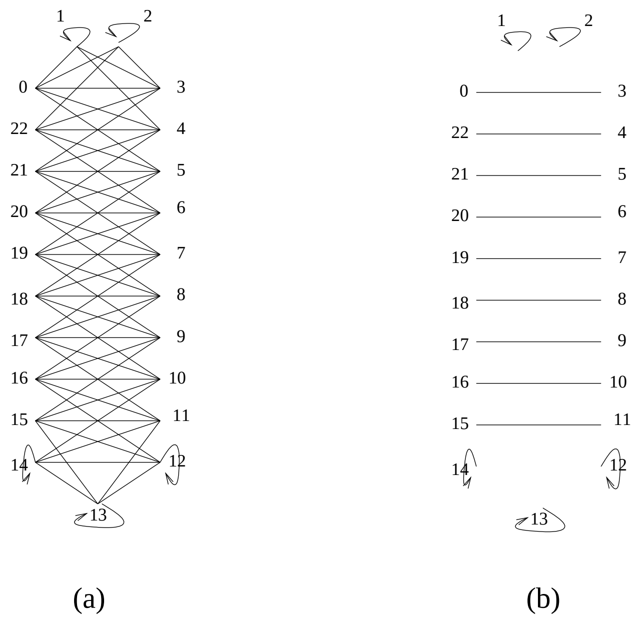


Figure 2: $G_{23,5,1}$ violating the conjecture

This shows that the information defect of the graphs $G \in \tilde{G}_{n,d}$ (and any shift $s$ of those) is at most $\frac{n}{2} - \frac{d}{2}$. For $1 > \epsilon > 0$ and for $d = n^{\epsilon}$ the graphs in $\tilde{S}$ are (for large values of $n$), sparse in the sense assumed in Valiant's conjectures. More specifically we have shown:

**Proposition 7** :

Let $\epsilon \in R$ with $0 < \epsilon < 1$ be any fixed real number. Let $G_n \in \tilde{G}_{n,n^{\epsilon}}$, $n = 1, 2, 3, \ldots$ be any sequence of graphs. Then each graph $G_n$ has (at most) $n^{1+\epsilon}$ edges and each graph $G_n$ as well as each of its shifts has information defect $b(G, s)$ with $b(G, s) \leq \frac{n}{2} - \frac{n^{\epsilon}}{2}$. This violates Valiant's Conjecture $2'$ that implies that for each sparse graph $G$ with $\leq n^{1+\epsilon}$ edges there always exists a shift $s$ such that $G^s$ has information defect at least $\frac{n}{2}$.

# 10 Modifying Valiant's conjectures

Valiant's conjecture 2 for graphs of in-degree (out-degree) bounded by 3 seems to be correct (except that we will have to change $\frac{n}{2}$ with $\frac{n-3}{2}$ for $n$ odd).

What about Conjecture 1 involving graphs of in-degree (out-degree) bounded by 3 and considering permutations instead of shifts? We will prove that this conjecture follows from the following statement:

**Statement** $S$ :

> Let $G$ be a directed graph with $n$ nodes and $dn$ edges (where $d \in N$). Assume that $G$ is loop-free and has no undirected edges. Then $G$ always has a guessing number $\leq \frac{n}{2} - \frac{n}{4d+2}$.

There is some evidence that Statement $S$ is valid, and it is tempting to upgrade statement $S$ to a conjecture. The main evidence for statement $S$ is that the 'clock' graphs $G_{clock}(2k+1,k)$ (with $2k+1$ nodes and out-degree $k$) have guessing number $k$. Statement $S$ is based on the assumption that the highest guessing number of a loopfree graph $G$ without undirected edges and with $\leq kn$ edges, is achieved by dividing the $n$ nodes into $\lfloor \frac{n}{2k+1} \rfloor$ disjoint clock graphs of type $G_{clock}(2k+1,k)$. This graph $G$ has guessing number close to $\frac{kn}{2k+1}$ (equality appears whenever $n$ is divisible by $2k+1$).

In the case where $k = 3$ we get graphs $G$ with guessing number $\frac{3n}{7}$. This value is $\leq \frac{n}{2} - 11$ for $n \geq 154$.

