LABORATORY FOR COMPUTER SCIENCE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

MIT/LCS/TM-196

ON CONCENTRATION AND CONNECTION NETWORKS

Sandeep Nautam Bhatt

March 1981

545 TECHNOLOGY SQUARE, CAMBRIDGE, MASSACHUSETTS 02139

ON CONCENTRATION AND CONNECTION NETWORKS

Sandeep Nautam Bhatt

May 1980

Keywords: Switching Networks, Rearrangeable concentrators, Expanders, Superconcentrators, Probabilistic constructions, Incrementally Non-blocking connectors

ON CONCENTRATION AND CONNECTION NETWORKS

by

Sandeep Nautam Bhatt

Submitted to the Department of Electrical Engineering and Computer Science

on May 15, 1980

in partial fulfillment of the requirements for the

Degree of Master of Science

ABSTRACT

This thesis deals with the structural complexity of switching networks which realize concentration and connection requests when operated in a rearrangeable or incremental manner. Some of the important results and constructions are briefly reviewed. On the basis of non-constructive proof techniques used to obtain linear upper bounds on the complexity of rearrangeable concentrators, it is shown that not only are certain random graphs very likely to be rearrangeably non-blocking concentrators, but that if a randomly constructed graph is not non-blocking, then, on the average, only a constant number of edges need be added to the graph to make it non-blocking. Although the problem of recognizing non-blocking networks appears to be a computationally hard problem, the extra edges may be added to the graph efficiently, during operation of the network. Finally, we obtain a constructive as well as an improved non-constructive upper bound on the complexity of incrementally non-blocking connection networks.

Supervisor : Ronald L. Rivest
Title : Associate Professor of Computer Science

ACKNOWLEDGEMENTS

My deep gratitude to Professor Ronald L. Rivest for his constant support and excellent guidance throughout the period of this research. I am also indebted to my professors who introduced and inspired me to study the methods of computational complexity.

Many thanks to Gail Kinburn for her willing and efficient help in preparing this manuscript.

This work was supported in part by NSF Grants MC 78-05 849 MCS A01 and MCS 7719754.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

Switching networks are used to establish various combinations of interconnection assignments between a specified set of network inputs and a disjoint set of network outputs. For different purposes, different types of interconnection assignments between inputs and outputs are required. The design of switching networks for different types of interconnection assignments is central to the design of switched line communication systems, such as in telephone, telecommunication, and computer systems. In many applications, the associated switching network contributes significantly to the cost and performance of the overall system. The need for optimal switching networks has been the primary motivation for research in the design and analysis of switching networks.

A switching network may loosely be described as a set of components consisting of contact switches and disjoint sets of input and output terminals, joined together by links. The interconnection of components establishes various combinations of simultaneous routes connecting inputs to outputs.

A signal entering an input is routed to an output via a series of switches, with consecutive switches joined by links. For the purposes of this thesis we may assume that each link allows at most one signal to pass through at a time. The state of the network at any given instant is described by the state of its switches; a switch is on if a signal is currently being sent through it, otherwise it is off. The state in which all switches are off is called the idle state. An input (output) is said to be busy if it is sending (receiving) a signal, otherwise it is idle. An input may send at most one signal at a time, but an output may receive any number of signals simultaneously.

Whenever a signal is ready to be sent from an input terminal, a request for connection is made. If the connection requested can be established by connecting the corresponding input to an appropriate output via a series of switches that are in the off position, then the request is said to be realizable. Once such a connection has been found, all the switches on the route are turned on. If no such connection exists between the input and an appropriate output, the request is said to be blocked.

Although many different types of requests are possible, only the following two will be considered here:

1) Connect the input to any idle output.
2) Connect the input to a specified idle output.

Furthermore, for any network, we restrict all requests to be of the same type. For instance, if the set of outputs consists of a set of equivalent devices, each supplying identical service to at most one user at a time, and the inputs correspond to users requesting that service, then it is immaterial which set of outputs any set of inputs is connected to, as long as each user is connected to a distinct output and the number of users requesting service simultaneously does not exceed the number of outputs. In this case all requests may clearly be restricted to be of type 1. Networks which establish connections between inputs and outputs for requests of type 1 exclusively are called concentration networks. Next, if no two outputs provide identical service then each input must specify which output it requests service from. In addition, if each input may request service from at most one output at a time, then requests may be restricted to be of type 2. Networks which are used to realize such requests exclusively are called connection networks. An example is a network for a telephone system where each input (caller) requests connection to a specific output (called party).

Associated with each network are two modes of operation for handling sets of requests for connection. These are analogous to "off-line" and "on-line" computations.

1) Rearrangeable: In this mode all requests are assumed specified and a network state realizing them is sought. The case where requests come in at different times is conceptually easy to handle. Each time a new request comes in, we have a new set of requests; consequently, we seek a state realizing this new set of requests. Thus, realizing an incoming request might involve rerouting previous connections, if necessary.

2) Incremental: In this case, requests may arrive at any time and each incoming request must be realized without disturbing busy routes established for realizing previous requests. If a request cannot be realized without disturbing busy routes, it is said to be blocked.

The topology of the network, the mode of operation, and the application (concentration, connection, etc.) together determine the set of sequences of requests which can be realized by the network. For rearrangeable operation the sequential ordering of requests is irrelevant. A sequence of requests of type 1 (2) is said to be consistent with respect to a mode of operation if there exists a concentrator (connection) network which, operated in the same mode can realize all requests in the sequence simultaneously. For example, consider two requests of type 2 in which different inputs request connection to the same output. Clearly, no rearrangeable or incremental connection network can realize both requests simultaneously. Thus the two requests form an inconsistent sequence for both modes of operation. Similarly, a set of requests of type 1 is inconsistent in either mode if and only if the number of requests exceeds the total number of outputs or if more than one request for connection is made by any input. It is easy to see that the number of distinct, consistent sets of requests for rearrangeable concentrators and connectors with n inputs and m outputs are $\binom{n}{m}$ and $[n]_m$ respectively. If the network is such that any consistent sequence of requests can be realized, the network is said to be non-blocking. Although it is desirable to have non-blocking networks whenever possible, sometimes we can construct much smaller networks which we cannot easily prove to be non-blocking, but which are non-blocking with a very high probability. Such networks also may be of practical value. The capacity of a network, of a given type, operating in a specified mode, is defined to be the largest integer k such that all sequences of k or less requests of the same type are realizable in that mode. This thesis is concerned primarily with the construction of rearrangeable concentrators and incremental connectors which, with very high probability, are non-blocking. Finally, note that a non-blocking connection network operating in any mode is, trivially, also a non-blocking concentrator network in the same mode, and that any non-blocking incremental network of any type is also a non-blocking rearrangeable network of the same type. However, the converse statements are, in general, not true.

1.1 A FORMAL MODEL

A Switching Network is formally defined to be a directed, acyclic graph. The set of inputs of the network consists of the vertices of the graph which have indegree 0. Vertices with outdegree 0 form the set of outputs of the network. Each vertex has non-zero total degree; thus the sets of inputs and outputs are disjoint. Associated with each network is a set of states. A state is a function which assigns to each edge of the graph either the value on or the vaue off. The set of states associated with a network consists of exactly those states which satisfy the property that the set of edges whose value is on forms a set of directed, vertex-disjoint paths, each path beginning at an input vertex and terminating at an output vertex. Furthermore, each such path is said to connect the corresponding input and output vertices.

The correspondence between the above definitions and the previous informal description is straightforward. Edges in the graph represent switches in the network and vertices with non-zero outdegree and indegree represent links between switches. The restriction that paths of edges that are on be vertex-disjoint corresponds to the restriction that each link can allow at most one signal to pass through at a time. Thus, if two edges emerge from a vertex of a concentrator or a connection network, they cannot both be on in the same state.

We now give graph-theoretic definitions of some of the networks relevant to this study:

1) A rearrangeable (n,m,k)-concentrator is a graph with n inputs and m outputs with the property that any r inputs, $r \leq k$, can be connected to some r outputs via r vertex-disjoint paths.

2) A rearrangeable (n,m,k)-hyperconcentrator is a network with n inputs and m outputs, $(1,\ldots,m)$, with the property that every set of r inputs, $r \leq k$, can be connected to outputs $1,\ldots,r$ via r vertex-disjoint paths.

3) A rearrangeable (n,m,k)-superconcentrator is a network with n inputs and m outputs with the property that every set of r inputs $r \leq k$, can be connected to every set of r outputs via r vertex-disjoint paths.

We will restrict attention to the case $m = \alpha n$, where α is a constant, $\alpha < 1$, to avoid triviality. From these definitions it is apparent that every (n,m,k)-superconcentrator is also an (n,m,k)-hyperconcentrator, and that every (n,m,k)-hyperconcentrator is also an (n,m,k)-concentrator. The converse statements are, however, not true in general. In future, if $m = k$ we shall refer to the network as an (n,m)-concentrator. If $n = m = k$ for a superconcentrator or hyperconcentrator, we shall refer to them as n-superconcentrators and n-hyperconcentrators respectively. Since we will be concerned only with rearrangeable concentrators, the term rearrangeable will frequently be omitted.

1.2 THE COMPLEXITY OF SWITCHING NETWORKS

Consider two networks, A and B, which realize identical sets of requests when operated in a certain mode. Also, suppose that A has fewer contact switches than B, and that the time required to determine switch settings in A to realize an arbitrary set of requests is not significantly more than in B. In this case, A is preferable to B for the particular mode of operation. In general, we are intersted in minimizing the number of switches in a network without reducing its capacity, as well as minimizing the time required to determine switch settings for realizing arbitrary sets of requests. The structural complexity of a network is defined to be the number of edges in the associated graph. The operational complexity of a family of networks of different sizes is defined to be the minimum worst-case time complexity over all algorithms which, given an arbitrary network in the family and an arbitrary, consistent, set of requests, determine a state realizing the set of requests. The time taken by an algorithm to find a state realizing a given consistent set of requests is, informally, proportional to the sum, over all edges, of the total number of times an edge is examined by the algorithm.

Although this thesis is concerned primarily with the construction of a family of concentrators whose structural complexity is minimized asymptotically, we will comment on the operational complexity of the same. In general, there appears to be a tradeoff between the two measures.

1.3 OVERVIEW OF THE THESIS

In the next chapter asymptotic upper and lower bounds on the structural complexity of rearrangeable and incremental concentrator and connection networks is reviewed. We also consider some attempts at explicitly constructing a family of expander graphs which may be used to construct superconcentrators with complexity linear in the number of inputs in the networks, and develop a simple yet general technique for proving why these attempts fail. In Chapter 3 a different approach, that of constructing graphs at random and using them, if they satisfy certain specified properties, to construct non-blocking networks is examined. The problem of determining whether such a randomly constructed graph satisfies the specified property appears to be a computationally hard problem whose complexity is open. However, it is shown that not only are the randomly constructed graphs very likely to satisfy the required properties, but also that if a randomly constructed graph does not satisfy them, then, on the average, very few edges need be added to a graph to make it satisfy the required properties. The extra edges may be added during the operation of the network without adding significantly to the operational complexity of the network. We also derive constructive as well as nonconstructive upper bounds on the structural complexity of incrementally non-blocking connection networks. Finally, some of the open problems are mentioned.

CHAPTER 2: BACKGROUND

In this chapter we review some of the results concerning non-constructive and constructive asymptotic upper and lower bounds on the structural complexities of rearrangeably non-blocking concentrators, superconcentrators, and related graphs. We also consider some attempts at explicitly constructing a family of linear size expander graphs which may be used to construct linear size superconcentrators, and develop a simple yet general technique for proving why these attempts fail.

2.1 BASIC CONSTRUCTIONS

Probably the most commonly used switching network is the complete crossbar switch which is represented as a complete bipartite graph with n inputs, m outputs, and nm edges. An example with $n = 5$, $m = 3$ is shown in the Figure 1. It is readily seen that such a graph is an incrementally non-blocking connection network; if, in any state, a request of type 2 appears, simply switch <u>on</u> the edge between the specified idle input and idle output. It follows that a complete crossbar switch with n inputs and m outputs is also an incrementally non-blocking (n,m,m)-concentrator, (n,m)-superconcentrator, etc. Assuming $m = \alpha n$, where α is a constant, $0 < \alpha < 1$, the structural complexity of an n-input, αn-output complete crossbar switch is αn^2, thus establishing an asymptotic upper bound of $O(n^2)$ on all switching networks considered in this thesis. Moreover, the operational complexity of a crossbar switch is $O(n)$, since exactly r edges need be examined to realize a consistent sequence of r requests. This is clearly optimal. In the rest of this thesis we will be concerned essentially with the problem of constructing networks which minimize the asymptotic structural complexity for different types of networks, at the cost of increasing the operational complexity.