**Proposition 8** :

> Assume that Statement $S$ is valid. Then the following statements are valid:
>
> Statement $S_1$: Assume that $G$ is a graph with $n$ nodes, with no selfloops and no unoriented edges. Then $G$ has guessing number strictly less than $n/2$.
>
> Statement $S_2$: Assume that each node in $G$ has in-degree (out-degree) at most 3 and that $G$ does not contain selfloops or undirected edges and contain at least 154 nodes. Then $G$ has guessing number $\leq \frac{n}{2} - 11$.
>
> Statement $S_3$: Assume that $d \in N$ and that $G$ is a graph with $n$ nodes and $\leq dn$ edges. Further, assume that $G$ does not contain self-loops or undirected edges. Then $G$ has guessing number $\leq \frac{n}{2} - \alpha$ for $n \geq \alpha(4d+2)$.

The statement $S_2$ implies that Valiant's conjecture 1 is valid for $n \geq 175$, while statement $S_3$ for $d = n^{\epsilon}$ implies that Valiant's conjecture $1'$ is valid for $n \geq 4d^2(4d+3)$ which is always valid for $\epsilon << \frac{1}{4}$ and $n \geq 4$.

Proposition 9 is a special case of Proposition 10. Though we could strictly speaking avoid proving Proposition 9 (since it can be derived from Proposition 10) the proof idea is somewhat more transparent in the proof of Proposition 9 (since the proof contains fewer parameters).

**Proposition 9** :

> Statement $S_2$ implies that each graph $G$ with $n \geq 175$ nodes satisfies Valiant's conjecture 1.

**Proof:** First we show that statement $S_2$ implies that for each graph with $n \geq 175$ nodes and in-degree (out-degree) $\leq 3$ there exists a permutation $\pi$ of $n$ such that the graph $G^\pi$ has guessing number $< n/2$. Let $G$ be a graph for which each node has in-degree $\leq 3$. Assume that $G$ has $n \geq 175$ nodes. The number of edges in $G$ is $\leq 3n$. Some of the edges might form a loop. Two pairs of edges might form an undirected edge. Let $\pi \in S_n$ be a randomly selected permutation (i.e. assume that each permutation is selected with the same probability $p = \frac{1}{n!}$). We want to calculate the expected numbers of loops in $G^\pi$ and the expected number of undirected edges in $G^\pi$. The nice thing about calculating expectation numbers is that we need not worry about dependence and independence issues. Any edge $(a, b)$ becomes a loop exactly for permutations with $\pi(b) = a$. Thus the expected number of loops produced by a single edge is $\frac{1}{n}$. The expected number of loops produced by $e$ edges is $\frac{e}{n}$. Thus the expected number of loops in the graph $G^\pi$ is exactly $\frac{e}{n} \leq \frac{3n}{n} = 3$. The expected number of undirected edges can be calculated as follows: For two edges $(a, b)$ and $(c, d)$ with $a, b, c$ and $d$ distinct they produce an undirected edge exactly when $(a, \pi(b))$ and $(c, \pi(d))$ have $a = \pi(d)$ and $c = \pi(b)$. The expectation value of this is $\frac{1}{n(n-1)}$. If $b = d$ or if $a = c$, $a = \pi(d)$ and $c = \pi(b)$ are not possible if $(a, b) \neq (c, d)$. If $a = c$ and $b = d$ (i.e when we have two loops $(a, a)$ and $(c, c)$), they produce an undirected edge exactly when $\pi(a) = c$ and $\pi(c) = a$. The expectation value for this to happen is $\frac{1}{n(n-1)}$. Thus if the number of all pairs $(a, b)$ and $(c, d)$ of edges in $G$ with $b \neq d$ and $a \neq c$, is $f$, the expected number of undirected edges in $G^\pi$ is $\frac{f}{n(n-1)}$. The maximum number of such pairs of edges is bound by $(3n)^2 = 9n^2$. Thus the expected number of undirected edges in $G^\pi$ is bound by $\frac{9n^2}{n(n-1)} \leq 9 + \frac{9}{n} < 10$. And the expected number of nodes that are involved in a loop or are one of the points in a undirected edge is $\leq 3+2(9+\frac{1}{n}) < 22$. Thus there must exist a permutation $\pi$ such that 21 or less points are involved with self loops or undirected edges. The graph $G^\pi$ restricted to those 21 points might be a complicated subgraph with a mixture of undirected edges and directed edges. The highest contribution these points can make to the guessing number is 21 (if each of the 21 point, against our expectations, contains a self-loop ). The guessing number never decreases when more edges are added so we can - without loss of generality - assume that the 21 points have self loops (producing a guessing number of 21). Now we restrict the graph $G^\pi$ to the remaining $n - 21$ nodes. This graph has $\geq 175 - 21 = 154$ nodes, it has in-degree $\leq 3$ and has no self loops and no undirected edges. Thus according to our conjecture, it has guessing number $\leq \frac{n-21}{2} - 11$. From this it is not hard to show that $G^\pi$ has guessing number $\leq \frac{n-21}{2} - 11 + 21 \leq \frac{n-1}{2} < \frac{n}{2}$.