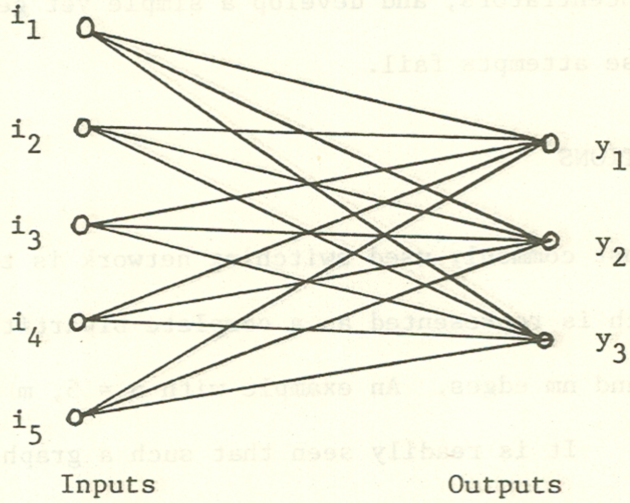


FIGURE 1.

Another classical switching network is the permutation network; a typical example is shown in Figure 2. It consists of n inputs $i_1,\ldots,i_n$, n outputs $y_1,\ldots,y_n$, and has the property that if π is any permutation over the set $\{1,\ldots,n\}$, then there is a state of the network in which, for each k input i_k is connected to output $y_{\pi(k)}$. In such state, π is said to be realized by the network. Networks with O(nlogn) edges realizing all permutations over {1,...,n} have been constructed independently by several researchers. The network below appears in Waksman [8]. The permutation network on n inputs and outputs is constructed recursively by using A and B which are permutation networks on $\frac{n}{2}$ inputs and outputs, and connecting them as shown.

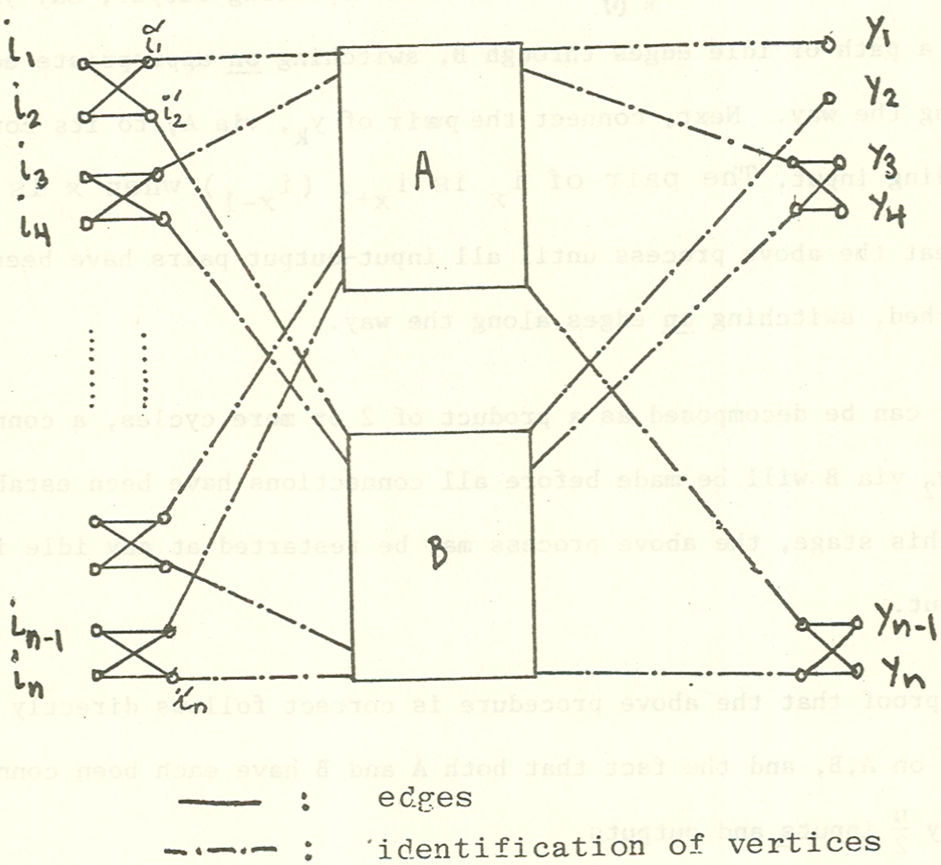


FIGURE 2.

To prove that this construction realizes all permutations over {1,...,n} consider the following informal algorithm (from [8]) which, given π, finds a state of the network realizing π.

1) Initially, let the network be in the idle state.

2) Connect y_1 to $i_{\pi^{-1}(1)}$ via A. If $\pi^{-1}(1)$ is even, then the edge $(i_{\pi^{-1}(1)}, i'_{\pi^{-1}(1)-1})$ is switched on, or else the edge $(i'_{\pi^{-1}(1)}, i_{\pi^{-1}(1)})$ is switched on.

3) Connect the pair of $i_{\pi^{-1}(1)}$ to its corresponding output, say y_k, via a path of idle edges through B, switching on appropriate edges along the way. Next, connect the pair of y_k, via A, to its corresponding input. The pair of i_x is i_{x+1} (i_{x-1}) when x is even (odd).

4) Repeat the above process until all input-output pairs have been matched, switching on edges along the way.

5) If π can be decomposed as a product of 2 or more cycles, a connection to y_2 via B will be made before all connections have been established. At this stage, the above process may be restarted at any idle input or output.

The proof that the above procedure is correct follows directly by induction on A,B, and the fact that both A and B have each been connected to exactly $\frac{n}{2}$ inputs and outputs.

Letting P(n) denote the minimum number of edges sufficient to construct a permutation network on n inputs and outputs, we have, from the construction given:

$$P(n) \leq 2P(\tfrac{n}{2}) + 4n - 2 \quad .$$

Also, P(1) = 1, which implies $P(n) \leq 4\ n\log_2 n$.

A similar construction based on dividing the network recursively into three parts instead of two yields:

$$P(n) \leq \frac{6}{\log_2 3}\ n\log n - 3n$$

or,

$$P(n) \leq 3{\cdot}79\ n\log_2 n - 3n$$

Since a consistent set of r requests of type 2, $r \leq n$, induces a set of (n-r)! permutations, each of which realizes the set of requests, a permutation network is clearly a rearrangeably non-blocking connection network. This means that the structural complexity of rearrangeably non-locking connection networks is O(nlogn). In the next section we will see that the structural complexity of such networks is also Ω(nlogn); hence the construction given is asymptotically optimal. Note, however, that the operational complexity of the permutation network is O(nlogn).

If all but the first m outputs are eliminated from the previous figure, along with their associated edges, the resulting graph is a rearrangeably non-blocking (n,m)-connector and hence an (n,m,m)-superconcentrator. This means that the structural complexity of each type of rearrangeably non-blocking (n,m,m)-concentrators is O(nlogn). Later we will discuss how to construct rearrangeably non-blocking concentrators with O(n) edges, but with an additional asymptotic increase in operational complexity.

2.2 LOWER BOUNDS

For both concentrators as well as connectors with n inputs and m outputs, $\Omega(n)$ is a trivial lower bound on the number of edges required.

In rearrangeably non-blocking connection networks, the number of realizable sets of requests is at most equal to the number of states associated with the network. Furthermore, the total number of states in a network with e edges is no more than 2^e, since each edge can be in either of two states, on or off. Thus, for a rearrangeably non-blocking connector with n inputs and outputs,

$$2^e \geq \text{number of distinct states}$$
$$\geq \text{number of distinct sets of requests}$$
$$= \sum_{i=0}^{n} [n]_i$$

Thus, $e \geq \log_2 n!$, or $e = \Omega(n\log n)$.

Thus we have a lower bound of $\Omega(n\log n)$ for rearrangeably and hence, also for incrementally non-blocking connection networks. Finally, for incrementally non-blocking $(n,\alpha n,\beta n)$-concentrators the following theorem proved by Pippenger [6] establishes a lower bound of $\Omega(n\log n)$.

Theorem (Pippenger): Any incrementally non-blocking $(n,\alpha n,\beta n)$-concentrator G, where α, β are constant such that $0 < \beta < \alpha < 1$, has $\Omega(n\log n)$ edges.

Proof: Let S be a state in which m inputs, $m \geq \beta n$, are connected to m outputs via the paths $\sigma_1, \ldots, \sigma_m$, such that:

1) A request for connection from input x is blocked in S.

2) If any connection in S is disestablished, i.e., all edges along σ_i, for some i, $1 \leq i \leq m$, are switched off, then a request for connection from input x is realizable via path σ'_i.

We are guaranteed to find a state S and input x satisfying the above conditions since the number of inputs in the network is greater than the number of outputs.

Now, from (2) it follows that each busy path σ_i in S must have at least one vertex in common with the path σ'_i. Since, in any state, busy paths are vertex-disjoint, it follows that the σ'_i are distinct. Thus, there are at least m distinct paths starting at x and ending at some output vertices.

Next, if, for any i, $1 \leq i \leq m$, vertices on the path σ'_i are removed along with their associated edges, the remaining network G'_i must be incrementally non-blocking with capacity $\beta n-1$. Otherwise, in some state T′ of G'_i assume that fewer than $\beta n-1$ connections exist and that some request is blocked. Then in the state T of G in which the set of busy edges includes exactly the busy edges in T′, plus the edges in σ'_i, there are fewer than βn connections, and at least one request is blocked. But, this contradicts the fact that G has capacity βn.

Let E(n) denote the minimum number of states in any incrementally non-blocking $(n,\alpha n,\beta n)$-concentrator. Then, for each i, G'_i has at least E(n-1) states. By the preceeding argument, G then must have at least mE(n-1) states.

Thus, number of states in $G \geq m\,E(n-1) \geq \beta n\,E(n-1)$.
Since $E(1) = 1$ it follows that $E(n) \geq (\beta n)!$

Thus, if the number of edges in G = e, we have

$$2^e \geq E(n) \geq (\beta n)!$$

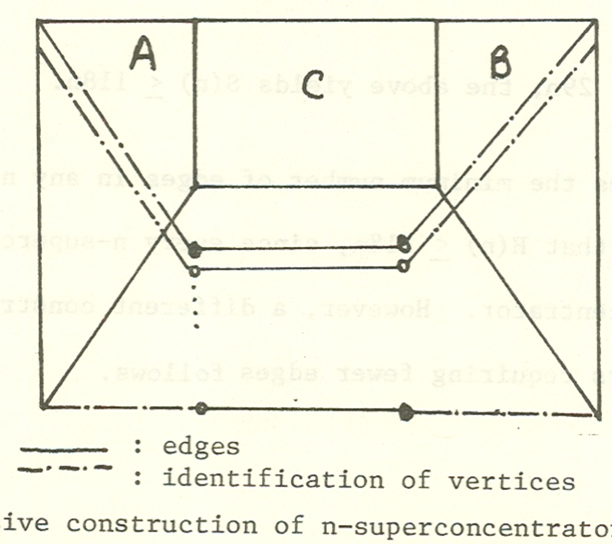
or, $e = \Omega(n\log n)$. Q.E.D.

2.3 LINEAR UPPER BOUNDS ON REARRANGEABLE CONCENTRATORS

Pinsker [4] used a non-constructive combinatorial argument to prove the existence of $(n,\alpha n,\beta n)$-concentrators with O(n) edges, where α and β are constants such that $0 < \beta < \alpha < 1$. Using these concentrators and a relatively complicated construction he showed that $(n,\alpha n)$-concentrators require at most 29n edges.

Valiant [7] first defined n-superconcentrators and n-hyperconcentrators to study the possibility of establishing non-linear lower bounds on the complexity of certain problems computed most efficiently, i.e., to within a constant factor, by straight-line programs. Straight-line programs may be represented as directed, acyclic graphs with vertices representing operators and an edge from vertex ν_1 to vertex ν_2 signifying that the output of the operator corresponding to ν_1 is an input to the operator corresponding to ν_2. If all operators are restricted to be binary, the number of edges in a straight-line program graph equals twice the number of instructions executed by the program. Now, the graph of any straight-line program to compute the convolution of polynomials of degree n-1 is an n-superconcentrator. Similarly, the graph of any straight-line program to compute the Discrete Fourier Transform of an n vector is an n-hyperconcentrator. Thus, the structural complexities of n-superconcentrators and n-hyperconcentrators provide lower bounds on the computational complexities of the two problems. However, Pinsker's result may be used to establish an O(n) upper bound on the structural complexity of n-superconcentrators and n-hyperconcentrators; consequently, as noted in [7], this approach does not yield non-linear lower bounds on the complexity of the two problems.

We now show that the structural complexity of n-superconcentrators and n-hyperconcentrators is linear in n. The following graph shows how an n-superconcentrator may be constructed recursively, using $(n,\frac{n}{2})$-concentrators.



Recursive construction of n-superconcentrator using $\frac{n}{2}$ -superconcentrator C and $(n,\frac{n}{2})$-concentrators A,B.

Proof of Construction: Given any set of r inputs and any set of r outputs, there are two cases to consider.

Case (1): $r \leq \frac{n}{2}$. In this case the r inputs and r outputs can be connected via r edges of the concentrators A and B to r inputs and r outputs of the $\frac{n}{2}$-superconcentrator C. By induction since C is a superconcentrator the r inputs and r outputs can be connected via r vertex-disjoint paths.

Case (2): $r > \frac{n}{2}$. In this case at least $(r - \frac{n}{2})$ inputs and $(r - \frac{n}{2})$ outputs can be connected one to one by edges in the identity map. This leaves at most $\frac{n}{2}$ connections to be established, thus reducing to Case 1 above. Q.E.D.

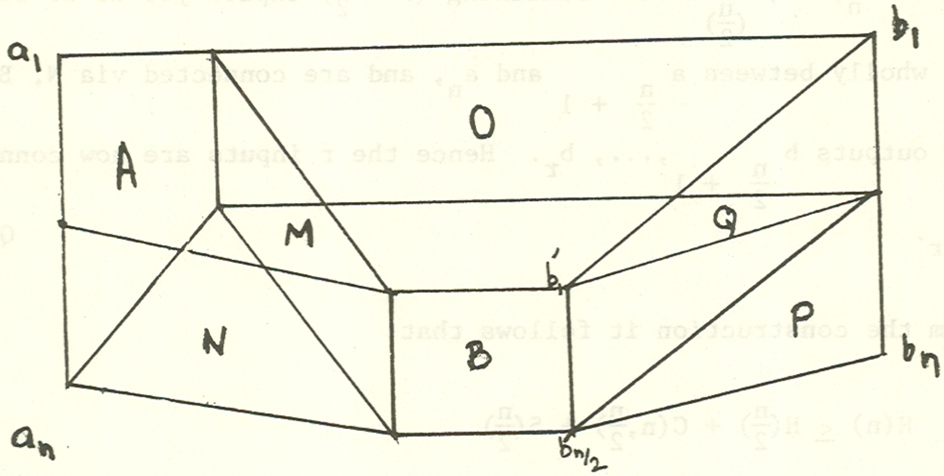
Letting $S(n)$ and $C(n,\frac{n}{2})$ denote the minimum number of edges required to construct n-superconcentrators and $(n,\frac{n}{2})$-concentrators respectively, we have:

$$S(n) \leq S(\tfrac{n}{2}) + 2\,C(n,\tfrac{n}{2}) + n$$

Also, $S(1) = 1$.

Since $C(n,\frac{n}{2}) < 29n$, the above yields $S(n) \leq 118n$.

If $H(n)$ denotes the minimum number of edges in any n-hyperconcentrator, it follows that $H(n) \leq 118n$, since every n-superconcentrator is also an n-hyperconcentrator. However, a different construction for n-hyperconcentrators requiring fewer edges follows.



A is an $(n,\frac{n}{2})$-concentrator.

B is a recursively constructed $\frac{n}{2}$ -hyperconcentrator.

O and M are each identity maps from the outputs of A to $b_1,\ldots b_{\frac{n}{2}}$ and to the inputs of B respectively.

N is an identity map from $a_{\frac{n}{2}+1}, \ldots, a_n$ to inputs of B.

Q and P are identity maps from outputs of B to $b_1,\ldots,b_{\frac{n}{2}}$, and to $b_{\frac{n}{2}+1},\ldots,b_n$ respectively.

Proof of Construction: We have to show that any r inputs can be connected to the r outputs $b_1,\ldots,b_r$. There are two cases to consider:

Case (1): $r \le \frac{n}{2}$. Connect the r inputs through A and M to r inputs of B. By induction, these are connected to $b'_1,\ldots,b'_r$ by r-vertex-disjoint paths, which are connected to $b_1,\ldots,b_r$ via the identity map Q.

Case 2: $r > \frac{n}{2}$. In this case connect the top $\frac{n}{2}$ inputs via A and O to the outputs $b_n, \ldots, b_{(\frac{n}{2})}$. The remaining $(r - \frac{n}{2})$ inputs yet to be connected must lie wholly between $a_{\frac{n}{2}+1}$ and a_n, and are connected via N, B, and P to the outputs $b_{\frac{n}{2}+1}, \ldots, b_r$. Hence the r inputs are now connected to $b_1, \ldots, b_r$. Q.E.D.

From the construction it follows that:

$$H(n) \leq H(\tfrac{n}{2}) + C(n,\tfrac{n}{2}) + 5(\tfrac{n}{2})$$

Also, $H(1) = 1$, $C(n,\frac{n}{2}) < 29n$, implying:

$$H(n) < 63n .$$

This is a tighter bound on H(n) than the one in [7].

In Chapter 3 we will see that $S(n) \leq 40n$, which gives tighter bounds on both S(n) and H(n) than the ones obtained above. However, if reduced bounds on $C(n,\frac{n}{2})$ are obtained, the construction of n-hyperconcentrators above could yield tighter bounds on H(n) than those known for S(n). For example, if it were true that $C(n,\frac{n}{2}) \leq 16n$, we have, from the above, that $H(n) \leq 37n$, which is less than any known upper bound on S(n).

2.4 EXPLICIT CONSTRUCTION OF LINEAR CONCENTRATORS

We now turn to the problem of explicitly constructing a family of concentrators of different sizes whose structural complexity is bounded by a linear function of n, the number of inputs to the network. The proofs of upper bounds in the previous section were based on the result of Pinsker

[4] which, in turn was proved using non-constructive techniques similar to those used in Information Theory. In such arguments, the existence of an object which is, in some sense, good is proven by showing that, of all objects of size n, the fraction of objects which are not good is always less than 1. However, it can be extremely difficult to explicitly construct a family of objects that are good. This problem has been experienced in the construction of error-correcting codes as well.

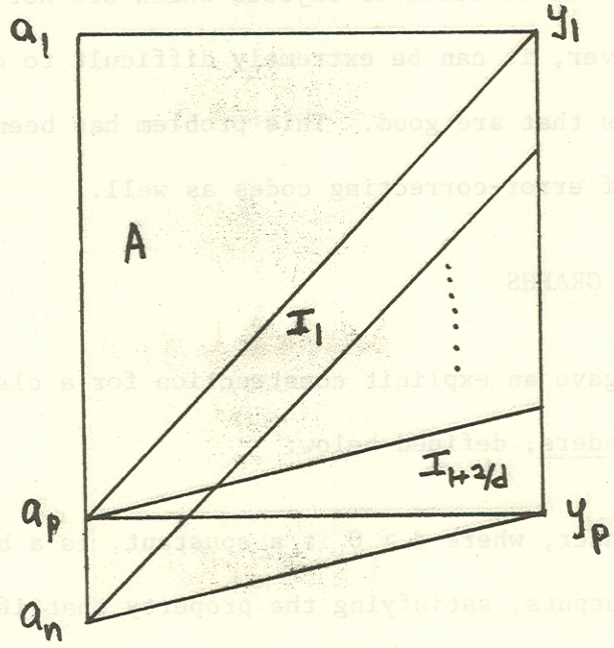
2.4.1 EXPANDER GRAPHS

Margulis [3] gave an explicit construction for a class of bipartite graphs called expanders, defined below:

An (n,d)-expander, where $d > 0$ is a constant, is a bipartite graph with n inputs, n outputs, satisfying the property that if T(X) is the set of outputs, each vertex in T(X) adjacent to at least one input in any set X, then

$$|T(X)| \geq |X|(1 + \alpha\ (1 - |X|/n)).$$

Given a family $\{G_n\}$ of (n,d)-expanders where, for each n, G_n has at most kn edges, we may construct a family $\{F_n\}$ of $\left(n, n.\left(\frac{2+d}{2+2d}\right), \frac{n}{2}\right)$-concentrators where each F_n has at most $n(k+1)\frac{(2+d)}{(2+2d)}$ edges. This is seen from the following construction, given in [2].



An (n,p,n/2)-concentrator C

In the figure above, $p = \left(\frac{2+d}{2+2d}\right)n$, and $\tau = n - p = \left(\frac{d}{2+2d}\right)n$. A is a (p,d)-expander, and each I_{j+1}, $0 \le j \le \frac{2}{d}$ is an identity map from $a_{p+1},\ldots,a_n$ to $y_{j+1},\ldots,y_{(j+1)\tau}$. We now show that C is an $(n, n\frac{(2+d)}{(2+2d)}, \frac{n}{2})$-concentrator.

Let X be any set of inputs such that $X = X_1 \quad X_2$, $X_i \subseteq \{i_1,\ldots,i_p\}$ $X_2 \subseteq \{i_{p+1},\ldots,i_n\}$ and $|X| \le \frac{n}{2}$. By Hall's marriage theorem, C is an $(n,p,\frac{n}{2})$-concentrator if and only if for each set of inputs X, such that $X \le \frac{n}{2}$, $|T(X)| \ge |X|$. There are two cases to consider.

Case (1): $|X_2| \ge |X|\tau/p$. Now each vertex in X_2 is connected to exactly (p/τ) output vertices, and each output vertex is connected to exactly one input vertex in X_2. Hence,

$$|T(X)| \ge |T(X_2)| \ge |X|(\tau/p)(p/\tau) = |X|.$$

Case (2): $|X_2| < |X|\,\tau/p$. Thus, $|X_1| > |X|\,(1-\tau/p)$

or, $|X_1| > |X|\,\left(1-\frac{d}{(2+d)}\right)$

or, $|X_1| > |X|\cdot\left(\frac{2}{(2+d)}\right)$

Let Y be any subset of X_1 such that

$$|Y| = \left\lceil 2|X|/(2+d)\right\rceil \le \left\lceil \frac{n}{(2+d)}\right\rceil = \left\lceil \frac{p(2+2d)}{(2+d)^2}\right\rceil$$

$$= \left\lceil \frac{p}{2}(1-(\tau/p)^2)\right\rceil$$

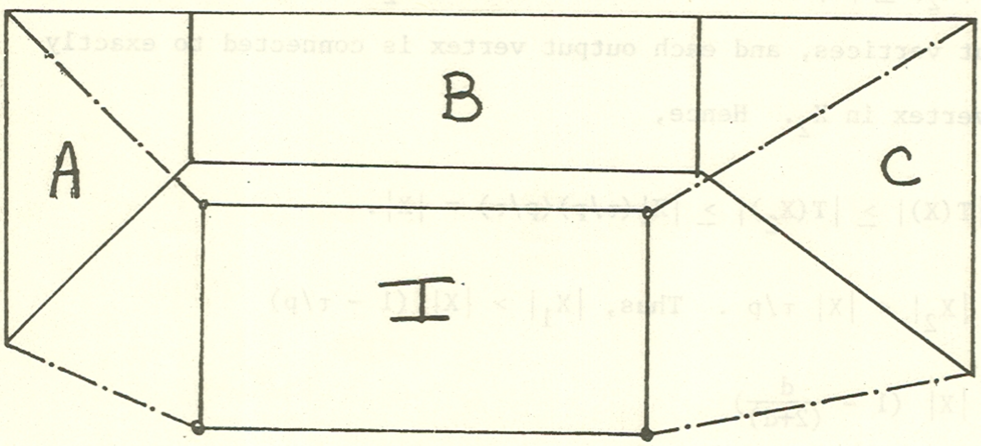
$$\le \frac{p}{2}.$$

Since A is an expander and $Y \subseteq X_1 \subseteq X$, we have:

$$|T(X)| \ge |T(Y)| \ge |Y|\;(1+d(1-|Y|/p))$$

$$\ge |Y|\;(1+\tfrac{d}{2}) = \left\lceil |X|\,\frac{2}{2+d}\right\rceil \frac{2+d}{2} \geq |X| \qquad \text{Q.E.D.}$$

Now, given an $(n, n\frac{2+d}{2+2d}, \frac{n}{2})$-concentrator with at most $\frac{n(k+1)(2+d)}{2+2d}$ edges, we can use the following construction to obtain linear size superconcentrators.