But since $G^\pi$ has guessing number $k(G^\pi, s) < n/2$ and since $k(G^\pi, s) + d(G^\pi, s) \geq n$ (Lemma 3), the information defect $d(G^\pi, s)$ is at least $n - k(G^\pi, s) > \frac{n}{2}$ thus proving Valiant's conjecture 1. ♣

**Proposition 10** :

Assume that statement $S_3$ is valid in general. Let $G$ be a directed graph $G$ with $n \geq 4d^2(4d + 3)$ nodes and $\leq dn$ edges. Then there exists a permutation $\pi \in S_n$ such that $G^\pi$ has guessing number $< \frac{n}{2}$. In particular, Valiant's Conjecture $1'$ is valid for each $n > 5$ and $\epsilon < \frac{1}{4}$

**Proof:** Assume that $G = (V, E)$ is a graph with $|V| = n \geq (4d^2)(4d+3)$ and $\leq dn$ edges. We want to show (using statement $S_3$) that there there exists a permutation $\pi$ of $n$ such that the graph $G^\pi$ have guessing number $< n/2$. Some of the edges in $G$ might form a loop. And two pairs of edges in $G$ might form an undirected edge. We want to show that there exists a permutation $\pi \in S_n$ such that $G^\pi$ has very few selfloops and undirected edges.

Let $\pi \in S_n$ be a randomly selected permutation (i.e. assume that each permutation is selected with the same probability $= \frac{1}{n!}$). We want to calculate the expected number of loops in $G^\pi$ and the expected number of undirected edges in $G^\pi$. Any edge $(a, b)$ becomes a loop exactly for permutations with $\pi(b) = a$. Thus the expected number of loops produced by a single edge is $\frac{1}{n}$. The expected number of loops produced by $e$ edges is $\frac{e}{n}$. Thus the expected number of loops in the graph $G^\pi$ is exactly $\frac{e}{n} \leq \frac{dn}{n} = d$. The expected number of undirected edges can be calculated as follows: For two edges $(a, b)$ and $(c, d)$ with $a, b, c$ and $d$ distinct they produce an undirected edge exactly when $(a, \pi(b))$ and $(c, \pi(d))$ have $a = \pi(d)$ and $c = \pi(b)$. The expectation value of this is $\frac{1}{n(n-1)}$. If $b = d$, or if $a = c$, $a = \pi(d)$ and $c = \pi(b)$ are not possible (when $(a, b) \neq (c, d)$). If $a = c$ and $b = d$ (i.e when we have two loops $(a, a)$ and $(c, c)$), the two edges produce an undirected edge exactly when $\pi(a) = c$ and $\pi(c) = a$. The expectation value for this to happen is $\frac{1}{n(n-1)}$. Thus if the number of all pairs $(a, b)$ and $(c, d)$ of edges in $G$ with $b \neq d$ and $a \neq c$, is $f$, the expected number of undirected edges in $G^\pi$ is $\frac{f}{n(n-1)}$. The maximum number of such pairs of edges is bound by $(dn)^2 + 2\frac{d^2}{n} + 2\frac{d^2}{n^2-n} \leq 3d^2$ (for $n \geq 3$). Thus the expected number of undirected edges in $G^\pi$ is bound by $3d^2$. The expected number of nodes that are involved in a loop or in an undirected edge is $\leq 3d^2$. Thus there must exist a permutation $\pi$ such that $3d^2+d \leq 4d^2$ or less points are involved with selfloops or undirected edges. The graph $G^\pi$ restricted to those $4d^2$ points might be very complicated with a mixture of undirected edges and directed edges. The highest contribution these points can make to the guessing number is $4d^2$ (if each of the $2d^4$ point, against our expectations, contains a selfloop). The guessing number never decreases when more edges are added so we can - without loss of generality - assume that the $4d^2$ points have self-loops (producing a guessing number of $4d^2$). Now we restrict the graph $G^\pi$ to the remaining $n' := n - 4d^2$ nodes. This graph has $\geq dn = d'n' = \frac{dn}{n'}n'$ edges, has no self loops and has no undirected edges. Thus according to Statement $S_3$, it has guessing number $\leq \frac{n'}{2} - \alpha'$ for $n' \geq \alpha'(4d' + 2)$.