Here, A, C are the $(n, n\frac{2+d}{2+2d}, \frac{n}{2})$-concentrators, B is a recursively constructed $n\left(\frac{2+d}{2+2d}\right)$-superconcentrator and I is an identity map from the inputs to the outputs. The proof of construction is similar to that of Section 2.3. Furthermore, it is easily shown that the above n-superconcentrator has at most $\left(\frac{2(k+1)(2+d)+(2+2d)}{d}\right)n + o(n)$ edges.

2.4.2 EXPLICIT CONSTRUCTIONS OF EXPANDER GRAPHS

From the last section we see that the problem of explicitly constructing a family of linear superconcentrators is effectively reduced to the problem of explicitly constructing a family of expander graphs. In 1973, Margulis [3] gave a simple construction for a family of graphs which, for some constant d > 0 was shown to be a family of (n,d)-expanders. The proof is based on a series of reductions which are proven by applying several deep theorems from the theory of group representations. However, only the existence of a non-zero constant d was proven; neither the value nor any estimate (non-zero lower bound) of d was given. Consequently, the expanders constructed cannot be used to construct superconcentrators as in the preceeding section, since the construction is based on the value of d.

In 1979 Gabber and Galil [2] gave a construction for (n,d)-expanders with $d \geq \frac{2}{15}$, and at most 7n edges. Using these expanders the constructions of the preceeding section give $(n, n\cdot\frac{16}{17}, \frac{n}{2})$-concentrators with at most $(\frac{128}{7})n$ edges and n-superconcentrators with 273n + o(n) edges. In fact, it is easy to show that a family of superconcentrators with at most 274n edges can be constructed simply by constructing n-permutation networks instead of the above n-superconcentrators for n less than, say 2^{50}

The construction of a family G_n of bipartite expander graphs in [2], similar to that in Margulis [3], is described below for the case when n is the square of an integer m. Let Z_m denote the additive group of integers modulo m. Each input is associated with a distinct ordered pair $(x,y) \in Z_m \times Z_m$, as is each output. For each permutation $\sigma_i : Z_m^2 \to Z_m^2$ given below, construct an edge between input pair (x,y) and output pair $\sigma_i(x,y)$, for each pair (x,y). This defines the set of edges in the graph. The + below denotes addition modulo m.

$$\sigma_0(x,y) = (x,y)$$
$$\sigma_1(x,y) = (x,2x+y)$$
$$\sigma_2(x,y) = (x,2x+y+1)$$
$$\sigma_3(x,y) = (x,2x+y+2)$$
$$\sigma_4(x,y) = (x+2y,y)$$
$$\sigma_5(x,y) = (x+2y+1,y)$$
$$\sigma_6(x,y) = (x+2y+2,y)$$

Lemma [2]:

$$(\forall X \subseteq Z_m^2) \sum_{\substack{i=0 \\ i \neq 2,5}}^{6} |\sigma_i(X) - X| + 2 \sum_{i=2,5} |\sigma_i(X) - X| \geq \frac{16}{15} |X| \left(1 - \frac{|X|}{n}\right)$$

where $\sigma_i(X) = \bigcup_{(x,y) \varepsilon X} \sigma_i(x,y)$

The above lemma is proved in [2] by transforming it into its continuous version, i.e., mapping the discrete space Z_m^2 into the continuous $[0,1) \times [0,1)$ torus and proving the corresponding lemma via a series of reductions involving basic theorems in Fourier and Complex analyses.

From the lemma, it follows that

$$(\forall X \subseteq Z_m^2)\ (\exists i)\quad |\sigma_i(X) - X| \geq \left(\frac{2}{15}\right) |X| \,(1 - |X|/n) \qquad (1)$$

which in turn implies

$$(\forall X \subseteq Z_m^2)\quad \left|\bigcup_i \sigma_i(X) - X\right| \geq \left(\frac{2}{15}\right) |X| \,(1 - |X|/n) \qquad (2)$$

Since $\bigcup_i \sigma_i(X) = T(X)$, we have :

$$(\forall X \subseteq Z_M^2) \quad |T(X)| \geq |X|(1+(\frac{2}{15})(1-|X|/n))$$

or, equivalently, G_n is an $(n,\frac{2}{15})$-expander.

It should be noted that (1) is a much stronger condition than necessary to prove (2). It follows that for the construction, G_n is an (n,d)-expander with $d \geq \frac{2}{15}$. From the construction of the previous section we note that the higher the value of d, the fewer the number of edges required to construct an n-superconcentrator. Unfortunately, since the proof in [2] of construction of the expander described above is rather indirect, we do not get much insight into how a better lower bound for d might be obtained, if, in fact, there is one. Also, it is not clear how expanders with higher d, but without a drastic increase in the number of edges, may be constructed. Finally, it would be somewhat more satsifying to have a combinatorial proof for an essentially combinatorial problem.

2.5 ATTEMPTS AT CONSTRUCTING LINEAR EXPANDERS COMBINATORIALLY

In the previous section we saw how n-superconcentrators with no more than 274n edges could be constructed using the construction of expanders. Also, in Section 2.1 we observed that permutation networks with no more than 3.79 nlogn edges could be used as n-superconcentrators. Now, the explicit construction is more efficient than the permutation network when $274n < 3.79\ n\log n$, or, roughly, only when $n > 10^{21}$. This means that for all practical purposes the O(nlogn) construction is more efficient than the O(n) construction. Furthermore, the operational complexity of the permutation network is O(nlogn) which is lower than that of the linear superconcentrator, which, as will be seen in the next chapter is $O(n\sqrt{n})$.

In what follows, we describe some attempts made at constructing expanders with the aim of providing purely combinatorial proofs for the constructions. Although the attempts have been unsuccessful, the techniques used to prove why the constructions do not yield expanders seem effective in ruling out various interesting possibilities and could be extended considerably to prove that larger classes of constructions do not yield expanders, thereby, hopefully, providing some insight into the nature of combinatorial structures necessary to construct expanders.

Given an undirected graph $G_n = (E,V)$, $V = \{\nu_1, \nu_2, \ldots, \nu_n\}$, we define $B(G_n)$ to be a bipartite graph with n inputs $\{x_1, \ldots, x_n\}$, n outputs $\{y_1, \ldots, y_n\}$, and having edges as follows:

(1) for each j, $1 \leq j \leq n$, an edge (x_j, y_j)

(2) for each edge (ν_k, ν_j) in E, an edge (x_k, y_j).

Furthermore, if B(G) is an (n,d)-expander, then G is said to be an (n,d)-expander graph.

Definition: If G = (E,V) is any graph and $X \subseteq V$ then the set $C(X) = \{y \in V-X \mid (\exists x \in X)\ (x,y) \in E\}$ is called the <u>co-boundary</u> of X.

From the definition of expanders it is clear that if G = (E,V) is an (n,d)-expander graph, then the co-boundary C(X) of any set $X \subseteq V$ must satisfy the following condition:

$$|C(X)| \geq d\ |X|\ (1 - \frac{|X|}{|V|}) .$$

Thus, the problem of constructing a family of linear d-expanders is equivalent to the problem of constructing a family of d-expander graphs G_n for which the number of edges grows linearly with the number of vertices.

As a simple example, consider the square grid Z_m^2, $m^2 = n$, in which each vertex is connected to its nearest four neighbors. If X is the set of vertices consisting of the top t rows of the grid, $|X| = tm$, then $|C(X)| = m = |X|/t$. Thus if $t = \alpha m$ for some constant α, $0 < \alpha < 1$, we have $|X| = \alpha m^2$, and $|C(X)| = m$. Since for all constants d, $m < d\, m^2 (1-\alpha)$ when $m > \frac{1}{d\alpha(1-\alpha)}$, it follows that the grid is not an expander graph.

Similarly, it has been shown that if $G = (E,V)$ is a planar graph, then there is a subset X of vertices, $|X| = c|V|$ for some $c < 1$, such that $|C(X)| = O(\sqrt{|X|})$. This implies that a family of planar graphs cannot be a family of d-expander graphs for any constant d.

For any subset X of vertices in a graph (E,V), we define $S(X) = \{(v_1,v_2) \in E \mid v_1 \in X \text{ and } v_2 \in V-X\}$, i.e., S(X) is the set of edges with exactly one end point in X. It is clear that for all graphs and subset X of vertices $|S(X)| \geq |C(X)|$. Thus for an (n,d)-expander graph (E,V) it follows that for every $X \subseteq V$, $|S(X)| \geq d\, |X| (1-|X|/|V|)$. The number $\omega = \min \{|S(X)| \mid |X| = |V|/2$ is called the minimum bisection width of a graph. Informally, it is the smallest number of edges whose removal disconnects one half of the vertices from the other. From the above discussion it follows that for a family of d-expander graphs G_n, if ω_n denotes the minimum bisection width of G_n, $\omega_n \geq \frac{dn}{4}$. In what follows we consider two families of graphs, and show that, for each, $\omega_n = o(n)$, thus proving that neither is a family of d-expanders, for any constant $d > 0$.

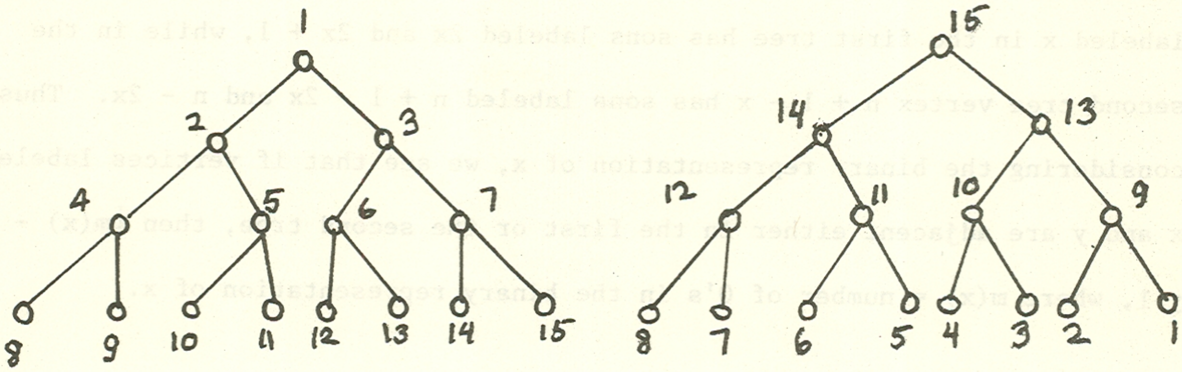
Construction 1: Consider a full binary tree on $n = 2^k-1$ vertices. For convenience, we shall restrict ourselves to the case where k is even. In the tree, for any set X of non-leaf vertices, $|C(X)| \geq X$. Thus, as long as we do not choose leaves, the tree has good expander properties. This suggests that it might be possible to construct expanders by adding extra edges to each leaf in the tree. One way to do this is as follows:

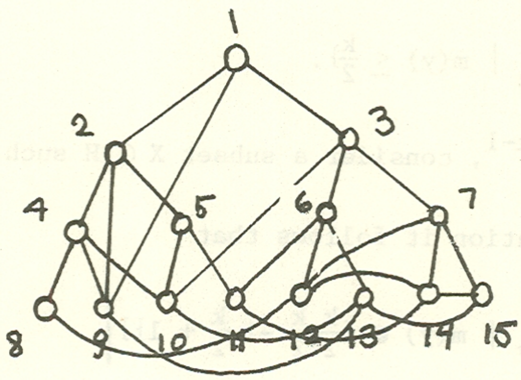
Create two copies of the tree. Traverse both trees left-to-right, breadth-first, visiting the root first. In the first tree label the i-th vertex visited i, while in the second tree label the i-th vertex visited (n-i). Now, $G_n = (E,V)$ is defined as follows :

$v = v_1,\ldots,v_n$

$E = \{(v_i,v_j) |$ vertices labeled i and j are adjacent in at least one of the trees$\}$

The case n = 15 is shown next





Note that in the two trees the labels of internal vertices of one correspond to labels of the leaves of the other, and vice-versa, with the exception of the left-most leaf which is labeled $\frac{(n+1)}{2}$ in both trees. The intuitive justification for this construction is that for any set X of vertices, the co-boundary will be "large enough" in at least one of the two labeled trees and hence also in G_n.

It is straightforward to see that for x between 1 and $\frac{(n-1)}{2}$, the vertex labeled x in the first tree has sons labeled 2x and 2x + 1, while in the second tree vertex n + 1 - x has sons labeled n + 1 - 2x and n - 2x. Thus, considering the binary representation of x, we see that if vertices labeled x and y are adjacent either in the first or the second tree, then $|m(x) - m(y)| \leq 1$, where m(x) = number of 0's in the binary representation of x.

Now, consider the following subset of vertices of G_n :

$$H = \{v_y \in G_n \mid m(y) \leq \frac{k}{2}\}.$$

Since $H = \sum_{i=0}^{\frac{k}{2}} \binom{k}{i} > 2^{k-1}$, consider a subset $X \subset H$ such that $|X| = 2^{k-1}$.

From the above observation it follows that

$$|S(X)| \leq \left|\{v_y \mid m(y) \in \{\frac{k}{2}, \frac{k}{2} -1, \frac{k}{2} + 1\}\}\right|$$

or, $$|S(X)| \leq 3 \binom{k}{\frac{k}{2}} = c2^k/\sqrt{k}, \text{ for some constant } c.$$

Hence, $\omega_n \leq \frac{c'n}{\sqrt{\log n}}$ for some constant c´. Thus, $\omega_n = o(n)$ and the proposed construction does not yield expander graphs.

We have considered only one scheme by which to connect leaves in a tree to internal vertices. Obviously, other schemes are possible, many of which may be ruled out on the basis of arguments similar to the one above, or by simple extensions of it.

Construction 2: We next consider a family of graphs G_n where the n-th graph has n! vertices. Again, for convenience, we assume that n is even. The construction of G_n is described below:

Let P be the set of all permutations over Z_n. Each permutation π is represented as a sequence $\pi(1)\ \pi(2),\ldots,\pi(n)$. G_n has n! vertices ν_i, $1 \leq i \leq n!$, each vertex ν_x is labeled with a distinct element $\hat{x}$ of P. The set of edges is

$$E = \left\{ (\nu_x, \nu_y) \mid \sigma_i(\hat{x}) = \hat{y} \text{ for } i \in \{1,2,3\} \right\}$$

where each σ_i is a bijective function over P, and defined as follows (let $a_1 \ldots a_n \in P$):

$$\sigma_1(a_1 a_2 \ldots a_n) = (a_2 a_1 \ldots a_n)$$

$$\sigma_2(a_1 a_2 \ldots a_n) = (a_2 \ldots a_n a_1)$$

$$\sigma_3(a_1 \ldots a_{n-1} a_n) = (a_n a_1 \ldots a_{n-1})$$

For this construction define a measure m on P as:

$$m\ (a_1 \ldots a_n) = \sum_{i=1}^{n/2} X\ (a_i)$$

where $$X(a_i) = \begin{cases} 1\ , & a_i > n/2 \\ 0\ , & a_i \leq n/2 \end{cases}$$

Thus, $m(a_1 \ldots a_n)$ = number of integers in $a_1 \ldots a_{n/2}$ that are greater than n/2. By construction, if $(\nu_x, \nu_y) \in E$, then $\left| m\ (\hat{x}) - m(\hat{y}) \right| \leq 1$.

Let $X \subseteq \left\{ \nu_x \;\middle|\; m(\hat{x}) \leq n/4 \right\}$ such that $|X| = n!/2$.

Then, as in the first construction, for any choice of X,

$$|S(X)| \leq 2 \left| \{ v_x \mid m(\hat{x}) = \tfrac{n}{4} \} \right|$$

$$= 2\binom{\frac{n}{2}}{\frac{n}{4}}^2 \left(\left(\tfrac{n}{2}\right)!\right)^2$$

$$= 2\left(\frac{\left(\frac{n}{2}\right)!}{\left(\frac{n}{4}\right)!}\right)^4 \quad = \quad 0\left(\frac{n!}{\sqrt{n}}\right)$$

$$= \quad o\,(|V|) \text{ since } |V| = n!$$

Thus we see that this proposed construction also fails since the minimum bisection width is a sub-linear function of the number of vertices in the graph. Informally, we may state that if for a graph G there is a "reasonable" measure on the vertices of G such that each edge connects vertices whose measures differ only very slightly, then it is likely that there are subsets of vertices with small co-boundary. The notion of what constitutes a "reasonable" measure seems difficult to formalize. As an extreme example a measure in which all vertices have identical value, is clearly "unreasonable".

Arguments such as the ones discussed here obviously do not directly help us in developing combinatorial techniques to prove the correctness of constructions that are known to be correct. However, developing combinatorial techniques to disprove interesting constructions as well as enlarging the class of promising constructions which fail might be of independent interest, and might even lead, indirectly to insights into the nature of combinatorial structures required to construct expanders.

Finally, the difficulty with developing combinatorial techniques to prove the correctness of known constructions seems to lie in the observation that, while it is comparatively easier to develop combinatorial techniques to handle well structured or modular constructions, expander graphs seem to require some, so far combinatorially intangible, notion of randomness. For example, if the intuitive notion of a "reasonable" measure on vertices were to be formalized, we would be required to find a construction for which, at the very least, there is no reasonable measure such that all edges lie between vertices whose measures differ only very slightly.

CHAPTER 3: MAIN RESULTS

In this chapter a probabilistic approach to constructing superconcentrators based on a proof technique of Pippenger [5], is considered in which randomly constructed bipartite praphs are used as components in constructing larger networks. The networks constructed are superconcentrators with probability greater than any fixed $c < 1$, and are efficient for reasonably large choices of c and large n as might be required in practice. Furthermore, if the networks constructed are not superconcentrators then, on the average, few edges need be added to make them superconcentrators. The problem of determining which edges to add may be solved efficiently during operation of the network.

In Section 3.2 we obtain constructive and non-constructive upper bounds on the structural complexity of incrementally non-blocking connection networks. Both are based essentially on the approach in [1]; the non-constructive upper bound presented is an improvement over that in [1] while the constructive upper bound is achieved employing the expanders constructed in [2]. Finally, some open problems are outlined.

3.1 PROBABILISTIC CONSTRUCTIONS

In Chapter 2 we noted that although the recent explicit construction in [2] of a family of expanders and superconcentrators solves a long outstanding open question, the construction of superconcentrators is not of practical value as it uses fewer edges than a permutation network only when $n > 10^{21}$. In this section a different approach is considered, in which randomly constructed graphs are used as components in constructing superconcentrators. We term such constructions "probabilistic constructions". This approach is suggested by the nature of non-constructive proofs of linear upper bounds on rearrangeable concentrators, wherein it is argued that the fraction of graphs that are, in some sense, good tends to unity as the number of inputs tends to infinity. In the context of switching networks, a network is good if and only if it is non-blocking, i.e., each request can be realized as long as the total number of requests does not exceed the capacity of the network. Now, in practice, instead of requiring that a constructed network be guaranteed to be non-blocking, the construction might still be useful if we can show that the probability that it is blocking is less than ε, where ε is some presepcified quantity, which might either be a constant or a function of the number of inputs of the network.

Networks of the latter kind will be referred to as ε-blocking networks. Note that, as in [6], for an ε-blocking network the probability that a particular set of requests is unrealizable is less than ε, the probability that at least one set of requests is unrealizable.

In the following sections we discuss ways to construct non-blocking, as well as ε-blocking linear superconcentrators probabilistically.

For the ε-blocking superconcentrator construction it is further shown that if the construction is not non-blocking, then, on the average, few edges need be added to the network to make it non-blocking.

3.1.1 PROBABILISTIC CONSTRUCTION OF LINEAR SUPERCONCENTRATORS

The construction discussed here corresponds to Pippenger's proof [5] of the existence of linear superconcentrators.

Let π be an arbitrary permutation over Z_{36m}, where m is a positive integer. Construct the graph G_π which has inputs $i_1,\ldots,i_{6m}$ and outputs $y_1,\ldots,y_{4m}$. The set of edges G_π is $\left\{(i_{x \bmod 6m}, y_{\pi(x) \bmod 4m}) \mid x \in Z_{36m}\right\}$. Note that the outdegree of each input is at most 6 and the indegree of each output is at most 9. Also, the minimum outdegree of any input i_x is 1, which occurs when $\{(x + i6m) \bmod 4m \mid 0 \leq i \leq 5\}$ is a singleton set. Similarly, the minimum possible indegree of any output is 2. The following lemma is a slightly modified version of that in [5].

Lemma 1: The fraction of all permutations π over Z_{36m} such that G_π is not a rearrangeably non-blocking (6m, 4m, 3m)-concentrator is $O(m^{-13})$.

Proof: We first observe that G_π is a rearrangeably non-blocking (6m, 4m, 3m)-concentrator if and only if for every subset X of inputs such that $|X| \leq 3m$, $|T(X)| \geq |X|$. Next, we obtain an upper bound for the fraction of permutations π for which at least one set of k requests s, $k \leq 3m$, is blocked.

Choose any set X of k inputs, $k \leq 3m$, and a set Y of (k-1) outputs. We count the number of permutations for which $T(X) \subseteq Y$, i.e., for which the request X is blocked. The set X corresponds to a set $X' \subseteq Z_{36m}$ such that $|X'| = 6k$ and, similarly Y corresponds to a set $Y' \subseteq Z_{36m}$ such that $|Y'| = 9k-9$. The number of permutations for which $T(X) \subseteq Y$ is then equal to the number of permutations over Z_{36m} for which $\{\pi(x) \mid x \in X'\} \subseteq Y'$, which is

$\left[{9k-9}\right]_{6k}$ (36m-6k)! Now, since X and Y can be chosen in $\binom{6m}{k}$ and $\binom{4m}{k-1}$ ways respectively, an upper bound on the fraction of permutations for which G_π is not a rearrangeable (6m, 4m, 3m)-concentrator is

$$I_m = \sum_{k=3}^{3m} \binom{6m}{k}\binom{4m}{k-1} \frac{\left[9k-9\right]_{6k} (36m-6k)!}{(36m)!}.$$

Observe that since each input has outdegree at least 1 and each output has indegree at least 2, $|T(X)| \geq |X|$ for all subsets of size 1 and 2.

From the above,

$$I_m = \sum_{k=3}^{3m} \frac{\binom{6k}{k}\binom{4m}{k-1}\binom{9k-9}{6k}}{\binom{36m}{6k}}$$

Since $\binom{36m}{6k} \geq \binom{6m}{k}\binom{4m}{k-1}\binom{26m}{4k+1}$,

$$I_m \leq J_m = \sum_{k=3}^{3m} \frac{\binom{9k-9}{6k}}{\binom{26m}{4k+1}}$$

Now, it is straightforward to show that the largest term in J_m is either the first or last. Thus, there are two cases to consider:

Case (1): The largest term in J_m is the last.

$$\text{Thus } I_m \leq J_m \leq 3m \cdot \frac{\binom{27m-9}{18m}}{\binom{26m}{12m+1}}$$

$$< 3m \cdot \frac{\binom{27m}{18m}}{\binom{26m}{12m}} = O(2^{-m}) .$$

Case (2): The largest term in J_m is the first.

$$\text{Thus } I_m \leq J_m \leq \frac{\binom{18}{18}}{\binom{26m}{13}} + \sum_{k=4}^{3m} \frac{\binom{9k-9}{6k}}{\binom{26m}{4k+1}}$$

$$\leq \frac{1}{\binom{26m}{13}} + \frac{3m \cdot \binom{27}{24}}{\binom{26m}{17}}$$

$$= O(m^{-13})$$

Combining Cases 1 and 2, the lemma follows. Furthermore, it is easy to show that for $m > 100$, $I_m < 10^{-7} m^{-13}$. Q.E.D.