The graph $G^\pi$ has then guessing number $< \frac{n'}{2} - \alpha' + 4d^2$ when $n \geq \alpha'(4d' + 2) + 4d^2$. In other words $G^\pi$ has guessing number $k(G, s)$ strictly less than $\frac{n}{2} - \alpha' + 4d^2$ when $n \geq \alpha'(4d+2) + 4d^2$. Thus if we let $\alpha' = 4d^2$, we have $k(G^\pi, s) < \frac{n}{2}$ for $n \geq 4d^2(4d+2) + 4d^2 = 4d^2(4d+3)$.

But the information defect $d(G^\pi, s) \geq n - k(G, s)$ and therefore $d(G^\pi, s) > n/2$ for $n \geq 4d^2(4d+3)$. ♣

## References


[1] R. Ahlswede, N. Cai, N. Li, and R.W. Yeung. An algebraic approach to network coding. page 104, 2001.

[2] R. Dougherty, C. Freiling, and K. Zeger. Linearity and solvability in multicast networks. *IEEE Transactions on Information Theory*, 50(10):2243–2256, December 2004.

[3] R. Dougherty, C. Freiling, and K. Zeger. Induffiency of linear coding in network information flow. *IEEE Transactions on Information Theory*, 51(8):2745–2759, August 2005.

[4] T. Ho, D. Karger, M. Merard, and Koetter R. Network coding from a network flow perspective. In *Proocedings of the ISIT 2003, Yokohama, Japan*, July 2003.

[5] R. Koetter and M. Medard. Beyond routing: An algebraic approach to network coding. In *Proceedings of the 2002 IEEE Infocom*, 2002.

[6] C.K. Ngai and R.W. Yeung. Multisource network coding with two sinks. In *International Conference on Communications, Circuits and Systems (ICCCAS)*, June 2004. Best Paper Award, Communication Theory.

[7] P. Pudlak, V. Rodl, and J. Sgall. Booean circuits, tensor ranks, and communication complexity. *SIAM Journal of Computing*, 26(3):605–633, June 1997.

[8] S. Riis. Linear versus non-linear boolean functions in network flow. In *38th Annual Conference on Information Science and Systems (CISS), Princeton, NJ, March 2004*.

[9] S. Riis. *Utilizing public information in Network Coding (appendix)*, pages 851–882. In: General Theory of information Transfer and Combinatorics, Springer 2006. Editors: Ahlswede, Baumer, Cai, Aydinian, Blinovsky, Deppe, Mashurian.

[10] S. Riis. Utilizing public information in information networks. In *Technical report*, June 2005.

[11] L. Valiant. Graph-theoretic arguments in low-level complexity. In *Lecture Notes in Computer Science*, volume 53, pages 162–176. Springer, 1977.

[12] L. Valiant. Why is boolean circuit complexity theory difficult? In M.S. Pattorson, editor, *Springer Lecture Series*, pages 84–94, 1992.

[13] R.W. Yeung. Multilevel diversity coding with distortion. *IEEE Trans. Inform. Theory*, 41:412–422, March 1995.

[14] R.W. Yeung and Z. Zhang. Distributed source coding for satellite communications. *IEEE Trans. Inform. Theory*, (IT-45):1111–1120, 1999.