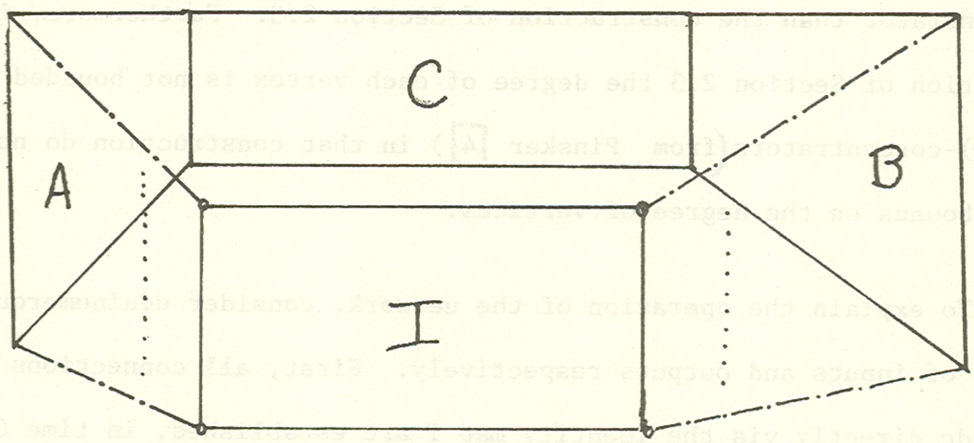
As shown in [5], a slight refinement of the calculation above shows that I_m is strictly less than 1 for all m.

A generalized version of the lemma is now stated:

Lemma 2: The function of permutations over Z_{6im} for which G is not a rearrangeable (6m, 4m, 3m)-concentrator is $O(m^{-3i+5})$, where $i \geq 6$ is a constant divisible by 2.

Proof: Similar to that of Lemma 1.

Now using rearrangeable $(n, \frac{2}{3}n, \frac{n}{2})$-concentrators with at most 6n edges, n-superconcentrators are constructed as shown below.



In the figure, A and B are copies of a rearrangeable $(n, \frac{2}{3}n, \frac{n}{2})$-concentrator, C is a recursively constructed $\frac{2}{3}$n-superconcentrator, and I is an identity map connecting the n-inputs to the n-outputs. The proof that the network thus constructed is an n-superconcentrator is identical to that in Section 2.3. From the figure we see that if S(n) denotes the number of edges used in the construction, then

$$S(n) = S(\lceil \tfrac{2}{3} n \rceil) + 13\, n$$

Also, $S(1) = 1$, which yields $S(n) \leq 39n + O(\log n)$. In fact, if for $n \leq 2187$, we construct permutation networks in the recursive construction, then Pippenger [5] shows that $S(n) \leq 40n$ for all n.

It should also be noted that the above construction establishes a tighter upper bound on the minimum number of edges required to construct a superconcentrator than the construction of Section 2.3. Furthermore, in the construction of Section 2.3 the degree of each vertex is not bounded since the $(n, \frac{n}{2})$-concentrators (from Pinsker [4]) in that construction do not have $O(1)$ bounds on the degree of vertices.

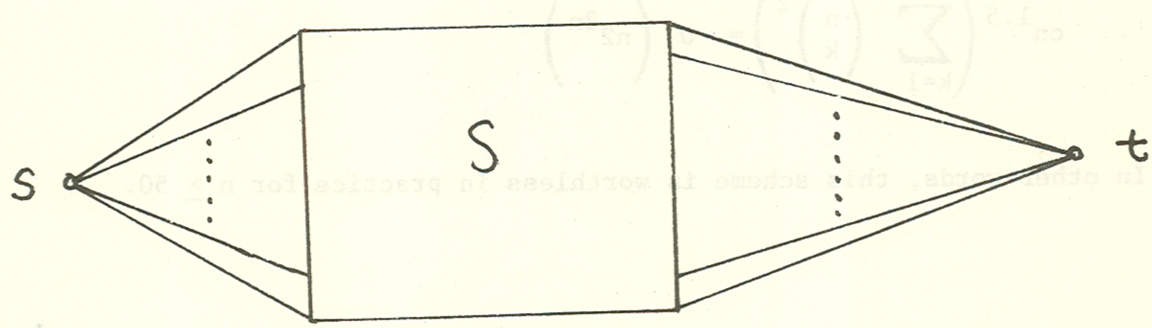
To explain the operation of the network, consider equinumerous sets X and Y of inputs and outputs respectively. First, all connections that can be made directly via the identity map I are established, in time $O(n)$. Then the remaining inputs X', $|X'| \leq \frac{n}{2}$, are connected 1-1 to a set X'' of inputs of the $\frac{2}{3}n$-superconcentrator C by finding an $|X'|$ $(=|X''|)$ flow through A. Similarly, the outputs $Y' \subseteq Y$ not connected over I, are mapped onto outputs of C. Finding an $|X'|$-flow through A takes $O(n^{1.5})$ time. Finally, the above procedure is repeated for each of the inner superconcentrators until all connections are established.

From the above, we see that if $T(n)$ is the time required to extablish connections in an n-superconcentrator, then $T(n) \leq cn^{1.5} + T(\frac{2}{3}n)$, and $T(1) = d$, for some constants c and d. From this we have $T(n) = O(n^{1.5})$. This contrasts with the $O(n\log n)$ operational complexity of the permutation network of Section 2.1. The above discussion also accounts for the $O(n^{1.5})$ operational complexity of the explicit construction of Section 2.4. Note, however, that in the case of an ε-blocking superconcentrator, it may happen that for some set X of inputs of one of the probabilistically constructed ε-blocking rearrangeable concentrators there is no $|X|$-flow to the corresponding set of outputs. We discuss a possible solution for this problem in Section 3.1.3.

3.1.2 GUARANTEED NON-BLOCKING SUPERCONCENTRATORS

In this Section we examine the possibility of probabilistically constructing superconcentrators that are guaranteed to be non-blocking, i.e. non-blocking with probability exactly 1. In general, the only possible approach known is to exhaustively test the networks constructed to verify that they are non-blocking. Unfortunately, the best known algorithm takes exponential time as we shall see here. The complexity of the problem of recognizing a graph that is not non-blocking is in NP, the class of problems solvable non-deterministically in polynomial time, but its relationship with respect to the NP-complete problems is not known.*

A rather naive way to test a probabilistically constructed superconcentrator would be to verify that for every choice of k inputs and k outputs, $1 \leq k \leq n$, there is a k-flow from the k inputs to the k outputs, using a maximum network flow algorithm for the following network:



* C.H.Papadimitriou (personal communication) has recently shown that the problem of recognizing a superconcentrator is co-NP complete.

s is the source vertex, $i_1, \ldots, i_n$ and $y_1, \ldots, y_n$ are respectively sets of inputs and outputs of the n-superconcentrator S, and t is the terminal vertex. There are edges from s to each input and from each output to t. Capacities of all vertices within S are 1. Given a set X of k inputs and a set Y of k outputs, the following edge capacities are defined:

$$C(x, i_x) = \begin{cases} 1, & \text{if } i_x \in X \\ 0, & \text{else} \end{cases}$$

$$C(y_x, t) = \begin{cases} 1, & \text{if } y_x \in Y \\ 0, & \text{else.} \end{cases}$$

With the edge capacities as defined, there is a k-flow from X to Y if and only if the maximum flow through the network from s to t is k. Thus, to test a superconcentrator we could run a maximum network flow algorithm once for each choice of input and output set pairs. However, using an $0(n^{1.5})$ maximum network flow algorithm, the complexity of the above scheme is

$$cn^{1.5}\left(\sum_{k=1}^{m} \binom{n}{k}^2\right) = 0\left(n2^{2n}\right)$$

In other words, this scheme is worthless in practice for $n \geq 50$.

This naive approach may be improved by testing separately for each probabilistically constructed bipartite graph G_m with m inputs and $\frac{2}{3}m$ outputs to verify that it is a rearrangeable $(m, \frac{2}{3}m, \frac{m}{2})$-concentrator. If for any m, G_m is not an $(m, \frac{2}{3}m, \frac{m}{2})$-concentrator, it is replaced by another randomly constructed graph G_m, which is then checked and replaced if necessary. This process is iterated until an $(m, \frac{2}{3}m, \frac{m}{2})$-concentrator is obtained. Since the probability that a randomly constructed graph is not a concentrator is extremely low, not more than a few iterations will be required in practice. Finally, when each bipartite graph G_m is known to be an $(m, \frac{2}{3}m, \frac{m}{2})$-concentrator, the overall construction is guaranteed to be a superconcentrator. Now, G_m is an $(m, \frac{2}{3}m, \frac{m}{2})$-concentrator, if and only if, for each input set X, $|X| \leq \frac{m}{2}$, $|T(X)| \geq |X|$. So, we could check these

$$\mu = \sum_{i=1}^{\frac{m}{2}} \binom{m}{i} = 0(2^m)$$

conditions instead of checking that each subset of the inputs of size k, $k \leq \frac{m}{2}$, a k-flow to the set of outputs. This is done as follows:

Generate the subsets of size at most $\frac{m}{2}$ sequentially such that for each i, S_i and S_{i+1}, the i-th and (i+1)-st subsets generated satisfy:

(a) $$|S_i \cap S_{i+1}| = \max\{|S_i|, |S_{i+1}|\} - 1$$

Next consider the graph G_m. Associated with each output vertex x is a counter C(x) which gives the number of input vertices in S_i that x is adjacent to. The quantities $|T(S_i)|$ and $|S_i|$ are stored in variables COUNT and SIZE respectively. the following program then verifies the required condition..

```
1.  Generate S_1,

2.  SIZE ← |S_1|

3.  For each input x∈S_1 do
        For each output y adjacent to x do
            C(y) ← C(y) + 1

4.  For each output y do
        If C(y) ≠ 0 then COUNT ← COUNT + 1

5.  For i = 2 until μ do
    Begin
        a. Generate S_i
        b. SIZE ← |S_i|
        c. For each output y adjacent to the single input in S_{i-1}-S_i do
            1) C(y) ← C(y) - 1
            2) If C(y) = 0 then COUNT ← COUNT - 1
        d. For each output y adjacent to the single input in S_i-S_{i-1} do
            1) If C(y) = 0 then COUNT ← COUNT + 1
            2) C(y) ← C(y) + 1
        e. If SIZE > COUNT then output "Not a Concentrator" and halt
                else if i = μ then output "Is a Concentrator"
                and halt
                    else i ← i + 1.
```

That the program is correct is straightforward. Furthermore, the initialization steps 1 through 4 take O(m) time, while for each i, the loop of step 5 takes O(1) time; consequently, step 5 takes $O(\mu) = O(2^m)$ time. Hence, the complexity of testing an $(m, \frac{2}{3}m, \frac{m}{2})$-concentrator is $O(2^m)$.

Finally, the complexity of testing an n-superconcentrator is

$$\leq \; C \cdot \sum_{i=0}^{\log n} 2^{\left(\frac{2}{3}\right)^i n} = O(2^n) .$$

Thus, this algorithm cannot be used for n > 100, since the time required to verify an n-superconcentrator is impossibly high. Besides, for $n < 3^7$ an n-permutation network uses fewer edges than the proposed construction. This virtually rules out any possibility of constructing networks guaranteed to be non-blocking by exhaustively testing probabilistic constructions. However, the complexity of recognizing graphs that are not non-blocking is of theoretical interest. It is clearly in the class NP; given a graph that is not non-blocking, "guess" a subset of inputs of size k for which there is no k-flow to the set of outputs, and verify this fact in polynomial time using a maximum network flow algorithm. The relationship of this problem with the class of NP-complete problems is not known.* Furthermore, it would be of interest if exponential algorithms whose complexity is $o(2^n)$ are found, since these would probably involve combinatorial techniques more powerful than Hall's marriage theorem.

*see footnote on page 47

3.1.3 ε-BLOCKING SUPERCONCENTRATORS

To construct an n-superconcentrator in general, if $n \leq N$, where N is some constant we construct an n-permutation network with at most $\frac{6}{\log 3}$ nlogn - 3n edges, and for n>N we obtain two copies A, B, of a probabilistically constructed $(n, \frac{2}{3}n, \frac{n}{2})$-concentrator and a $\frac{2}{3}$n-superconcentrator constructed recursively according to the above rules, and combine these as in Section 3.1.1. The reason for constructing a permutation network whenever a superconcentrator with fewer than N inputs is required is that for small graphs a probabilistic construction is more likely to fail than for bigger graphs. In any case, whenever $(\frac{6}{\log 3})\, n\log n - 3n \leq 39n$, i.e., $n \leq 3^7$, a permutation network is at least as efficient as the proposed construction. Thus $N \geq 3^7$. For general N, the probability that the probabilistic construction above fails to be a superconcentrator is calculated below:

Let ε = probability that the construction fails

= probability that at least one probabilistically constructed bipartite graph G is not an $(n, \frac{2}{3}n, \frac{n}{2})$-concentrator.

(Note: here π is a permutation over Z_{6n})

= 1 - probability that each G above is a $(n, \frac{2}{3}n, \frac{n}{2})$-concentrator.

By the observation made in the proof of lemma 2, since m > 100, we have that

$$\varepsilon \leq 1 - \prod_{i=0}^{\log_{\left(\frac{3}{2}\right)}\left(\frac{n}{N}\right) - 1} \left(1 - \left(\left(\frac{2}{3}\right)^{i} \frac{n}{6}\right)^{-13} 10^{-7}\right)$$

Now,

$$\prod_{i=0}^{M} (1 - \alpha^{i} x) = 1 - C_1 x + C_2 x^2 + \ldots + (-1)^{M+1} C_{M+1} x^{M+1} ,$$

where

$$C_1 = \sum_{i=0}^{M} \alpha^{i} = \frac{\alpha^{M+1} - 1}{\alpha - 1}$$

If $x \ll 1$, neglecting higher order terms, we have:

$$\prod_{i=0}^{M} (1 - \alpha^{i} x) \leq 1 - C_1 x$$

In the case above, $\alpha = \left(\frac{3}{2}\right)^{13}$, $x = \left(\frac{10^{-7}}{6^{-13}} \cdot n^{-13}\right)$, giving

$$\varepsilon \leq (N^{-13} - n^{-13}) \, 10^{-7} . 2^{26}$$

For, $N = 3^{7}$, $\varepsilon \leq 2^{-150}$

which should be low enough for practical purposes.

In general, if we want the probability that the construction fails to be less than ε_o then it is sufficient to ensure:

$$2^{26} . (N^{-13} - n^{-13}) . 10^{-7} \leq \varepsilon_o$$

or

$$N \geqslant \left(10^{7} \cdot 2^{-26} \cdot \varepsilon_0 + n^{-13}\right)^{-\frac{1}{13}}$$

So N need at most equal the limit of the right side of the inequality as $n \to \infty$, giving $N \simeq 1.2 \, \varepsilon_0^{-1/13}$

Suppose we construct an n-superconcentrator probabilistically using a permutation network only for n = 100. To ensure that the probability the graph fails is less than ε, we replace the subgraph which is an N-superconcentrator by an N-permutation network. Letting $S_{\varepsilon(n)}$ denote the minimum possible number of edges required to construct a network with n inputs and outputs which is a superconcentrator with probability greater than 1 − ε, it is clear that

$$S_{\varepsilon(n)} \leq S(n) + 4\, N \log N - 39N$$

or

$$S_{\varepsilon(n)} \leq 40n + \left(\frac{5}{13}\right)\left(\varepsilon^{-\frac{1}{13}} \log \varepsilon^{-1}\right) - 46\, \varepsilon^{-\frac{1}{13}}$$

Given arbitrarily small ε, we could make use of Lemma 2 to obtain ε-blocking superconcentrators, by simply increasing the number of edges in each bipartite graph constructed probabilistically. However, to keep the number of edges low, we should increase the number of edges in the smaller bipartite graphs more than in the bigger graphs. In the expression above this would correspond to reducing the second term on the right side asymptotically, while increasing the coefficient of n only slightly.

Another way to increase the probability that the construction yields a superconcentrator is to partially check each probabilistically constructed G_π. Given G_π, check whether or not for each subset X of inputs of size at most c, where c is a constant, $|T(X)| \geq |X|$. If not, either replace G_π by another probabilistically constructed graph, or add extra edges to G so that the above condition is satisfied. This partial verification can be done efficiently, i.e. in time polynomial in m. From the argument in Lemma 1 we observe that if G_π is partially correct in the sense above, then it is extremely likely to be an $(m, \frac{2}{3}m, \frac{m}{2})$-concentrator. Furthermore, as we shall see further on, on the average, only 0(1) edges need be added to G_π, given that it is not an $(m, \frac{2}{3}m, \frac{m}{2})$-concentrator.

Suppose that one of the probabilistically constructed bipartite graphs G_π with m inputs and $\frac{2}{3}m$ outputs with at most 6m edges, is not an $(m, \frac{2}{3}m, \frac{m}{2})$-concentrator. This means that during the operation of the network, for some set k of inputs, $k \leq \frac{m}{2}$, there is no k-flow to the outputs of G_π. At this stage one possible approach would be to replace G by another randomly constructed graph G_π. Another alternative is to add extra edges to G_π so that for the k inputs a k-flow is now obtained. Determining which edges to add may be performed efficiently; simply obtain a maximal flow for the k inputs in G and add edges from the remaining unconnected inputs to an equal number of unconnected outputs.

Since the second alternative involves increasing the number of edges in the network, we must be careful that not too many edges are added. Otherwise, the structural complexity of the network could become non-linear in the number of inputs. We now show that the expected number of edges added to the network is 0(1).

First we note that each time edges are added to a set of inputs X of G_π to obtain an $|X|$-flow, there is at least one subset $Y \subseteq X$ such that, before edges were added, $|T(Y)| < |Y|$, but after the addition of edges to G_π $|T(Y)| \geq |Y|$. Also, of the extra edges added, at most $|Y|$ would have an endpoint in Y. Letting E(A) denote the expected value of A, and $E(A|B)$ the conditional expected value of A given event B, we have:

E(number of edges added $|$ G_π is not a (6m,4m,3m)-concentrator)

(Note: As in Lemma 1, π is now a permutation over Z_{36m})

$$< \sum_{i=0}^{3m} i.E \left(\begin{array}{l} \text{Number of subsets X of} \\ \text{inputs such that } |X| = i \\ \text{and } |T(X)| < i \end{array} \middle| \begin{array}{l} G_\pi \text{ is not a} \\ (6m,4m,3m)\text{-concentrator} \end{array} \right).$$

For convenience, we term a subset X of inputs bad if $|T(X)| < |X|$. Also term G_π bad if it is not a (6m,4m,3m)-concentrator.

$$\text{Let } X_i = E\left(\begin{array}{l} \text{number of bad subsets} \\ \text{of size i} \end{array} \middle| G_\pi \text{ is bad} \right)$$

$$= (\text{number of subsets of size i}) \cdot \left(\begin{array}{c} \text{prob a given subset} \\ \text{of size i is bad} \end{array} \middle| G_\pi \text{ is bad} \right)$$

Now, in general if $A \subseteq B$ then $P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A)}{P(B)}$.

Since a bad subset implies G_π is not a concentrator, we have:

$$X_i = \binom{6m}{i} \frac{\text{prob(subset X of size i is bad)}}{\text{prob } (G_\pi \text{ is bad})}$$

Now, as seen in Lemma 1, prob(X is bad)

$$= \frac{\binom{4m}{i-1}\left[9i-9\right]_{6i}(36m-6i)}{(36m)!}$$

Thus,

$$X_i = \frac{\binom{6m}{i}\binom{4m}{i-1}\binom{9i-9}{6i}}{\binom{36m}{6i}} \cdot \frac{1}{\text{prob}(G_\pi \text{ is bad})}$$

$$\leq \frac{\binom{9i-9}{6i}}{\binom{26m}{4i+1}} \cdot \frac{1}{\text{prob}(G_\pi \text{ is bad})}$$

So,

$$\sum_{i=3}^{3m} iX_i \leq \left(\frac{1}{\text{prob}(G_\pi \text{ is bad})}\right) \cdot \sum_{i=3}^{3m} i \cdot \frac{\binom{9i-9}{6i}}{\binom{26m}{4i+1}}$$

As in the proof of Lemma 1, either the first or the last term in the summation on the right side is the largest. If the largest term is the last.

$$\sum_{i=3}^{3m} i \frac{\binom{9i-9}{6i}}{\binom{26m}{4i+1}} = O(m^2 2^{-m})$$

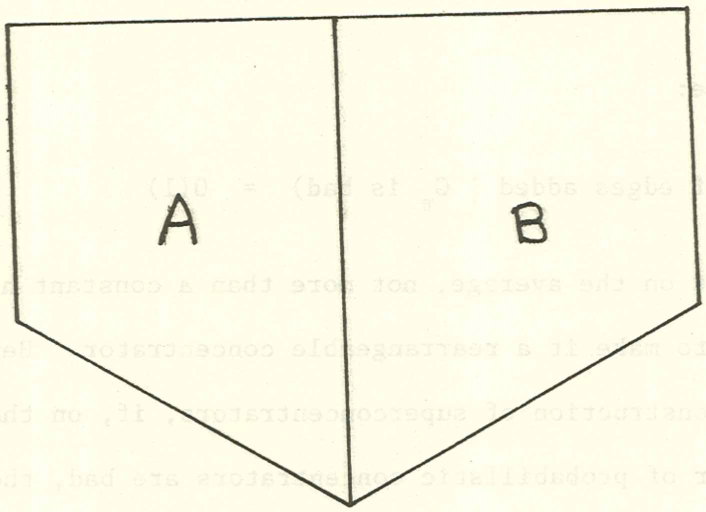
If the largest term is the first,

$$\sum_{i=3}^{3m} i \frac{\binom{9i-9}{6i}}{\binom{26}{4i+1}} \leq \frac{3 \cdot \binom{18}{18}}{\binom{26m}{13}} + O(m^{-15}) = O(m^{-13})$$

Combining the above cases for m larger than some constant m_o,

$$\sum_{i=3}^{3m} i \frac{\binom{9i-9}{6i}}{\binom{26}{4i+1}} \leq c\, m^{-13}, \text{ for some constant } c.$$

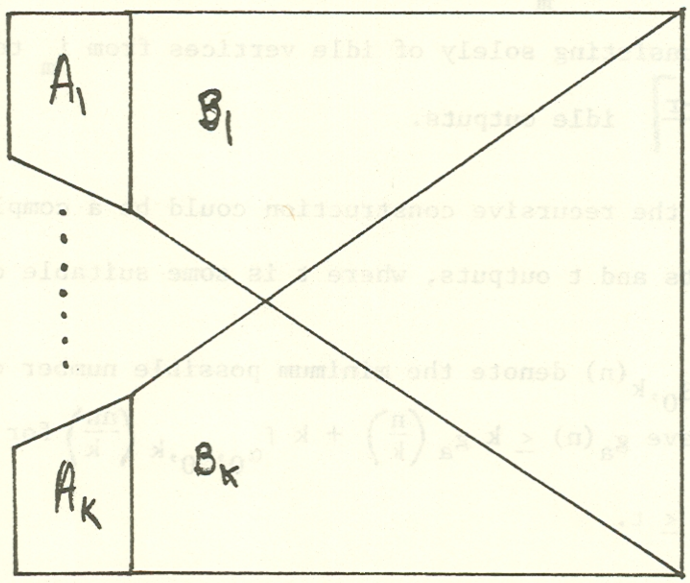
idle outputs. Given such a family of graphs, an incrementally non-blocking connector with n inputs and outputs is formed as shown below:



Here both A and B are copies of $G_a(n)$. In any state in which r connections have been made, any idle input and any idle output can both be connected to at least one common idle vertex in C. It follows that the above construction yields an incremental n-connector.

Let C(n) and $g_a(n)$ denote the minimum possible number of edges in an incremental n connector, and a graph, with n inputs, in $G_a(n)$ respectively. We have that $C(n) \leq 2g_a(n)$. Note also that the problem of constructing incremental n-connectors is reduced to the problem of constructing a family $\{G_a(n)\}$.

To construct $\{G_a(n)\}$ we define a family $\{F_{\alpha,\beta,k}(n)\}$ of networks where $F_{\alpha,\beta,k}(n)$ is a network with n inputs, kn outputs having the property that for every set X of n inputs, $|T(X)| \geq \beta kn$. We construct $G_a(n)$ recursively using the family $\{F_{\alpha_0,\beta_0,k}(n)\}$ with $\alpha_0 = \frac{(a-1)}{2a}$, $\beta_0 = \frac{(a+1)}{2a}$ as shown below:



In the above, each A_i $(1 \leq i \leq k)$ is a copy of $G_a(n)$ and each B_i is a copy of $F_{\alpha_0,\beta_0,K}(an)$.

First we show that the above construction is correct. Suppose that r input-output connections have been established. We have to show that in this state, if i_m (an input of A_i) is idle, then there are paths consisting solely of idle vertices from i_m to at least $\frac{kan-r}{2}$ idle outputs.

Let $|A_i \cap I| = r$, where I is the set of busy inputs. By induction, there are paths from i_m to at least $\left\lceil \frac{an-r}{2} \right\rceil$ idle inputs of B_j. Now, $\left\lceil \frac{an-r}{2} \right\rceil \geq \left\lceil \frac{an-n}{2} \right\rceil = \alpha_0 an$. By definition of B_j these $\left\lceil \frac{an-r}{2} \right\rceil$ inputs of B_j. are adjacent to at least $\beta_0 ank = \frac{kan+kn}{2} - r$ outputs, of which at least $\frac{kan+kn}{2} - r$ must be idle.

Now, $r \leq kn-1$ since i_m was assumed to be idle, and hence we see that there are paths consisting solely of idle vertices from i_m to at least $\frac{kan+kn}{2} - r = \left\lceil \frac{kan-r}{2} \right\rceil$ idle outputs.

The basis of the recursive construction could be a complete bipartite graph with t inputs and t outputs, where t is some suitable constant.

Letting $f_{\alpha_0,\beta_0,k}(n)$ denote the minimum possible number of edges in $F_{\alpha_0,\beta_0,k}(n)$, we have $g_a(n) \leq k\, g_a\left(\frac{n}{k}\right) + k\, f_{\alpha_0,\beta_0,k}\left(\frac{an}{k}\right)$ for $n > t$ and, $g_a(n) \leq at^2$ for $n \leq t$.

In what follows, we will show that $F_{\alpha_0,\beta_0,k}(n) \leq sn$ where s is a constant for fixed α_0, β_0, and k. Thus it will follow that

$$g_a(n) \leq \left(\frac{san}{\log k}\right) n \log n + O(n).$$

We first establish a non-constructive upper bound on $F_{\alpha,\beta,k}(n)$ with $k = 3$, $\alpha = \frac{1}{3}$, and $\beta = \frac{2}{3}$. It is straightforward to generalize what follows; however the above values of α, β, and k chosen seem to minimize C(n) for the construction technique adopted.

Consider bipartite graphs, with 3n inputs and 9n outputs such that for any set I_n of n inputs, $|T(I_n)| \geq 6n$. Such bipartite graphs clearly satisfy the requirements of the family $F_{\frac{1}{3}, \frac{2}{3}, 3}(n)$. We show that the bipartite graphs with 3n inputs need have no more than 45n edges when $n \geq 20$. The proof is similar in spirit to that of Lemma 1 in Section 3.1.1.

Consider the set P of permutations over Z_{45n}. Let π be an element of P. Define G_π to be the bipartite graph with inputs $i_1, \ldots, i_{3n}$ and outputs $y_1, \ldots, y_{9n}$, and set of edges $\{(i_{x \bmod 3n}, y_{\pi(x) \bmod 9n}) \mid x \in Z_{45n}\}$. We term G_π <u>good</u> if every set of n inputs is adjacent to at least 6n outputs, and <u>bad</u> otherwise.

Let X be any set of n inputs, corresponding to 15n elements of Z_{45n} and let Y be a set of k outputs corresponding to 5k elements of Z_{45n}. The fraction of permutations π such that, in G_π, $T(X) \subseteq Y$ is bounded above by the expression

$$[5k]_{15n} \frac{(45n-15n)!}{(45n)!}$$

Thus the fraction of permutations π for which G is <u>bad</u>, is bounded above by

$$I_n = \sum_{k=3n}^{6n-1} \binom{3n}{n} \binom{9n}{k} \frac{[5k]_{15n} (36n)!}{(45n)!}$$

$$= \sum_{k=3n}^{6n-1} \frac{\binom{3n}{n} \binom{9n}{k} \binom{5k}{15n}}{\binom{45n}{15n}}$$

The lower limit of 3n for k corespondes to the fact that if $|X| = n$ then $|T(X)| \geq 3n$ because of degree constraints.

Thus

$$I_n \leq \frac{\binom{3n}{n}}{\binom{45n}{15n}} \sum_{k=3n}^{6n} \binom{5k}{15n} \binom{9k}{k}$$

Let $L_k = \binom{5k}{15n} \binom{9n}{k}$

$$\text{Then,} \qquad \frac{L_{k+1}}{L_k} = \frac{\binom{5k+5}{15n}\binom{9n}{k+1}}{\binom{5k}{15n}\binom{9n}{k}}$$

$$= \frac{(9n-k)}{(k+1)\left(1 - \frac{15n}{5k+5}\right)\cdots\left(1 - \frac{15n}{5k+1}\right)}$$

We see that as k increases, the numerator in the expression above decreases, while each term in the denominator increases. Thus,

$$\frac{L_{k+1}}{L_k}$$

is a decreasing function of k. Furthermore, the last ratio in the series is

$$\frac{L_{6n}}{L_{6n-1}} = \frac{(3n+1)}{6n}\frac{(30n)}{(15n)} \cdots \frac{(30n-4)}{(15n-4)}$$

which is greater than 1, for all n. Hence, we conclude that the largest term in the summation is the last, and therefore,

$$I_n \leq 3n \frac{\binom{3n}{n}\binom{30n}{15n}\binom{9n}{6n}}{\binom{45n}{15n}}$$

$$= O(n2^{-n})$$

For $n \geq 20$ we find that the right side of the inequality above is less than 1. For $n \leq 20$, we would have to check directly from the definition of I_n, using a table of binomial coefficients, that $I_n < 1$.

However, in any case, we have

$$f_{\frac{1}{3}, \frac{2}{3}, 3}(n) = \begin{cases} 15n & , \quad n \geq 60 \\ 3n^2 & , \quad n \leq 60 \end{cases}$$

Thus,

$$g_3(n) = \begin{cases} 3g_3\left(\frac{n}{3}\right) + 45n & , \quad n \geq 60 \\ O(1) & , \quad n \leq 60 \end{cases}$$

which yields $g_3(n) \leq \left(\frac{45}{\log 3}\right) n\log n + O(n)$ and, hence,

$$C(n) \leq \left(\frac{90}{\log 3}\right) n\log n + O(n)$$

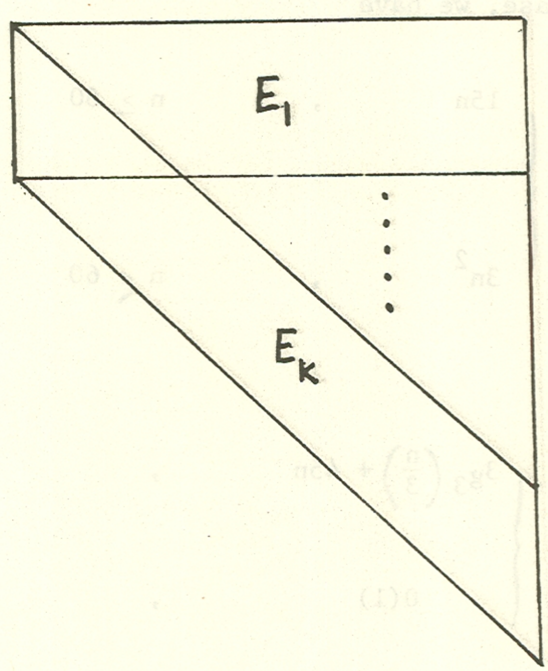
or,

$$C(n) \leq 56\cdot 79 \; n\log n + O(n),$$

as compared to $C(n) \leq 67\cdot 26 \; n\log n + O(n)$ in [1].

Next we obtain an explicit construction for a family $\{G_a(n)\}$ with $a \simeq 30\cdot 064$, using the explicit construction of $(n, \frac{2}{15})$-expanders in [2].

Consider the following construction for $F_{\alpha,\beta,k}(an)$



Each E_i above is an $(an, \frac{2}{15})$-expander such that, for every X

$$|T(X)| \geq |X| \left(1 + \frac{2}{15}\left(1 - \frac{|X|}{an}\right)\right)$$

For any set X of inputs, since the outputs of E_i are disjoint,

$$|T(X)| \geq k|X| \left(1 + \frac{2}{15}\left(1 - \frac{|X|}{an}\right)\right)$$

Now we require the property that if $|X| = \alpha an$ then $|T(X)| \geq \beta kan$, where $\alpha = \frac{(a-1)}{2a}$ and $\beta = \frac{(a+1)}{2a}$ as argued earlier. Setting $|X| = \alpha an$, we note it is sufficient to have

$$k\alpha na\left(1 + \frac{2}{15}(1-\alpha)\right) \geq k\beta a n$$

or equivalently

$$a\left(\frac{a-1}{2a}\right)\left(1 + \frac{2}{15}\left(1 - \frac{a-1}{2a}\right)\right) \geq \left(\frac{a+1}{2a}\right) a$$

which reduces to $a^2 - 30_a - 1 \geq 0$, which is satisfied by $a \simeq 30\cdot064$.

Thus we obtain $f_{\alpha,\beta,k}(an) \leq 7ank$ for the values of α,β determined by the value of a above, since each E_i has at most 7n edges. Now,

$$g_a(n) \leq kg_a\left(\frac{n}{k}\right) + k\, f_{\alpha,\beta,K}\left(\frac{an}{K}\right)$$

or

$$g(n) \leq kg_a\left(\frac{n}{k}\right) + 7ank$$

Also, $g_a(1) = 1$, which yields $g_a(n) \leq \left(\frac{7ak}{\log k}\right) n\log n + O(n)$.

With $a = 30\cdot034$, and $k = 3$ which minimizes the term $\left(\frac{k}{\log k}\right)$, we finally have,

$$C(n) \leq 2g_a(n) \leq 795\ n\log n + O(n)\ .$$

3.3 CONCLUSIONS AND OPEN PROBLEMS

We have studied constructive as well as non-constructive upper bounds on the structural complexity of concentration and connection networks. Although the O(n) and O(nlogn) constructions for rearrangeable concentrators and incremental connectors respectively are asymptotically optimal, they are inefficient in practice. The non-constructive proofs of upper bounds suggest that these networks may be constructed efficiently in a probabilistic manner, with arbitrarily small, fixed, blocking probability. In addition, blocking networks may be made non-blocking by adding edges to the network during operation. On the average, very few edges need be added to a blocking network. Although we have shown this only for rearrangeable concentrators, the argument is easily extended to other networks.

Probably the most intriguing open problem is to find a combinatorial construction for linear expanders and concentrators. It would also be interesting to obtain tighter upper bounds on the complexity of concentrators and hyperconcentrators than those obtained by utilizing Pippenger's superconcentrators. Finally, the structural complexity - operational complexity tradeoff for networks needs to be better understood.

REFERENCES

1. L.A. Bassalygo and M.S. Pinsker: On the Complexity of Optimal Non-blocking Switching Networks Without Rearrangement, Problems of Information Transmission, vol. 9 (1973) 84-87.

2. O. Gabber and Z. Galil: Explicit Constructions of Linear Size Superconcentrators, 20th Annual Symposium on Foundations of Computer Science, (October, 1979) 364-370.

3. G. A. Margulis: Explicit Constructions of Concentrators, Problems of Information Transmission, vol. 9 (4) (1973) 71-80.

4. M. S. Pinsker: On the Complexity of a Concentrator, Seventh International Teletraffic Congress, Stockholm (1973).

5. N. J. Pippenger: Superconcentrators, SIAM Journal of Computing (6) (1977) 298-304.

6. N. J. Pippenger: The Complexity Theory of Switching Networks, Ph.D. Thesis, Department of Electrical Engineering, Massachusetts Institute of Technology, August 1973.

7. L. G. Valiant: On Non-linear Lower Bounds in Computational Complexity, Proceedings of the 7th Annual ACM Symposium on Theory of Computing, Albuquerque, New Mexito (May 1975) 45-53.

8. A. Waksman: A Permutation Network, Journal ACM 15, 1 (January 1968) 159-163.